MARI

USER GUIDE

VERSION 8.0v1

THE FOUNDRY.

Mari® User Guide. Copyright © 2026 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this Mari User Guide and the Mari software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This User Guide and the Mari software may be used or copied only in accordance with the terms of the EULA. This User Guide, the Mari software and all intellectual property rights relating thereto are and shall remain the sole property of The Foundry Visionmongers Ltd. ("Foundry") and/or Foundry's licensors.

The EULA can be found in this Mari User Guide, End User License Agreement (EULA).

Foundry assumes no responsibility or liability for any errors or inaccuracies that may appear in this User Guide and this User Guide is subject to change without notice. The content of this User Guide is furnished for informational use only.

Except as permitted by the EULA, no part of this User Guide may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, electronic, mechanical, recording or otherwise, without the prior written permission of Foundry. To the extent that the EULA authorizes the making of copies of this User Guide, such copies shall be reproduced with all copyright, trademark and other proprietary rights notices included herein. The EULA expressly prohibits any action that could adversely affect the property rights of Foundry and/or Foundry's licensors, including, but not limited to, the removal of the following (or any other copyright, trademark or other proprietary rights notice included herein):

Mari® software © 2026 The Foundry Visionmongers Ltd. All Rights Reserved. Mari® is a registered trademark of The Foundry Visionmongers Ltd.

Weta™ Digital is a trademark of Weta Digital Ltd.

In addition to those names set forth on this page, the names of other actual companies and products mentioned in this User Guide (including, but not limited to, those set forth below) may be the trademarks or service marks, or registered trademarks or service marks, of their respective owners in the United States and/or other countries. No association with any company or product is intended or inferred by the mention of its name in this User Guide.

Linux ® is a registered trademark of Linus Torvalds.

Windows ® is the registered trademark of Microsoft Corporation.

Mac and Mac OS X are trademarks of Apple, Inc., registered in the U.S. and other countries.

Adobe ® and Photoshop ® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Maya ® is a registered trademark of Autodesk, Inc., in the USA and other countries.

Mari includes Disney technology licensed from Walt Disney Animation Studios.

The Foundry Visionmongers Ltd.

5 Golden Square,

London, W1F 9HT

UK

Rev: Wednesday, October 14, 2026

Contents

System and Hardware Requirements 24

Installing on Windows 27

Installing on Linux 40

Mari Non-Commercial 51

Discover the Workspace 55

Mari 3.3 vs 4.0 57

Ready-to-use Shelf Content 71

Managing Projects 87

Customizing the View 100

Setting the Main View Options 100

Navigating the Mari Canvas 103

Configuring Your Workspace 105

Configuring the Lighting 110

Setting the Perspective Camera Details 117

Configuring Navigation 119

Customizing your splash screen 120

Using Shortcuts 122

Viewing the Default Shortcuts and Gestures 123

Trying the Mouse Gestures 124
Customizing General Keyboard Shortcuts 125
Setting Keyboard Shortcuts for Project Items 126
Quick Palettes and Pie Menu 128
Using the Pie Selection Control 129
Using the Paint Target Quick Palette 130
Unassigning Shortcuts 131
Configuring Brushes 132
Reviewing the Preset Brushes 133
Brush Properties 134
Changing Brush Properties "On the Fly" 151
Customizing a Brush 152
Changing Brush Color Dynamically 154
Exporting and Importing Brushes 161
Storing Resources Using Shelves 164
Opening Shelves 165
Adding a Color to a Shelf 166
Adding an Image to a Shelf 168
Adding Materials to a Shelf 169
Adding Nodes to a Shelf 171

Creating a Custom Shelf 172
Adding any Function to a Shelf 172
Run Custom Python Scripts Directly from Shelves 173
Modifying Custom Shelves and Their Contents 179
Importing and Exporting Shelves and Items 180
Creating a Floating Shelf 182
Painting 183
Modeling Requirements 190
Controlling the Way Mari Applies Paint to Your Model 191
Paint Buffer 195
Configuring the Paint Buffer 198
Erasing Paint 202
Masking What You Can Paint On 204
Edge Masks 205
Channel Masks 208
Ambient Occlusion Masks 212
Depth Masks 213
Backface Masks 215
Fractal Noise Masks 215
Adding a mask to a node 217
Creating masks from colors 221
Creating a mask from a selection 227
Creating ID Masks from Selection Groups 229
Setting the Paint Blending Mode 231

Paint Blending Modes 232
Paint Buffer Symmetry 246
Mirror Projection 248
Paint Continuous, Repeating Textures with the Roller Brush 257
Transforming and Warping Paint 263
Moving or Transforming Paint Buffer 264
Warping Paint Using a Grid 266
Pulling, Smudging, Growing, and Shrinking Paint 270
Distorting the Paint with Pins 273
Towing Paint Around 276
Selecting and Towing Paint 277
Copying and Pasting Paint 279
Editing Paint on Layers 281
Blurring Paint on a Layer 282
Pulling Paint into the Buffer 283
Clone Stamping 283
Choosing a Source Point 284
Setting a Start Point 285
Image Options 286
Setting the Clone Stamp Options 286
Clone Stamp 290
Applying a Color Grade as You Clone 292
Painting Through an Image or Gradient 295
Paint Through 296
Setting the Paint Through Options 297

Painting Through an External Image 300
Cloning from a Floating Image 302
Using Tint and Stencil Options 303
Painting Through a Gradient 304

Multi-Paint 306

Managing Images 311
Opening the Image Manager Palette 313
Organizing images in the Image Manager 313
Image Groups 314
Organizing images in the Image Manager 318
Loading and Selecting Images 318
Managing Megascans Assets 322
Cropping and Filtering Images 323
Running Scripts on Images 325
Using 2D Paint Mode 326

Managing Colors 329
Picking Colors from Paint on the Model 331
Selecting Colors from a Picker or Image 335
Color Management 339
Setting How Colors Are Displayed Onscreen 343
Color Data and Scalar Data 348
Advanced Color Management 354

Managing Materials in Mari 355
Example Material Design Project 357
Creating Materials with the Material Ingest Tool 358
Working with Materials in the Layers Palette 367
Multi-Channel Layer Workflow Rules 396

Filtering Images and Paint 409
Selecting and Applying Your Filter 410
Available Filters 413
Vector Brush 426
Painting Vectors 427
Modifying Vectors with Paint Blending Modes 428
Inspecting Your Vector Work 429
Adding Vector Shaders 431
Adding Vector Adjustments and Procedurals 432
Changing the File Space 434
Creating Flow Maps on Multiple Objects 435
Painting Animated Objects 436
Loading Animated Objects 439
Playing Animation 440
Painting Animated Textures 442
Exporting Animated Textures 443
Configuring the User Interface 444
Choosing and Organizing Palettes 445
Choosing and Organizing Toolbars 446
Creating Custom Toolbars 447
Configuring the HUD 449
Saving Your Layout 450
Understanding the Status Bar 450

Status Bar Icons 451

Configuring USD Plug-ins 454

Items and Objects 455

Selecting Items 455

Selecting Objects, Patches, or Faces 456

Hiding and Locking 461

Saving Selections on the Model 464

Selecting Areas Onscreen 467

Moving and Distorting Marquee Selections 470

Working With Patches 471

Viewing and Resizing Patches 474

Resizing Patches 475

Using UDIM Values 476

Extracting Patches to the Image Manager 477

Filling Patches with Color 477

Flipping and Rotating Painting 478

Mirroring Painting 478

Saving Images of the UV Masks 479

Bleeding Across Patch Edges 480

Linking or Unlinking Patches 482

Working with Objects 484

Adding Objects 485

Filling Objects with Color 488

Adding and Removing Object Versions 489

Adding Locators for Objects 492

Adding Geo-Channels to Objects 494
Duplicating Objects 496
Removing Objects and Locators 497
Moving, Viewing, and Locking Objects 497
Calculating Ambient Occlusion 501
Subdividing Objects 501
Shadow Settings 505
Ptex 506
Ptex Modeling Requirements 508
Creating a New Project with OBJ and Ptex Files 509
Setting the Ptex Face Texture Options 511
Managing Ptex Faces 512
Exporting a Ptex Channel or Layer 513
Import a Ptex Channel or Layer 516
Painting Preservation with Multiple Ptex Geometries 519
Alembic and FBX 520
Alembic and FBX Modeling Requirements 522
Objects and Object Hierarchies 523
Animated Geometry 525
Deformations 526
Creating a New Project with Alembic and FBX Files 527
Adding and Merging Alembic or FBX Geometry 530
Importing Objects 533
Importing New Versions 533
Importing Cameras 535
Naming of Objects 536
Selection Groups 537

Shading Networks 539

Channels 539

Creating and Deleting Individual Channels 540

Flattening Channels 544

Creating Multiple Channels from Presets 544

Navigating Between Channels 547

Setting and Changing Channel Properties 547

Taking and Managing Channel Snapshots 548

Using the Snapshots Palette 549

Autosnapshots 551

Sharing Channels 552

Locking and Unlocking Channels 552

Exporting Channels 555

Exporting Flattened Channels 556

Importing Channels 557

Transferring Channels 558

Layers 559

Opening the Layers Palette 564

Layer Blend Modes 565

Adding Layers and Masks 566

Adding Adjustments and Procedurals 568

Adding Graph Layers 572

Cube Mapping 573

Vector Brushing 574

Removing and Disabling Layers 574

Groups, Pass-Through, and Mask Stacks 575

Sharing Layers 581
Caching and Uncaching Layers 583
Exporting and Importing Layers or Masks 585
Exporting Layers 585
Exporting Masks 586
Exporting Flattened Layers 586
Exporting Flattened Masks 587
Importing Layers 587
Converting Masks 588
Transferring Layers 590
Layer Types and Examples 591

Shaders 618
Opening the Shaders Palette 622
Default Shaders 624
Creating a Layered Shader 629
Setting Shader Details 631
Removing a Shader 634
Shader Types and Examples 634

Node Graph 646
Working with Nodes 650
Connecting Nodes 661
Navigating Inside the Node Graph 666
Working with Node Properties 672
Using Backdrops, StickyNotes, Groups, and Gizmos 674
Transferring Nodes to a Different Object 682
Variations with the Switch node 685

Baking in Mari With the Bakery 688

Texture Transfers Between Projects 697

Previewing and Rendering 699

Projectors 699

Setting the View 700

Creating the Projector 701

Unprojecting from Your Projector 704

Editing the Files Externally 704

Projecting the Files Back onto the Model 706

Using the Quick Project Functions 706

Projecting onto a Model Using .fbx Files 708

Generating a Turntable from a Projector 710

Generating a Render Turntable 711

Generating a Diagnostic Turntable 713

Saving, Loading, and Deleting Projectors 715

Previewing and Rendering 715

Rendering with Modo 716

Baking from Modo 719

Modo Render Settings 721

Exporting, Importing, and Session Scripts 723

Exporting and Importing Textures 723

Importing Textures to Channels 724

Importing Individual Files 725

Importing a Sequence of Files 726

Importing Textures to Layers 729
Exporting Textures from Channels 735
Exporting Textures from Layers 740
Exporting to Maya 747
Exporting EXR Files for Render 748
Exporting a Flattened USD Preview of Your Asset's Look 750
Session Scripts 753
Exporting a Session 754
Importing a Session 756
Importing Channels 757
Importing Shaders 758
Importing Node Graphs 759
Mari Quick Start Guides 761
Setting up a Project 764
Setting the View 769
Painting 780
Painting Through and Clone Stamping 789
Exporting and Importing 800
Mari Sample Projects 811
Example Project: Patches 812
Sample Project: Material Design 818
Sample Project: Example 824

Reference Guide 836

Nodes 836

Bubbles Node 836

Camouflage Node 847

Channel Node 851

Color Node 857

Color to Scalar Node 860

Compare node 862

Constant Node 863

Vector Node 866

Brightness Node 871

Clamp Node 875

Color Mask node 879

Color Remap node 880

Color Switch Node 881

Contrast Node 884

Multi Color Mask node 886

Smart Mask Node 888

Exposure Node 904

Gamma Node 905

Grade Node 908

Height As Normal Node 912

HSL Node 915

HSV Node 919

Hue Shift Node 922

Image Write Node 926

Invert Node 927

Levels Node 934
Multi-Channel Merge Node 941
Multi-Channel Bake Point Node 953
Multi-Channel Paint Node 965
Normal Strength Node 977
Saturation Node 979
Scalar to Color Node 982
Shuffle Node 984
Switch Node 988
Ambient Occlusion Node 989
Curvature Node 993
Geo-Channel Node 995
Position Node 1000
UDIM Mask Node 1002
UV Node 1005
UV Mask Node 1007
View as Color Node 1014
View as Scalar Node 1016
Bottom Transparency Node 1018
Merge Node 1019
Ceil Node 1043
Distance Node 1046
Floor Node 1047
Fract Node 1050
Length Node 1051
Mix Node 1054
Modulo Node 1055

Normalize Node 1055
Power Node 1059
Square Root Node 1062
Vector Dot Node 1065
Backdrop Node 1066
Bake Point Node 1070
Image Input node 1083
Paint Node 1095
Sticky Note Node 1103
Teleport Broadcaster and Teleport Receiver Nodes 1107
Vector Combine Node 1112
Vector Split Node 1117
Atlas Random Node 1119
Cellular Node 1130
Checkerboard Node 1136
Cloud Node 1141
Hex Tile node 1152
Marble Node 1154
Oil Node 1158
Perlin Node 1166
Object Space Linear Gradient Node 1172
Scratches Node 1180
Squiggle Node 1184
Tiled Node 1190
Turbulence Node 1201
UV Linear Gradient Node 1208
Voronoi Node 1211

Weave Node 1214
Wood Node 1219
Camera Distance Gradient Node 1229
Camera Facing Gradient Node 1231
Camera Projection Node 1233
Object Distance Gradient Node 1238
Object Facing Gradient Node 1240
Projection Node 1242
Triplanar Projection Node 1248
Tri-Planar Tiled node 1261
Two Point Gradient Node 1264
BRDF (Shader Network) Node 1266
Coordinate Transform node 1278
Locator Transform node 1281
Rotation Transform node 1282
Scale Transform node 1282
Translation Transform node 1283
UV Radial Transform node 1284

Types of Controls 1286
Changing Numerical Values 1291

Palettes 1292
Channels Palette 1292
Layers Palette 1296
Multi-Paint Palette 1328
Node Graph Palette 1330
Node Properties Palette 1330
Patches Palette 1331

Objects Palette 1332

Shaders Palette 1338
3Delight Principled 1340
BRDF 1346
VRayMtl 1349
Principled BRDF 1358
Arnold Standard Surface 1363
Unreal 1374
Layered 1377

Lights Palette 1380

Projectors Palette 1384

Selection Groups Palette 1399

Color Manager (View Transform) Palette 1400

Image Manager Palette 1414

Texture Sets Palette 1419

History View Palette 1421

Python Console Palette 1421

Colors Palette 1422

Tool Properties Palette 1430

Painting Palette 1439

Shelf Palette 1451

Snapshots Palette 1455

Modo Render Palette 1457

Control Functions 1461

Project Functions 1462

Preference Functions 1464

Selection Functions 1465

Object Functions 1475

Channel Functions 1479

Layer Functions 1486

Node Graph Functions 1499
Ptex Functions 1503
Patch Functions 1504
Shading Functions 1509
Painting Functions 1513
Filter Functions 1515
Camera Functions 1526
View Functions 1532
Tool Functions 1534
Python Functions 1534
Help Functions 1535

Dialogs 1537

Project Dialogs 1537
New Project Dialog 1537
Project Settings Dialog 1555
Manage Toolbar Dialog 1558
Manage Keyboard Shortcuts Dialog 1560
HUD Manager Dialog 1562
Mari Preferences Dialog 1568
Display Properties Dialog 1604
Screenshot Settings Dialog 1607

Image Dialogs 1608
Select Color Dialog 1609
Advanced Blending Mode 1617
Add Channel Dialog 1619
Add Paint/Bake Point Dialog 1621
Channel Presets Dialog 1623
Masks Formats Dialog 1629
Convert Type Dialog 1630
Copy Multiple Textures Dialog 1630
Channel Transfer Dialog 1632
Layer Transfer Dialog 1635
Mask from Selection Dialog 1638
Mask from Selection Group Dialog 1639
ID Mask from Selection Group Dialog 1640
Node Transfer Dialog 1642
Open an Image Dialog 1644
Save As Dialog 1647

Select Patches By Range Dialog 1649
Assign Channel Dialog 1650
Material Ingest Tool Dialog 1651
Create Shader Dialog 1657

Object Dialogs 1658
Add Object Settings Dialog 1659
Add Version Settings Dialog 1662
Subdivide Dialog 1664
Gaussian Blur Dialog 1668
Height Dialog 1670
Displacement Dialog 1672

Import and Export Dialogs 1674
Export Manager Dialog 1674
Export Dialog 1682
Import Dialog 1687
Mari to Maya Export Dialog 1691
Load Camera Dialog 1693
Render Turntable Dialog 1694
Diagnostic Turntable Dialog 1697
Quick Projection Dialog 1700
USD Preview Export Dialog 1701

Toolbars 1703

Project Toolbar 1703

Palettes Toolbar 1704

Paint Buffer Toolbar 1709

Tools Toolbar 1710

Canvas toolbar 1716

Tool Properties Toolbar 1717

Ptex Toolbar 1741

View Transform Toolbar 1742

Navigation Toolbar 1745

Vector Painting Toolbar 1746

Project Controls toolbar 1747

Status Bar 1752

Play Controls Toolbar 1755

Nuke<>Mari Bridge 1757
Installation 1758
Connection Setup 1758
Communicating with Mari 1761
Workflow 1762
Using the Nuke<>Mari Bridge over a Network 1767
Software API Overview 1769
Using Mari's C API 1777
Custom Shaders 1778
Extending Mari 1780
Adding Custom Attributes to Objects 1780
Removing Custom Attributes from Objects 1782
Reviewing the Custom Shortcut File Format 1782
Using Modifier Keys 1784
Example: DefaultShortcuts.conf 1785
Defining Custom Shortcuts 1788
Reviewing the Custom Channel Preset File Format 1789
Example: DefaultChannels.mtd 1793
Defining Custom Channel Presets 1796

Reviewing the Custom Channel Template File Format 1796

Example File: DefaultTemplates.mtt 1797

Defining Custom Channel Templates 1798

Environment Variables That Mari Recognizes 1798

Image Importing 1808

Overriding Mari Configuration File 1808

Setting Environment Variables 1809

Checking Environment Variables Are Set 1811

Importing Plug-ins 1811

Crash Reporting in Mari 1812

Appendices 1815

Mouse Gestures 1825

Appendix B: Frequently Asked Questions and Troubleshooting 1827

Appendix C: Third-Party Licenses 1833

End User License Agreement (EULA) 1902

System and Hardware Requirements

Note: Mari increases its level of performance with newer, more advanced hardware configurations. However, Mari is tested and capable of operating on many older, prior-generation systems. For this reason we are listing below-minimum requirements, recommended, and on which tests have been performed. Your particular needs may vary from that of other users.

System Requirements

Officially Supported Operating Systems

- Windows 11 64-bit
- Linux 64-bit operating system (Rocky/RHEL 9)

For a full list of minimum and recommended hardware specifications and tested GPUs, see the Mari technical specification on our website: https://www.foundry.com/products/mari/requirements

What Hardware to Use for Mari

This section introduces which Mari features use which pieces of hardware. The following are hardware suggestions to improve Mari performance.

Article: Take a look at the Optimising your Mari projects for best performance article.

GPU - Speed and number of cores

The GPU is used for rendering and also for baking out results to textures. Faster GPUs can render a heavier scene at a better frame rate and shorten the waiting time for baking to textures, such as flattening layers, merging layers, and converting procedural to paintable.

GPU - Memory

The more GPU memory you have, the easier it is to paint more details in general. The two main features that use a lot of GPU memory are the following:

- **Paint Buffer** - If you have a lot of GPU memory, you can set the **Buffer Size** (paint buffer's resolution) higher and/or set the **Color Depth** (bit depth) of the paint buffer higher (16- or 32-bit instead of 8-bit). A higher resolution paint buffer allows Mari to put more details without having to repeatedly zoom, paint, and bake. If the paint buffer is set to a higher **Color Depth**, it prevents stepping especially for displacement maps.

> **Note:** To configure the paint buffer, see Configuring the Paint Buffer

- **Virtual Texture Atlas** - Mari uses virtual texturing to contain a large texture data into GPU Memory. However, the virtual texture is limited. If Mari can't process all of the data in a timely manner, it starts using lower resolution mipmaps.

  If you have a lot of GPU memory, you can set the **Resolution** (virtual texture size) to a high resolution so that Mari can render a really heavy scene with, for instance, a scene with lots of layers, UDIMs, and/or fragmented UV bits.

> **Note:** To set the **Resolution**, see the Mari Preferences Dialog.

> **Article:** Take a look at the Calculating your GPU memory usage for Virtual Textures in Mari article.

GPU Requirements for the Bakery

In order to use the Bakery, at a minimum your graphics card must support GPU-accelerated Ray Tracing. Cards with dedicated Ray Tracing hardware (e.g. anything with 'RTX' in the name for NVIDIA cards, and AMD

6000/7000 families) are supported, but some older cards such as the NVIDIA GTX 1080 line also support Ray Tracing and have been tested with the Bakery.

Using multiple GPUs for baking is also supported.

If you are unsure if a specific card has been tested or want to check whether we expect it to work, please contact support@foundry.com and we will be happy to investigate for you.

CPU

In general, a moderate quad-core processor should be 'good enough', but some non-GPU operations benefit from more cores or faster CPUs. The following are examples of non-GPU operations in Mari:

- Ambient occlusion calculation
- Whole patch bleed
- Tile level bleed after baking
- Changing bit-depth of a channel
- Changing resolutions of textures

RAM

4GB of RAM is sufficient to run Mari, but a RAM of 8GB is better for more stable operations especially if you're running other 3D applications. If you want to work with heavy scenes, it's recommended to have as much RAM as possible.

Ultimately, all data in Mari is cached to disk, so even if you don't have a lot of RAM, Mari still runs properly, but there is more disk reading. The areas where Mari uses a lot of RAM are:

- General application (UI, application logic, and so on), and
- The texture data loaded into the RAM from a disk stays in the RAM, but is removed from the RAM in Least Recently Used (LRU) cache.

Disk

A solid-state disk (SSD) is highly recommended. Lengthy operations are often bottle-necked by disk writing so having an SSD greatly helps reduce the time taken. Regardless of the data processed by CPU or GPU, the resulting data is ultimately written to disk.

If your project is light, for example, it uses a few UDIMs with up to 4k textures, an SSD may not make a big difference.

Note: Scratch space has not been tested yet. Mari manages its own data by keeping the recent data in RAM using a Least Recently Used (LRU) algorithm while all data is written to a disk.

Installing on Windows

For a full list of minimum and recommended hardware specifications and tested GPUs, see the Mari technical specification on our website: https://www.foundry.com/products/mari/requirements

Mari 8.0 is available to download from our web site at https://www.foundry.com/products/mari as a **.zip** file. In the extracted directory, you'll find an **.msi** executable and several **.cab** files, which are required for installation on Windows.

1. Download and extract the **.zip** installation file from our web site.
2. Double-click on the **.msi** file to start the installation. Follow the on-screen instructions. By default, Mari is installed to <drive letter>**:\Program Files\Mari** <version number>.
3. That's it! Proceed with Launching on Windows.

Installing from the Command Line

1. Download the **.zip** file from our web site at https://www.foundry.com/products/mari and extract the contents.
2. To open a command prompt window, select **Start** > **All Programs** > **Accessories** > **Command prompt**.
3. Use the **cd** (change directory) command to move to the directory where you saved the **.msi** installation file. For example, if you saved the installation file in **C:\Temp**, use the following command and press **Return**:
   ```
   cd \Temp
   ```
4. To install Mari, run the following command:
   ```
   start /wait msiexec.exe /i Mari8.0v1-win-x86-release-64.msi ACCEPT_FOUNDRY_
   EULA=ACCEPT /qb
   ```

Note: By using the **ACCEPT_FOUNDRY_EULA=ACCEPT** property, you agree to the terms of the EULA. To see this agreement, please refer to *Appendix D: End User Licensing Agreement* in the *Mari User Guide*, or run the installer in standard non-silent mode.

Tip: Tip: The **/passive** argument can be used in place of **/qb** to suppress popup error messages during the installation. In this case it is advisable to also use **/l** to enable logging so that you can check the log file for errors in the event of a failed installation: `start /wait msiexec.exe /i Mari8.0v1-win-x86-release-64.msi ACCEPT_FOUNDRY_EULA=ACCEPT /passive /l log.txt`

Tip: To install Mari without any Shortcuts, use the **ADDDEFAULT MSI** property: `start /wait msiexec.exe /i Mari8.0v1-win-x86-release-64.msi ACCEPT_FOUNDRY_EULA=ACCEPT ADDDEFAULT=ProductFeature /qn`

By default, the application is installed to **Program Files > Mari 8.0v1**

Tip: You can display a list of install options by running the installer command by itself: `msiexec.exe`

5. Proceed with the following section, **Launching on Windows**.

Launching on Windows

Article: Take a look at the Launching Mari in safe mode article.

Launching Mari on High-Resolution Monitors

On Windows, QT5 does not handle scaling properly on high-resolution monitors. Windows includes an in-built mode, called `--dpiscaling=native`, which displays Mari with the correct resolution. If you need to disable this mode, you can use the `--dpiscaling=none` mode.

To set the `--dpiscaling=none` mode:

1. Before launching Mari, **Shift**+right-click on the Mari shortcut icon and select **Properties**.

   The Mari 8.0v1 **Properties** window opens.
2. In the **Shortcut** tab, in the **Target** field, type `--dpiscaling=none` after **"C:\Program Files\Mari4.0v1\Bundle\bin\Mari4.0v1.exe"**. For instance, **"C:\Program Files\Mari4.0v1\Bundle\bin\Mari4.0v1.exe" --dpiscaling=none.**

Note: The `--dpiscaling=native` mode is set by default but does not display in the **Target** field.

Launching Mari

1. Click **Start** > **All Programs** > **The Foundry** > **Mari 8.0v1** > **Mari 8.0v1**.
2. If you haven't installed a license for Mari, you need to choose from one of the options for . Once you've done so, click **Launch** in the **Mari Licensing** dialog.
3. Mari checks for the most frequently used graphics cards, and if a graphics card configuration is detected that's not in the tested GPU hardware list, a dialog displays warning you about it. Depending on the configuration detected, you have the option to **Quit** or **Continue** and, in the case of configurations with known issues, also **Ignore Permanently**.

Note: If you ignore the warning and run Mari anyway, you may experience instability or performance issues. If a tested graphics card configuration is detected, Mari runs as normal.

4. If you are launching Mari for the first time, Mari selects a default cache directory for its project files. You can change this directory and select one of your own. In the **Confirm Projects Directory** dialog, navigate to the directory you want to use and click **Choose**.

   The Mari graphical interface displays.

   Please note that the project directory should be:

   - empty
   - local to the machine (not a network mount)
   - as fast as possible
   - a location that's not temporary (to avoid the data disappearing)
   - a directory that the user has read and write permissions to.

Project cache locations are directories that Mari uses to store project data files. These directories must be persistent; not in temporary locations. If more than one project location is specified, Mari spreads each project's data files across all locations. Once a project has been created for a given set of project cache locations, that set must not be changed. In order to move a project between different sets of caches, archive the project.

Note: Do not add, remove, or amend project locations once they have been set unless they are empty. Once projects have been created for a given set of project cache locations, that set must not be changed or project corruption will occur.

Note: If the MARI_CACHE environment variable has been set, its value is used to determine the location of the project cache files. In this case, Mari does not prompt you to select the project directory. For more information on environment variables that Mari understands, please refer to Environment Variables That Mari Recognizes.

Article: Take a look at the Temporarily setting the project directory in Mari article.

Note: You can also launch Mari in verbose mode by selecting **Start** > **All Programs** > **Mari 8.0v1** > **Mari 8.0v1 (Verbose Output)**. This way, Mari provides a running log of each action that it performs. You can see this log in a command line window.

5. If you are upgrading, Mari prompts you if you'd like to import your user preferences from your previous installed version of Mari.

If you are launching Mari from the command line, you can run the application in one of two safe modes: `--safe` and `--safer`. The `--safe` mode:

- Disables startup Python scripts
- Disables custom user plug-ins
- Bypasses user Python libraries.

The `--safer` mode performs all of the above options as well as:

- Bypasses user settings.
- Bypasses user environment variables.

These safe modes are primarily used for debugging purposes. For advanced debugging methods, please see *Appendix B: Frequently Asked Questions and Troubleshooting* in the *Mari User Guide* for more information.

Mari Non-Commercial

If you want to try out or learn Mari, you can run Mari Non-commercial. This version allows you to explore most of Mari's features, but prevents the commercial use of the application. For more information, see Mari Non-Commercial.

To launch the application in Non-Commercial mode, do one of the following:

- Double-click the **Mari 8.0 (Non-commercial)** icon on the Desktop.
- Using a command prompt, navigate to the Mari application directory and enter:

```
.\Mari8.0.exe --nc
```

> **Note:** The default install location on Windows is:
> `C:\Program Files\Mari<version>\Bundle\bin`

If you have already activated Mari Non-commercial on the current device, the graphical interface appears, and a command line window opens. If you haven't activated the device yet, proceed to the next section.

Licensing on Windows

If you simply want to try out Mari, you can obtain a trial license, which allows you to run Mari for free for 15 days. See Install the License.

To use Mari after this trial period, you need either a valid **subscriptionlicense**, **activation key**, **node locked license**, or a **floating license** and server running the Foundry Licensing Tools (FLT):

- **SubscriptionLicenses** - allows you to activate a single license or entitlement on any authorized device up to the entitlement's maximum number of activations.
- **Activation Keys** - activation keys allow you to activate and generate your actual product license key, at a later point after purchase, on the machine for which you require the license. They are provided for both node locked and floating license, and generate the appropriate license type once installed using the product's Licensing dialog or online using the **Activate a Product** page: https://www.foundry.com/user/login?destination=/licensing/activate-product
- **Node Locked Licenses** - these can be used to license an application on a single machine. They do not work on different machines and if you need them to, you'll have to transfer your license.

  Node locked licenses, sometimes called uncounted licenses, do not require additional licensing software to be installed.
- **Floating Licenses** - these are used to license applications on any networked client machine. The floating license is put on the server and is locked to a unique number on that server.

  Floating licenses, sometimes called counted licenses, require additional software to be installed on a server to manage the licenses and give them out to the client stations that want them. This software is called the Foundry Licensing Tools (FLT) and can be downloaded at no extra cost from our website.

The instructions below run through both licensing methods. You can also find a more detailed description in the *Foundry Licensing Tools User Guide* available on our website: https://www.foundry.com/licensing

Installing a Subscription License

Subscription licensing differs from traditional node locked or floating licenses in that a single license, or entitlement, is valid on any authorized device up to the entitlement's maximum number of activations.

• A **Entitlement** represents the right to run a Foundry product for a set amount of time on a set number of devices.

• An **Authorized Device** is a recognized device, such as a desktop computer, on which entitlements can be activated.

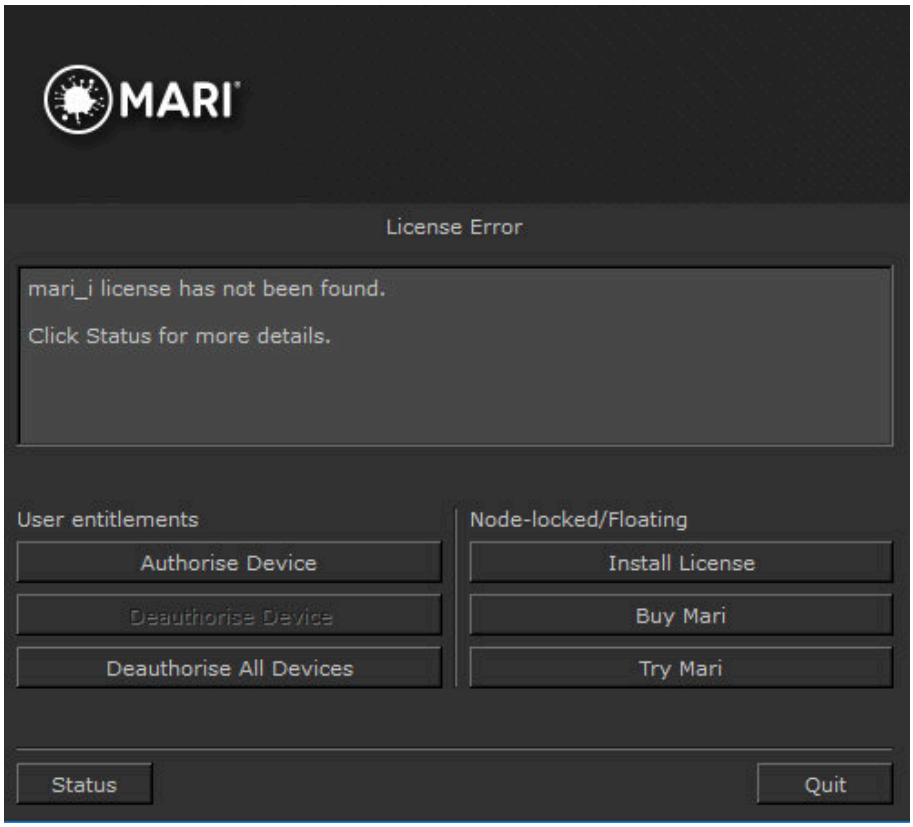
For example, if a **Entitlement** for Mari has two activations, you can use Mari on two separate **Authorized Devices** simultaneously. If you want to activate another device, you have to deactivate an existing one, but you can activate and deactivate devices as often as you like.

Subscription licenses require that you connect to the Internet at least every 30 days to extend your activation.

To authorize your device, follow these steps:

1. Create a Foundry account using a valid e-mail address on our website at https://www.foundry.com/
2. Launch Mari using any of the methods described in .

   The **Licensing** dialog displays.



3. Click **Authorise Device**.
4. Enter your account e-mail address and password and then click **Authorise Device**.

   A subscription license is created in your home directory:
   C:\Users<username>**\FoundryLicensing**<SystemID>

> **Note:** Replace <username> and <SystemID> with the current user and the MAC address of the device, respectively.

   The license looks similar to this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**

5. Once the license is installed, click **Launch** to start using Mari.

> **Note:** On Windows, there is a known issue with user names containing non-ASCII characters causing licensing to fail. If a licensing error similar to the following displays:
> **Unable to create subscription license directory: C:\Users\Zoë Hernández\FoundryLicensing**
> Try changing the license directory to an alternate location using the **FN_SUBSCRIPTION_LICENSE_DIR** environment variable. See Setting Environment Variables for more information.

6. If you need to deactivate an entitlement or deauthorize a device, navigate to **Help** > **License** and, click:
   - **Deactivate Mari** to reclaim one of your entitlements,

- **Deauthorize Device** to reclaim your existing Foundry entitlements on this device and stop additional ones running, or
- **Deauthorize All Devices** to reclaim your existing Foundry entitlements on all devices associated with your account, and stop additional ones running.

Licensing on a Single Machine

Obtain an Activation Key

To obtain a license, you need your machine's System ID (sometimes called Host ID or rlmhostid). Just so you know what a System ID number looks like, here's an example: 000ea641d7a1.

> **Note:** Bear in mind that, for floating licenses, you need the System ID of the license server, not the machines on which you intend to run Mari.

There are a number of ways you can find out your machine's System ID:

- Launch Mari without a license, click **Status**, and then scroll down the error report until you see your System ID.
- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Your System ID is displayed.
- Download the Foundry Licensing Tools (FLT) free of charge from https://www.foundry.com/licensing and then run C:\Program Files\TheFoundry\LicensingTools7.0\ Foundry License Utility.exe

When you know your System ID, you can request a license for Foundry products:

- from Foundry's Sales Department at sales@foundry.com
- from the product pages on our website, such as https://www.foundry.com/products/mari
- by launching Mari without a license and selecting:
    - **Buy Mari** - opens a web browser directly to Foundry website to purchase a license.
    - **Try Mari** - displays the 15-day trial license download screen. Enter your Foundry account details or create a new account and follow the on-screen instructions to receive a trial license.

Note: By default, if you have installed a temporary license, Mari displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior, you can set the FN_DISABLE_LICENSE_DIALOG environment variable to 1 to suppress the warning message about imminent license expiration. See *Environment Variables That Mari Recognizes* in the *Mari User Guide* for more information.

Install the License

You are prompted to enter your activation key when you start Mari without a license. Mari skips this step if you've already activated the application.

When you start the application before installing a license, a **Licensing** dialog displays an error, informing you that no license was available. The installation process is dependent on what type of license you requested:

- **License file** - if you requested a license file, typically **foundry.lic**, this option allows you to browse to the file location and install it automatically. See Licensing on Windows for more information.
- **Activation Key or license text** - if you requested an Activation Key or license by e-mail, this option allows you to paste the key or license text into the **Licensing** dialog, which then installs the license in the correct directory. See To install an Activation Key or license text for more information.
- **A floating license** - if you requested a floating license to supply licenses to multiple client machines, this option allows you enter the server address that supplies the client licenses.

Note: You must install a floating license and additional software on the license server to use this option.

See Install Floating Licenses for more information.

Tip: If you later want to check the current status of your license (for example, to see if it's a node locked license or a floating license), select **Help** > **License** in Mari to display the **Mari Licensing** dialog.

To install a license from disk

1. Save the license file to a known location on disk.
2. Launch Mari.

   The Licensing dialog displays.
3. Click **Install License** to display the available license installation options.

4. Click **Install from Disk**.
5. Browse to the location of the license file.
6. Click **Open** to install the license automatically in the correct directory.

   The **Licensing** dialog opens showing you that the license is installed.
7. Click **Launch** to start Mari.

To install an Activation Key or license text

1. Launch Mari.

   The Licensing dialog displays.
2. Click **Install License** to display the available license installation options.
3. Click **Activation Key / License Text** and then either:
   - Enter the **Activation Key** string in place of **Insert Activation Key Here**. A license key typically looks something like this:
     `mari-0101-77d3-99bd-a977-93e9-8035`

   OR
   - Copy the license text and paste it over the **Copy/Paste license text here** string. License text typically looks something like this:
     ```
     LICENSE foundry mari_i 2026.1223 permanent 2 share=h min_timeout=30
     start=23-jan-2026 issuer=sf issued=23-jan-2026 replace
     _ck=6dd78e4c69 sig="60PG452MPDMMM6MJAMRGKNQAN3PEAK8JYTHN45022M0C98H
     XFA9N7SAASMRABT8TUJ6FAMX8TWU0"
     ```
4. Click **Install**.

   The license is automatically installed on your machine in the correct directory.

> **Note:** Activation Keys require an internet connection. If you access the internet through a proxy server and cannot connect to the activation server, you may get an error dialog prompting you to either:
>
> -Click **Use Proxy** to enter the proxy server name, port number, username, and password. This enables the application to connect to the activation server and obtain a license. The license is then installed automatically, or
> -Click on the web link in the dialog and use the System ID (also known as hostid) provided to manually activate and install a license.

5. Click **Launch** to start Mari.

Licensing over a Network

Obtain Floating Licenses

Alternatively, you can purchase a floating license key from our website.

1. Display your system ID by doing one of the following:
   - Launch Mari without a license, click **Status**, and then scroll down the error report until you see your System ID.
   - Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Your System ID is displayed.
   - Download the Foundry Licensing Tools (FLT) free of charge from https://www.foundry.com/licensing and then run C:\Program Files\TheFoundry\LicensingTools7.0\ Foundry License Utility.exe
2. Once you have provided us with your System ID number and a license key has been generated for you, you'll receive the license key in an e-mail or Internet download. The license key is contained in a text file called foundry.lic. For information on what to do with the **foundry.lic** file, see Licensing on Windows.

Install Floating Licenses

If you requested a floating license from Foundry, you will receive your license key (foundry.lic) in an e-mail or Internet download. You should also receive the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the *client* machines on your network.

Note: The FLU is also available to download from https://www.foundry.com/licensing

1. Make sure you have saved both the license key (foundry.lic) and the Foundry License Utility application in the same directory.
2. Run the FLU application.

   The license key automatically appears in the FLU window if the FLU and **foundry.lic** are in the same directory.

Tip: If they are not in the same directory, you can either copy and paste the contents of the license key or drag-and-drop the file into the FLU window.

3. Click **Install**.

This checks the license file and, provided that the license is valid, installs it into the correct directory.

4. In order for the floating license to work, you will need to install the Foundry Licensing Tools (FLT) on the license server machine.

   For more information on how to install floating licenses, refer to the *FLT User Guide*, which you can download from our website https://www.foundry.com/licensing

5. Once your license server is up and running, launch Mari on the client machine.
6. Click **Use Server** and enter the server address in the field provided. The format for the server name is: <port>@<servername>, for example, 30001@red.

   The **Licensing** dialog opens showing you that the license is installed.

7. Click **Launch** to start Mari.

Note: You must perform steps 5 and 6 on each client machine that requires a Mari license from the server.

Tip: If you later need to display the **Mari Licensing** dialog again, you can select **Help** > **License** in Mari.

Further Reading

For more information on licensing Mari, displaying the System ID number, setting up a floating license server, adding new license keys and managing license usage across a network, you should read the *Foundry Licensing Tools User Guide* available on our website at: https://www.foundry.com/licensing.

Uninstalling on Windows

To uninstall Mari on Windows, there are a few things you need to do:

1. Navigate to **C:\Program Files\Mari8.0v1** and double-click **Uninstall**.

   The **Mari Uninstall** dialog displays.

2. Click **Yes** to uninstall the application files.

Tip: You can also uninstall Mari silently, with no prompts or popups, from the Command Prompt or PowerShell by entering:
`"C:\Program Files\Mari8.0v1\Uninstall.exe" /S`

3. Delete, rename, or move your **.mari** folder.

   The **.mari** folder contains customized files such as preferences, workspaces, and so on.

The **.mari** folder is usually found under the directory pointed to by the **HOME** environment variable. If this variable is not set, which is common, the **.mari** directory is under the folder specified by the **USERPROFILE** environment variable, which is generally one of the following:

• drive letter:**\Documents and Settings**login name****

• drive letter:**\Users**<USERNAME>****

To find out if the **HOME** and **USERPROFILE** environment variables are set and where they are pointing at, enter **%HOME%** or **%USERPROFILE%** into the address bar in Windows Explorer. If the environment variable is set, the folder it's pointing at is opened. If it's not set, you get an error.

4. Delete, rename, or move your **Mari** folder by navigating to drive letter:**\Users**<USERNAME>**\Documents**.

Note: Note: This folder is an output folder and doesn't affect the stability of Mari.

5. Delete, rename, or move your Mari projects folder that contains your cached files. This folder resides in the location you picked when you created your first project.

Deleting Mari-related environment variables

To delete any Mari-related environment variables, do the following:

1. Click on **Start**.
2. Select **Computer**.
3. Click **System Properties**.
4. Click the **Advanced system settings** tab.
5. Click the **Environment Variables...** button.

   The **Environment Variables** dialog opens.
6. Select any Mari-related environment variables and click **Delete**.

Article: Take a look at the Uninstalling Mari article for more information.

Installing on Linux

For a full list of minimum and recommended hardware specifications and tested GPUs, see the Mari technical specification on our website: https://www.foundry.com/products/mari/requirements

Mari 8.0 is available to download from our website at https://www.foundry.com/products/mari. The download is in a compressed **.run** format.

1. Download the **.run** installation file from our web site.
2. Extract Mari from the **.run** archive with the following terminal command, replacing <version number> with the current version:

   ```
   sudo ./Mari<version number>-linux-x86-release-64.run
   ```

   The installer displays the End User Licensing Agreement (EULA) and prompts you to accept it.
3. If you agree with the EULA, enter **y** and press **Return** to accept the EULA. (If you don't agree with the EULA and press **n** instead, the installation is canceled.)

   By default, Mari is installed in an appropriately named folder in the current working directory.

Note: If you leave out **sudo** from the terminal command, you need to ensure that you have sufficient
permissions to install Mari under your current working directory or selected target directory.

After the Mari application files have been installed, the installer also runs a post-installation script that creates the following directory: /usr/local/foundry/RLM

If you don't have sufficient permissions on the **/usr/local** folder for this directory to be created, the post-installation script prompts you for your sudo password as necessary.

Tip: You can also use the following options after the terminal command when installing Mari:
--info
This lets you see what the default installation directory is.
--prefix-dir MyMariDirectory
This lets you specify a different directory to install Mari to (in this case, MyMariDirectory).
--accept-eula
This lets you automate the installation so that you are not prompted to accept the EULA. Note that if you use this option, you agree to the terms of the EULA. To see the EULA, please refer to End User License Agreement.
-- help
This lets you see additional help and installer options.
Here's an example of the syntax using the **--accept-eula** option: sudo ./Mari<version number>-linux-x86-release-64.run --accept-eula

Launching on Linux

Article: Take a look at the Launching Mari in safe mode article.

1. Open a terminal.
2. Navigate to the directory you installed Mari to.
3. Enter **./mari**
4. If you haven't installed a license for Mari, you need to choose from one of the options for . Once you've done so, click **Launch** in the **Mari Licensing** dialog.

   Mari checks for the most frequently used graphics cards, and if a graphics card configuration is detected that's not in the tested GPU hardware list, a dialog displays warning you about it. Depending on the configuration detected, you have the option to **Quit** or **Continue** and, in the case of configurations with known issues, also **Ignore Permanently**.

Note: If you ignore the warning and run Mari anyway, you may experience instability or performance issues. If a tested graphics card configuration is detected, Mari runs as normal.

5. If you are launching Mari for the first time, Mari selects a default cache directory for its project files. You can change this directory and select one of your own. In the **Confirm Projects Directory** dialog, navigate to the directory you want to use and click **Choose**.

   The Mari graphical interface displays.

   Please note that the project directory should be:

- empty
- local to the machine (not a network mount)
- as fast as possible
- a location that's not temporary (to avoid the data disappearing)
- a directory that the user has read and write permissions to.

Project cache locations are directories that Mari uses to store project data files. These directories must be persistent; not in temporary locations. If more than one project location is specified, Mari spreads each project's data files across all locations. Once a project has been created for a given set of project cache locations, that set must not be changed. In order to move a project between different sets of project caches, archive the project.

Warning: Don't add, remove, or amend project locations once they have been set unless they are empty. Once projects have been created for a given set of project cache locations, that set must not be changed or project corruption will occur.

Note: If the MARI_CACHE environment variable has been set, its value is used to determine the location of the project cache files. In this case, Mari does not prompt you to select the project directory. For more information on environment variables that Mari understands, please refer to Environment Variables That Mari Recognizes.

Article: Take a look at the Temporarily setting the project directory in Mari article.

Tip: You can also launch Mari in verbose mode by using the following command:
./mari --verbose

This way, Mari provides a running log of each action that it performs. You can see this log in a terminal window.

6. If you are upgrading, Mari prompts you if you'd like to import your user preferences from your previous installed version of Mari.

You can run the application in one of two safe modes: `--safe` and `--safer`. The `--safe` mode:

- Disables startup Python scripts
- Disables custom user plug-ins
- Bypasses user Python libraries.

The `--safer` mode performs all of the above options as well as:

- Bypasses user settings.
- Bypasses user environment variables.

These safe modes are primarily used for debugging purposes. For advanced debugging methods, please see *Appendix B: Frequently Asked Questions and Troubleshooting* in the *Mari User Guide* for more information.

Mari Non-Commercial

If you want to try out or learn Mari, you can run Mari Non-commercial. This version allows you to explore most of Mari's features, but prevents the commercial use of the application. For more information, see Mari Non-Commercial.

To launch the application in Non-Commercial mode, navigate to the Mari application directory and enter:
`./mari --nc`

> **Note:** The default install location on Linux is:
> `/usr/local/Mari<version>`

If you have already activated Mari Non-commercial on the current device, the graphical interface appears, and a command line window opens. If you haven't activated the device yet, proceed to the next section.

Licensing on Linux

If you simply want to try out Mari, you can obtain a trial license, which allows you to run Mari for free for 15 days. See Install the License.

To use Mari after this trial period, you need either a valid **subscription license**, **activation key**, **node locked license**, or a **floating license** and server running the Foundry Licensing Tools (FLT):

- **Subscription Licenses** - allows you to activate a single license or entitlement on any authorized device up to the entitlement's maximum number of activations.
- **Activation Keys** - activation keys allow you to activate and generate your actual product license key, at a later point after purchase, on the machine for which you require the license. They are provided for both node locked and floating license, and generate the appropriate license type once installed using the product's Licensing dialog or online using the **Activate a Product** page: https://www.foundry.com/user/login?destination=/licensing/activate-product
- **Node Locked Licenses** - these can be used to license an application on a single machine. They do not work on different machines and if you need them to, you'll have to transfer your license.

Node locked licenses, sometimes called uncounted licenses, do not require additional licensing software to be installed.

- **Floating Licenses** - these are used to license applications on any networked client machine. The floating license is put on the server and is locked to a unique number on that server.

  Floating licenses, sometimes called counted licenses, require additional software to be installed on a server to manage the licenses and give them out to the client stations that want them. This software is called the Foundry Licensing Tools (FLT) and can be downloaded at no extra cost from our website.

The instructions below run through both licensing methods. You can also find a more detailed description in the *Foundry Licensing Tools User Guide* available on our website: https://www.foundry.com/licensing

Installing a Subscription License

Subscription licensing differs from traditional node locked or floating licenses in that a single license, or entitlement, is valid on any authorized device up to the entitlement's maximum number of activations.

• A **Entitlement** represents the right to run a Foundry product for a set amount of time on a set number of devices.

• An **Authorized Device** is a recognized device, such as a desktop computer, on which entitlements can be activated.

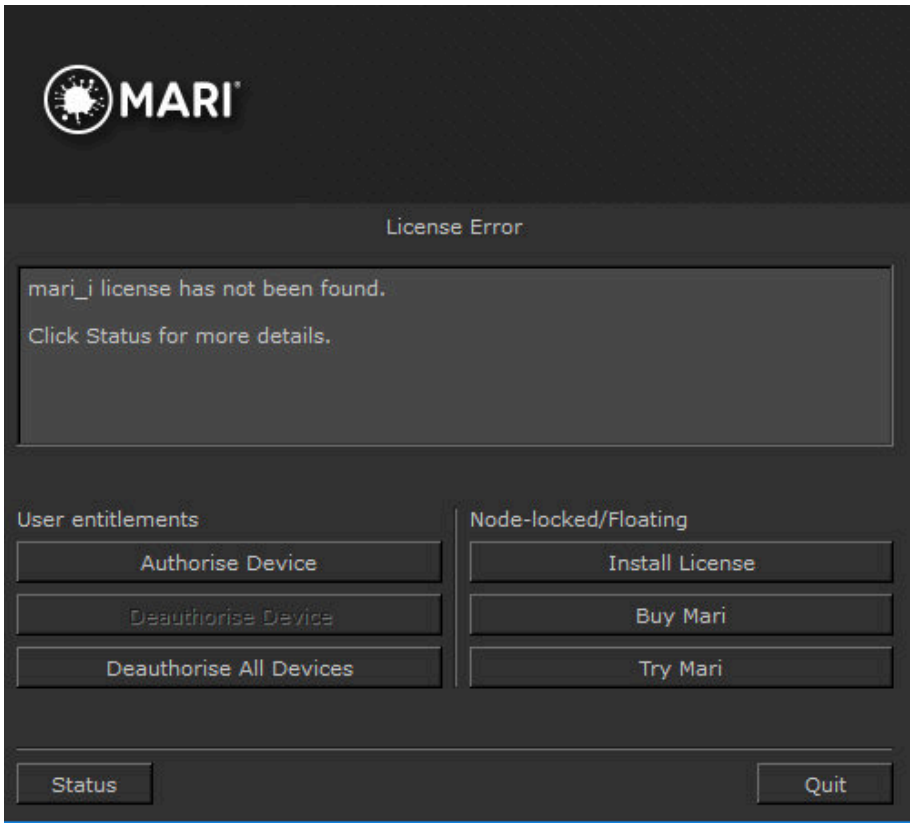
For example, if a **Entitlement** for Mari has two activations, you can use Mari on two separate **Authorized Devices** simultaneously. If you want to activate another device, you have to deactivate an existing one, but you can activate and deactivate devices as often as you like.

Subscription licenses require that you connect to the Internet at least every 30 days to extend your activation.

To authorize your device, follow these steps:

1. Create a Foundry account using a valid e-mail address on our website at https://www.foundry.com/
2. Launch Mari using any of the methods described in Installing on Linux.

   The **Licensing** dialog displays.



3. Click **Authorise Device**.
4. Enter your account e-mail address and password and then click **Authorise Device**.

   A subscription license is created in your home directory:
   /home/<username>**/FoundryLicensing/**<SystemID>

> **Note:** Replace <username> and <SystemID> with the current user and the MAC address of the device, respectively.

   The license looks similar to this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**

5. Once the license is installed, click **Launch** to start using Mari.
6. If you need to deactivate an entitlement or deauthorize a device, navigate to **Help** > **License** and, click:
   - **Deactivate Mari** to reclaim one of your entitlements,
   - **Deauthorize Device** to reclaim your existing Foundry entitlements on this device and stop additional ones running, or
   - **Deauthorize All Devices** to reclaim your existing Foundry entitlements on all devices associated with your account, and stop additional ones running.

Licensing on a Single Machine

Obtain an Activation Key

To obtain a license, you need your machine's System ID (sometimes called Host ID or rlmhostid). Just so you know what a System ID number looks like, here's an example: 000ea641d7a1.

> **Note:** Bear in mind that, for floating licenses, you need the System ID of the license server, not the machines on which you intend to run Mari.

There are a number of ways you can find out your machine's System ID:

- Launch Mari without a license, click **Status**, and then scroll down the error report until you see your System ID.
- Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Your System ID is displayed.
- Download the Foundry Licensing Tools (FLT) free of charge from https://www.foundry.com/licensing and then run the following command in a terminal shell:

```
/usr/local/foundry/LicensingTools7.0/bin/systemid
```

When you know your System ID, you can request a license for Foundry products:

- from Foundry's Sales Department at sales@foundry.com
- from the product pages on our web site, such as https://www.foundry.com/products/mari
- by launching Mari without a license and selecting:
  - **Buy Mari** - opens a web browser directly to Foundry's website to purchase a license.
  - **Try Mari** - displays the 15-day trial license download screen. Enter your Foundry account details or create a new account and follow the on-screen instructions to receive a trial license.

> **Note:** By default, if you have installed a temporary license, Mari displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior, you can set the FN_DISABLE_LICENSE_DIALOG environment variable to 1 to suppress the warning message about imminent license expiration. See *Environment Variables That Mari Recognizes* in the *Mari User Guide* for more information.

Install the License

You are prompted to enter your activation key when you start Mari without a license. Mari skips this step if you've already activated the application.

When you start the application before installing a license, a **Licensing** dialog displays an error, informing you that no license was available. The installation process is dependent on what type of license you requested:

- **License file** - if you requested a license file, typically **foundry.lic**, this option allows you to browse to the file location and install it automatically. See Licensing on Linux for more information.
- **Activation Key or license text** - if you requested an Activation Key or license by e-mail, this option allows you to paste the key or license text into the **Licensing** dialog, which then installs the license in the correct directory. See Licensing on Linux for more information.
- **A floating license** - if you requested a floating license to supply licenses to multiple client machines, this option allows you enter the server address that supplies the client licenses.

Note: You must install a floating license and additional software on the license server to use this option.

See Licensing on Linux for more information.

Tip: If you later want to check the current status of your license (for example, to see if it's a node locked license or a floating license), select **Help** > **License** in Mari to display the **Mari Licensing** dialog.

To install a license from disk

1. Save the license file to a known location on disk.
2. Launch Mari.

   The Licensing dialog displays.
3. Click **Install License** to display the available license installation options.
4. Click **Install from Disk**.
5. Browse to the location of the license file.
6. Click **Open** to install the license automatically in the correct directory.

   The **Licensing** dialog opens showing you that the license is installed.
7. Click **Launch** to start Mari.

To install an Activation Key or license text

1. Launch Mari.

   The Licensing dialog displays.
2. Click **Install License** to display the available license installation options.
3. Click **Activation Key / License Text** and then either:
   - Enter the **Activation Key** string in place of **Insert Activation Key Here**. A license key typically looks something like this:
     `mari-0101-77d3-99bd-a977-93e9-8035`

   OR
   - Copy the license text and paste it over the **Copy/Paste license text here** string. License text typically looks something like this:
     ```
     LICENSE foundry mari_i 2026.1223 permanent 2 share=h min_timeout=30
     start=23-jan-2026 issuer=sf issued=23-jan-2026 replace
     _ck=6dd78e4c69 sig="60PG452MPDMMM6MJAMRGKNQAN3PEAK8JYTHN45022M0C98H
     XFA9N7SAASMRABT8TUJ6FAMX8TWU0"
     ```
4. Click **Install**.

   The license is automatically installed on your machine in the correct directory.

> **Note:** Activation Keys require an internet connection. If you access the internet through a proxy server and cannot connect to the activation server, you may get an error dialog prompting you to either:
>
> -Click **Use Proxy** to enter the proxy server name, port number, username, and password. This enables the application to connect to the activation server and obtain a license. The license is then installed automatically, or
> -Click on the web link in the dialog and use the System ID (also known as hostid) provided to manually activate and install a license.

5. Click **Launch** to start Mari.

Licensing over a Network

Obtain Floating Licenses

Alternatively, you can purchase a floating license key from our website.

1. Display your system ID by doing one of the following:
   - Launch Mari without a license, click **Status**, and then scroll down the error report until you see your System ID.
   - Download the Foundry License Utility (FLU) from https://www.foundry.com/licensing and run it. Your System ID is displayed.
   - Download the Foundry Licensing Tools (FLT) free of charge from https://www.foundry.com/licensing and then run C:\Program Files\TheFoundry\LicensingTools7.0\ Foundry License Utility.exe
2. Once you have provided us with your System ID number and a license key has been generated for you, you'll receive the license key in an e-mail or Internet download. The license key is contained in a text file called foundry.lic. For information on what to do with the foundry.lic file, see Licensing on Linux.

Install Floating Licenses

If you requested a floating license from Foundry, you will receive your license key (foundry.lic) in an e-mail or internet download. You should also receive the Foundry License Utility (FLU) application to help you install the license key on the license server machine. The server manages licenses for the *client* machines on your network.

Note: The FLU is also available to download from https://www.foundry.com/licensing

1. Make sure you have saved both the license key (foundry.lic) and the Foundry License Utility application in the same directory.
2. Run the FLU application.

   The license key automatically appears in the FLU window if the FLU and **foundry.lic** are in the same directory.

Tip: If they are not in the same directory, you can either copy and paste the contents of the license key or drag-and-drop the file into the FLU window.

3. Click **Install**.

   This checks the license file and, provided that the license is valid, installs it into the correct directory.
4. In order for the floating license to work, you need to install the Foundry Licensing Tools (FLT) on the license server machine.

   For more information on how to install floating licenses, refer to the *FLT User Guide*, which you can download from our website https://www.foundry.com/licensing
5. Once your license server is up and running, launch Mari on the client machine.

6. Click **Use Server** and enter the server address in the field provided. The format for the server name is: <port>@<servername>, for example, 30001@red.

   The **Licensing** dialog opens showing you that the license is installed.
7. Click **Launch** to start Mari.

> **Note:** You must perform steps 5 and 6 on each client machine that requires a Mari license from the server.

> **Tip:** If you later need to display the **Mari Licensing** dialog again, you can select **Help** > **License** in Mari.

Further Reading

For more information on licensing Mari, displaying the System ID number, setting up a floating license server, adding new license keys and managing license usage across a network, you should read the *Foundry Licensing Tools User Guide* available on our website at: https://www.foundry.com/licensing.

Uninstalling on Linux

To uninstall Mari on Linux, there are a few things you need to do:

> **Note:** Make sure you have the right permissions to uninstall an application on your Linux system.

8. Navigate to where you installed Mari to delete the application folder of the required version of Mari.
9. Delete, rename, or move the **.config/TheFoundry/** folder located in your home directory. You can use the **$HOME** environment variable to check the path to your home directory.
10. Delete, rename, or move your **Mari** folder by navigating to: **/home/**<USERNAME>.

> **Note:** This folder is an output folder and doesn't affect the stability of Mari.

11. Delete, rename, or move your **Mari** projects folder that contains your cached files. This folder resides in the location you picked when you created your first project.
12. Delete your Mari environment variables.

Article: The procedures on how to delete environment variables on a Linux operating system vary depending on which version of Linux you are using.
For more information, see How to delete/unset environment variables.

Article: Take a look at the Uninstalling Mari article for more information.

Mari Non-Commercial

Mari Non-commercial is a free version of Mari that runs outside the regular licensing model. Mari Non-commercial is meant for personal, educational, and other non-commercial use. It is aimed at students, industry professionals, and others interested in Mari. It includes most of the features of the commercial version of Mari, offering you a chance to explore and learn the application fully while using it from the comfort of your own home.

Mari Non-commercial is designed for non-commercial use only and does differ from the commercial version in some aspects. Here are the differences:

- Mari Non-commercial projects (**.mnc**), can only be created and shared in Mari Non-commercial.
- Projects created in commercial versions of Mari (**.mra**) cannot be opened in Mari Non-commercial.
- The output formats **.tif**, **.tiff**, **.hdr**, **.dds**, and **.ptx** are not available.
- The patch count is limited to 6 patches, with a maximum face size of 4k, but unlimited channels and layers.
- The object count is limited to 5 objects.
- The export texture resolution size has been limited to 4k.
- The export texture bit-depth is limited to 8- and 16-bit.
- The Custom Shader API has been disabled.
- Session Scripts have been disabled.
- The Nuke<>Mari bridge has been disabled.
- The use of OCIO files is limited to Nuke's default OCIO file.
- The Python Console has been disabled.
- Python user scripts cannot be run at startup.
- Python example scripts are not accessible from Mari Non-commercial.
- Support for user C API plug-ins has been disabled.

In other respects, Mari Non-commercial contains all the functionality of the commercial version of Mari.

Support

If you experience any problems and need help, the forums are an excellent resource for finding help and learning tips from other users.

Licensing Mari Non-commercial on Windows

Subscription licensing differs from traditional node locked or floating licenses in that a single license, or entitlement, is valid on any authorized device up to the entitlement's maximum number of activations.

- An **Entitlement** represents the right to run a Foundry product for a set amount of time on a set number of devices.
- An **Authorized Device** is a recognized device, such as a desktop computer, on which entitlements can be activated.

For example, if an **Entitlement** for Mari has two activations, you can use Mari on two separate **Authorized Devices** simultaneously. If you want to activate another device, you have to deactivate an existing one, but you can activate and deactivate devices as often as you like.

To get started with Mari Non-commercial, follow these steps:

1. Create a Foundry account using a valid e-mail address on our website, here.
2. Launch Mari in non-commercial mode as described under Installing on Windows.

   A **Licensing** dialog displays, similar to regular licensing. Mari Non-commercial is free, but your entitlement only contains two activations.
3. Click **Authorise Device**.
4. Enter your account e-mail address and password and then click **Authorise Device**.
5. A subscription license is created in your home directory:
   C:\Users<username>**\FoundryLicensing**<SystemID>

> **Note:** Replace **<username>** and <SystemID> with the current user and the MAC address of the device, respectively.

   The license looks something like this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**
6. Once the license is installed, click **Launch** to start using Mari.

Note: On Windows, there is a known issue with user names containing non-ASCII characters causing licensing to fail. If a licensing error similar to the following displays:
Unable to create subscription license directory: C:\Users\Zoë Hernández\FoundryLicensing
Try changing the license directory to an alternate location using the **FN_SUBSCRIPTION_LICENSE_DIR** environment variable. Refer to the *Mari User Guide* in the *Setting Environment Variables* section for more information. for more information.

7. If you need to deactivate an entitlement or deauthorize a device, navigate to **Help** > **License** and, click:
   - **Deactivate Mari** to reclaim one of your entitlements,
   - **Deauthorize Device** to reclaim your existing Foundry entitlements on this device and stop additional ones running, or
   - **Deauthorize All Devices** to reclaim your existing Foundry entitlements on all devices associated with your account, and stop additional ones running.

Licensing Mari Non-commercial on Linux

Subscription licensing differs from traditional node locked or floating licenses in that a single license, or entitlement, is valid on any authorized device up to the entitlement's maximum number of activations.

- An **Entitlement** represents the right to run a Foundry product for a set amount of time on a set number of devices.
- An **Authorized Device** is a recognized device, such as a desktop computer, on which entitlements can be activated.

For example, if an **Entitlement** for Mari has two activations, you can use Mari on two separate **Authorized Devices** simultaneously. If you want to activate another device, you have to deactivate an existing one, but you can activate and deactivate devices as often as you like.

To get started with Mari Non-commercial, follow these steps:

1. Create a Foundry account using a valid e-mail address on our website, here.
2. Launch Mari in non-commercial mode as described under Installing on Windows.

   A **Licensing** dialog displays, similar to regular licensing. Mari Non-commercial is free, but your entitlement only contains two activations.
3. Click **Authorise Device**.
4. Enter your account e-mail address and password and then click **Authorise Device**.
5. A subscription license is created in your home directory:
   /home/<username>**/FoundryLicensing/**<SystemID>

Note: Replace <username> and <SystemID> with the current user and the MAC address of the device, respectively.

The license looks something like this: **c58edf7e-17ab-435b-8d8a-b3a9b347ab11.lic**

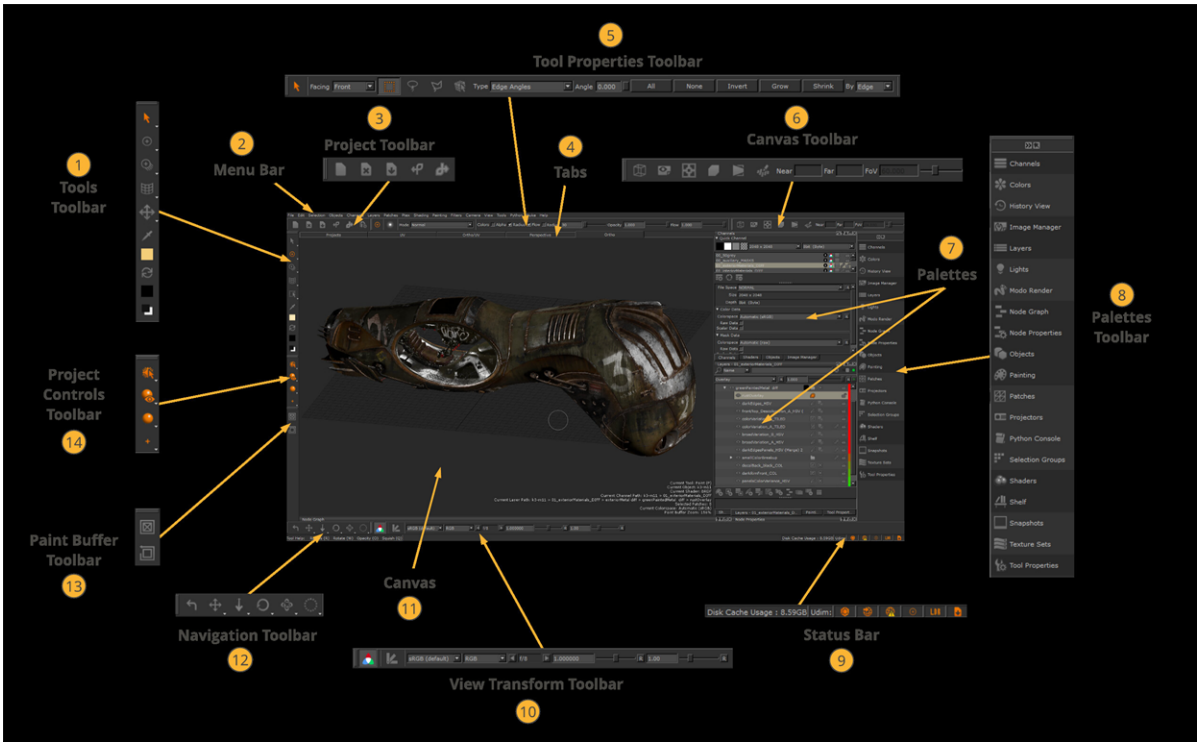
6. Once the license is installed, click **Launch** to start using Mari.
7. If you need to deactivate an entitlement or deauthorize a device, navigate to **Help** > **License** and, click:
   - **Deactivate Mari** to reclaim one of your entitlements,
   - **Deauthorize Device** to reclaim your existing Foundry entitlements on this device and stop additional ones running, or
   - **Deauthorize All Devices** to reclaim your existing Foundry entitlements on all devices associated with your account, and stop additional ones running.

Discover the Workspace

The Mari workspace consists of a menu bar, a canvas, toolbars, palettes, and a status bar. Palettes are used for working with items such as channels or shaders. They can be either docked in the application or made to float wherever you find them most useful. You can also place toolbars at the top or bottom and on the left- or right-hand side of the canvas.

What it Looks Like

Here is an illustration of the default layout of the Mari workspace that displays with an open project.



1	**Tools Toolbar**	The **Tools** toolbar selects a tool for painting. See the Toolbars.
2	**Menu Bar**	The menu bar selects Mari commands, many of which are also available using either shortcut keys or icons.
3	**Project Toolbar**	The **Project** toolbar performs common project functions, such as creating, closing, and saving projects. See the Toolbars.

4	**Tabs**	The tabs switch between the following views: **Projects**, **UV**, **Ortho/UV**, **Perspective**, and **Ortho**.
5	**Tool Properties Toolbar**	The **Tool Properties** toolbar displays information about the selected tool. See the Toolbars. **Note:** The illustration above shows the **Select** tool's **Tool Properties** toolbar as the **Select** tool is selected in the **Tools** toolbar.
6	**Canvas Toolbar**	The **Canvas** toolbar sets the view options for the UV and 3D views.
7	**Palettes**	The palettes contain controls for viewing and changing different aspects of the geometry on the canvas.
8	**Palettes Toolbar**	The **Palettes** toolbar allows quick access to your Mari palettes. See the Configuring Your Workspace.
9	**Status Bar**	The **Status** bar displays information about the project generally, progress of long operations (such as baking), icons, and basic tool help for the current tool. It also displays how much of your cache disk has been filled. See the Status Bar.
10	**View Transform Toolbar**	The **View Transform** toolbar displays a number of options for managing your monitor's colorspace in Mari. See View Transform Toolbar.
11	**Canvas**	This is where you view and paint your geometry.
12	**Navigation Toolbar**	The **Navigation** toolbar displays a number of options for managing navigation in Mari. See the Toolbars.
13	**Paint Buffer Toolbar**	The **Paint Buffer** toolbar clears all un-baked paint and reset the paint buffer transform. See the Toolbars.
14	**Project Controls Toolbar**	The **Project Controls** toolbar displays a set of four tools: • **Selection** - lets you switch between three selection modes. • **Default Shaders** - displays one of the four default shaders in the **Shaders** palette and the last-selected user shader. • **Lighting** - lets you change the lighting on the object.

		• **Paint Buffer Symmetry** - displays the four **Mirror Painting** modes for the Paint Buffer Symmetry functionality. • **Mirror Projection** - displays the four **Mirror Projecting** modes for the Mirror Projection functionality. See Mirror Projection. See the Toolbars.

Mari 3.3 vs 4.0

Mari's UI has been redesigned for better usability. Below is a list briefly describing the changes. Click on the links for more details.

Article: Read Q100355: Transitioning between Mari 3.3 and Mari 4.0 for more information on the changes done, including a video tutorial.

Creating a Project

- **New Project** dialog - **The New Project** dialog has been redesigned and now includes shading and lighting modes.

Controlling the Project

- The **Selection**, **Default Shaders**, **Lighting**, and **Mirroring** toolbars have been removed in 4.0. They have been added to the new **Project Controls** toolbar.

Using Tools

- All of Mari's tools are now grouped into categories, in the **Tools** toolbar, to reduce the space they use in the UI.
- The **Tool Properties** toolbars have been redesigned. They now include a brush splat icon and fewer button labels.
- The **Ptex** and **Vector Painting** toolbars are hidden in the default layout. They can be made visible using the toolbar right-click menu.

Painting in Mari

- The **Painting** toolbar has been renamed to the **Paint Buffer** toolbar. An additional button for resetting the translation of the paint buffer has been added to the toolbar.
- The **Projection** palette has been merged with the **Painting** palette.
- You can now access all Mari's palettes through the new **Palettes** toolbar.
- The **Paint Target** quick palette has been extended to include the paint target when a mask is present.
- The **Layers** palette is now more intuitive when working with Group layers.

Editing Brushes

- The **Brush Editor** palette has been removed. The **Brush Properties** are now accessible through the **Tool Properties** palette. Instead of using the **Brush Editor** palette to save your custom brushes, you can now save them by dragging-and-dropping the brush splat from the **Tool Properties** toolbar of any painting tool.

Working with Colors

- The **Colors** palette is now scalable for better precision. The **Pixel Analyzer** palette has been merged with the **Colors** palette. The **Color Manager (Viewer Transform)** palette has been removed and you can still find the Colorspace settings in the **Project Settings** dialog.
- Using the **Marquee Select** tool, you can now drag a color to the canvas to fill the selected area of the paint buffer.
- You can now drag-and-drop colors to fill the current item selection. The fill works across all selection modes (**Object**, **Patches**, and **Faces**).
- The curve editor has been redesigned entirely.
- The **Source Grade** feature has been extended to match the source grading controls available in the Grade Adjustment layer.

Using the Node Graph

- The basic Node Graph mode has been removed and the Advanced mode is now standard. This also unlocks the full Node Graph in Mari Non-Commercial.
- In the **Node Properties** palette, property groups are now displayed in separate tabs for better usability.

Subdividing Objects

- The latest features from OpenSubdiv 3.1, including scheme choice, geometry, and UV boundary interpolation methods have been added. Mari now matches renderer mesh subdivision more closely, increasing efficiency when painting.

Interface

- Some preferences in the **Mari Preferences** dialog have been reorganized and renamed.
- The Head-Up Display now displays in the bottom-right corner of the canvas instead of the top-left corner.

Exporting Channels

- A new **Export Manager** dialog for managing the batch exporting of channels and Bake Point nodes has been implemented. You can now configure and manage multiple export targets from the same source, as well as perform format conversions during export.

Note: Have a look at the new Mari 4.0 workspace. See Discover the Workspace.

Creating a Project

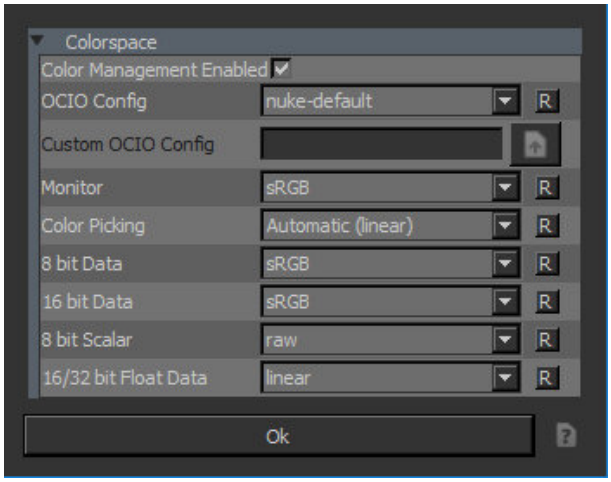
New Project Dialog

Video: Watch Mari 4.0 | Creating a Project.

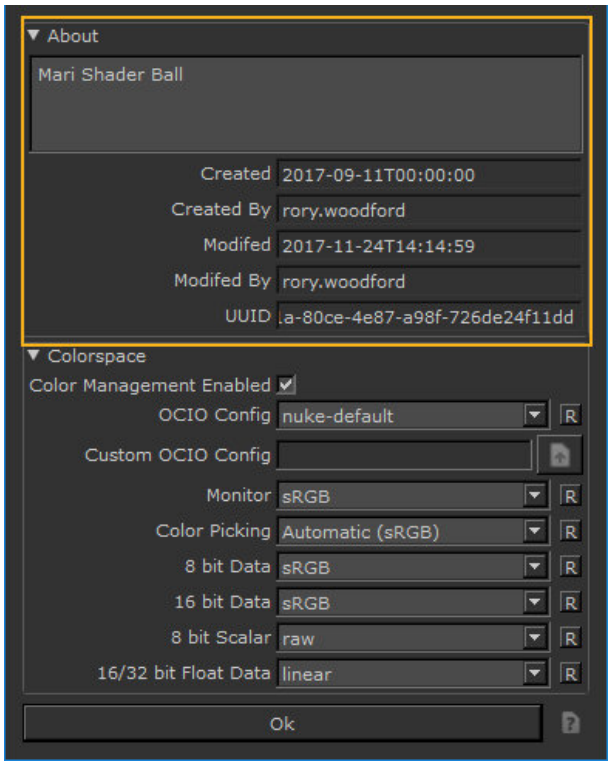
In Mari 3.3, the **New Project** dialog contains a lot of fields and information in a single tab only.

In Mari 4.0, the **New Project** dialog now contains four tabs to properly lay out the information. It also includes a new **Lighting** tab for you to set up the initial project lighting. When using channel presets, Mari now automatically builds a shader and connects the created channels to the corresponding shader inputs.

Project Settings Dialog



In Mari 3.3, the **Project Settings** dialog contains only Color Management information.



In Mari 4.0, the **Project Settings** dialog now contains details about your project and a description text box.

See the Project Settings Dialog.

Controlling the Project

Video: Watch Mari 4.0 | Controlling The Project.

Mari 3.3 contains the **Lighting**, **Selection**, **Paint Buffer Symmetry**, and the **Default Shaders** toolbars.

In Mari 4.0, the **Lighting**, **Selection**, **Paint Buffer Symmetry**, and the **Default Shaders** toolbars have been added to the **Project Controls** toolbar.

To change the current shader in the canvas to a system shader, click and hold the second button in the new **Project Controls** toolbar. Mari 4.0 has also introduced the concept of a current User Shader. Selecting this option in the dropdown menu uses the last selected User Shader as the current shader in the canvas.

Additionally, Mari now saves the lighting mode used alongside certain shaders. User Shader and Current Paint Target have separate saved lighting modes, while the **Current Channel**, **Current Layer** and **Current Layer & Below** system shaders share a saved lighting mode. This means you have the option to always view your channel components with flat lighting, while having full lighting with shadows automatically changed when switching to a User Shader.

See the Toolbars.

Using Tools

Tools Toolbar

In Mari 3.3, the **Tools** toolbar is spread out and depending on your screen vertical height resolution, you need to click the double arrow icon to display all the toolbar buttons.

In Mari 4.0, apart from the eye dropper and color controls, the tools have been grouped by category to make room for other toolbars. You can now find the following tool categories:

- Objects tools
- Painting tools
- Image Painting tools
- Paint Transformation tools
- Paint Buffer tools

The icon changes to the selected tool. You can also use keyboard shortcuts to switch to any tool. Click and hold any tool group button to access a menu of tools in the group. **Ctrl**+click a tool group button to cycle through the tools in the group.

For more information on the **Tools** toolbar, See the Toolbars.

Tool Properties Toolbar

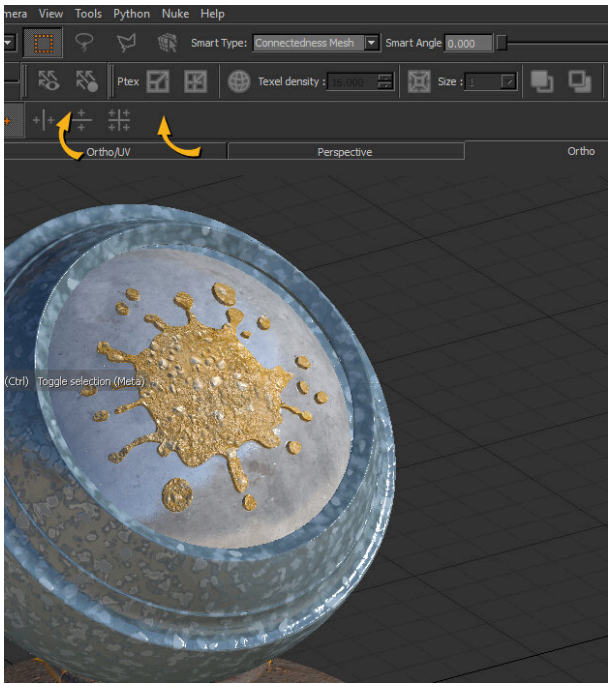
Video: Watch Mari 4.0 | Tool Properties Toolbar.

In Mari 3.3, the tool's **Tool Properties** toolbars contain too many text labels making them too long to display on one line.

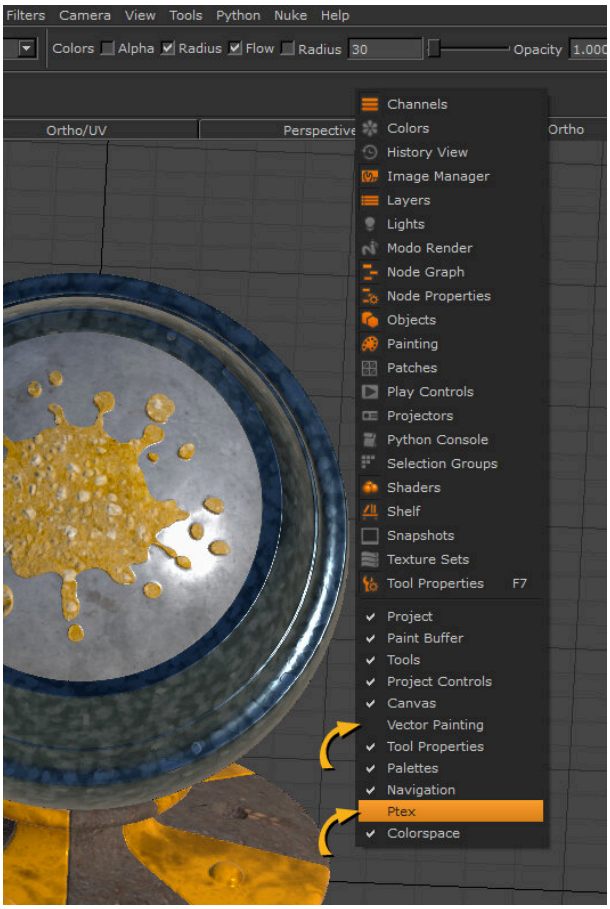
In Mari 4.0, the tool's **Tool Properties** toolbars have been revisited. Icons have replaced text labels, dropdowns have been replaced with checkboxes, a brush splat is used for all painting tools, and so on.

For more information on the **Tool Properties** toolbar, See the Toolbars.

Ptex and Vector Painting Toolbars



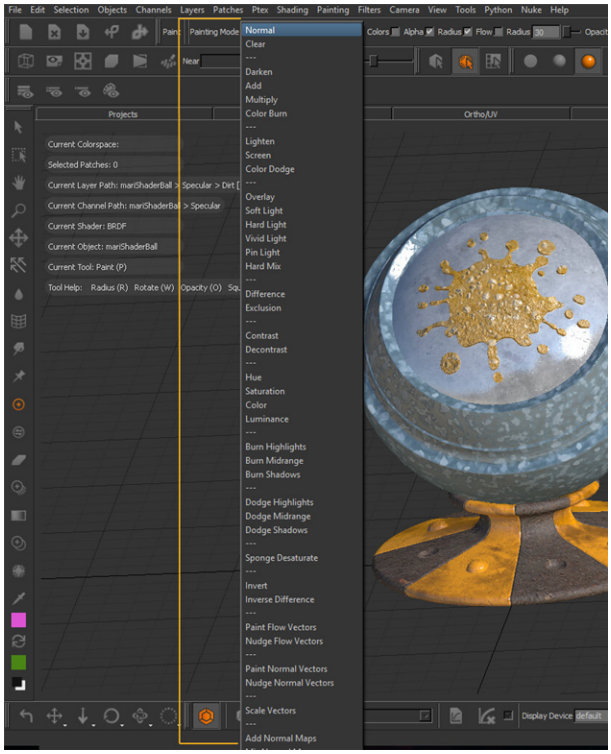
In Mari 3.0, both **Ptex** and **Vector Painting** toolbars are displayed in Mari default layout.



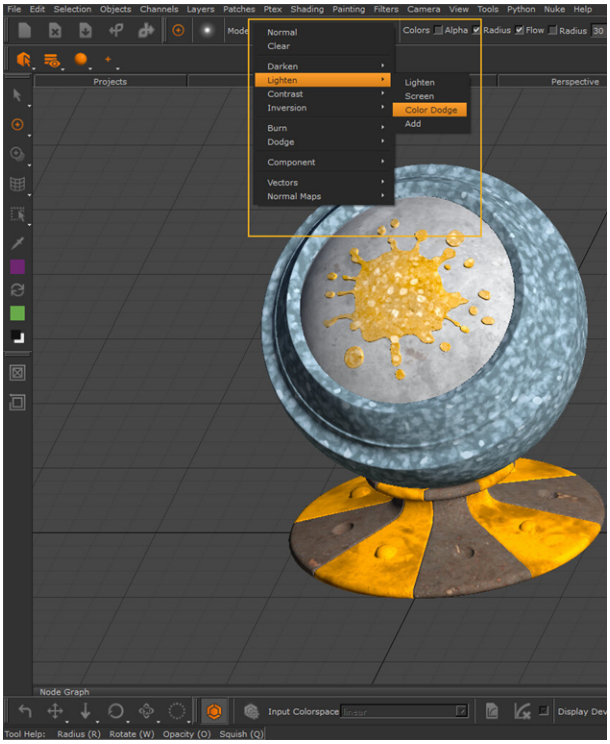
In Mari 4.0, both **Ptex** and **Vector Painting** toolbars are hidden by default. You can access them through the toolbar's right-click menu.

See the Toolbars.

Painting in Mari



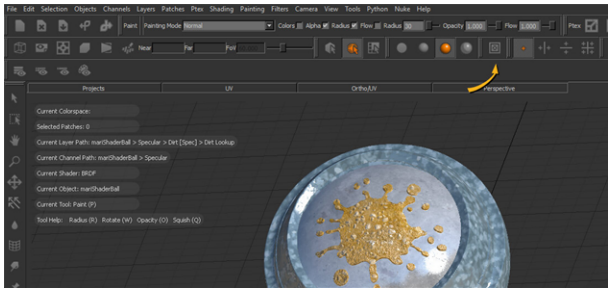
In Mari 3.3, the Blending modes display in a long dropdown menu taking a lot of space and making it hard to see what Blending mode to use.



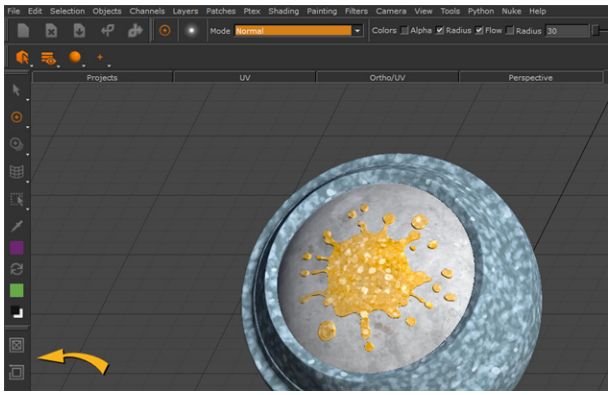
In Mari 4.0, Blending modes have been categorized making them easier to navigate to in the dropdown menu.

See Paint Blending Modes.

Using the Paint Buffer



In Mari 3.3, the **Painting** toolbar only



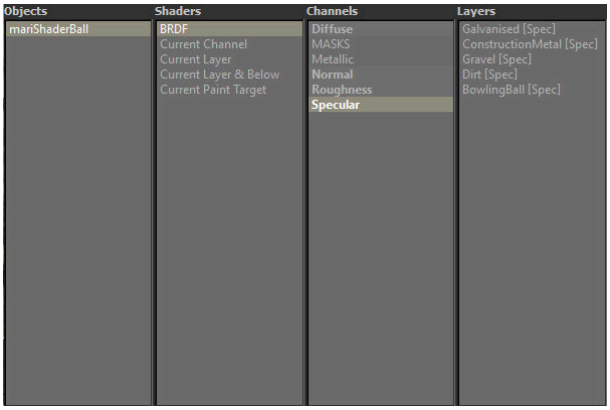
In Mari 4.0, the **Painting** toolbar has been renamed to

contains one tool that clears unbaked paint.

Paint Buffer toolbar and contains another tool that resets the transformation of the paint buffer.

For more information on the **Paint Buffer** toolbar, See the Toolbars.

Choosing a Paint Target



In Mari 3.3, the **Paint Target** quick palette (hold the **I** key) allows you to select your objects, shaders, channels, and layers.

In Mari 4.0, the **Paint Target** quick palette (hold the **I** key) allows you to select your objects, shaders, channels, layers, and paint targets (paintable layers or masks).

Accessing Palettes

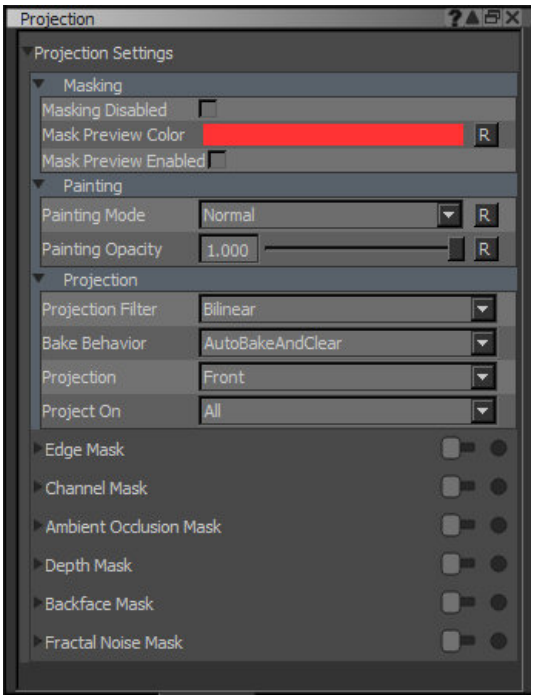
Video: Watch Mari 4.0 | Accessing Palettes.

In Mari 3.3, you can access palettes by navigating to **View** > **Palettes** and through the toolbar's right-click menu.

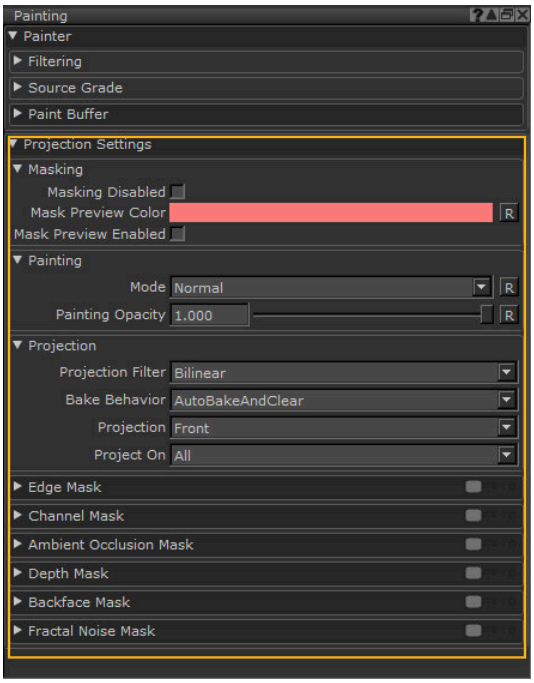
In Mari 4.0, in addition to accessing the palettes using the main menu and the toolbar's right-click menu, there is now a new **Palettes** toolbar.

For more information, see Configuring Your Workspace.

Projection Settings



In Mari 3.3, the projection settings are located in the **Projection** palette.



In Mari 4.0, the projection settings are now in the **Painting** palette under the Projection Settings group.

See the Painting Palette.

Group Layer Workflows

Video: Watch Mari 4.0 | Group Layer Workflows.

In Mari 3.3, grouping, ungrouping, and adding layers is not very intuitive.

In Mari 4.0, grouping, ungrouping, and adding layers can now be achieved quickly and efficiently. Creating new Group layers, with layers selected, now groups those layers inside the new Group Layer. Creating new layers when a Group Layer is selected, adds those new layers to the selected Group Layer.

For more information, see Groups, Pass-Through, and Mask Stacks and the Layer Functions.

Editing Brushes

Video: Watch Mari 4.0 | Editing Brushes.

In Mari 3.3, you can edit your brushes using the **Brush Editor** and the **Tool Properties** palettes. In the **Brush Editor**, the **Shelves**, and **Presets** tab, in which your brushes are stored, can also be found in the **Shelf** palette.

In Mari 4.0, to make it simpler the **Brush Editor** palette has been removed and you can find the brush properties in the **Tool Properties** palette. To save your custom brushes, you can drag the brush splat icon from the **Tools Properties** toolbar to the **Shelf** palette.

See the Tool Properties Palette and Configuring Brushes for more details.

Working with Colors

Video: Watch Mari 4.0 | Working with Colours.

In Mari 3.3, the **Colors** palette is not resizable and contains the foreground and background swatches, which are duplicates of the ones in the **Tools** toolbar. You can pick colors using the **Colors** palette, the **Colors** quick palette, and the **Select Color** dialog. The **Pixel Analyzer** is a separate palette.

You can find the color management settings in the **Color Manager (Viewer Transform)** palette as well as in the **Project Settings** dialog.

In Mari 4.0, the **Colors** palette has been entirely redesigned. You can now resize the palette and the **Pixel Analyzer** palette has been merged with the **Colors** palette. The gray and blend controls have been improved. The **Colors** palette, the **Colors** quick palette, and the **Select Color** dialog that are used to pick colors now share a similar design.

The **Color Manager (Viewer Transform)** palette has been removed (although can be displayed using an environment variable), and the color management settings can still be found in the **Project Settings** dialog.

For more information, see the Colors Palette and the Project Settings Dialog.

Filling Selected Objects, Faces, and Patches with Colors

Video: Watch Mari 4.0 | Fill Mechanisms.

In Mari 3.3, you fill the selected patches or faces using the **Patches** > **Fill** options.

In Mari 4.0, you fill the selected objects, patches, or faces using the **Selection** > **Fill** options and by dragging-and-dropping a color swatch to the selection.

See the Selection Functions.

Curve Editor

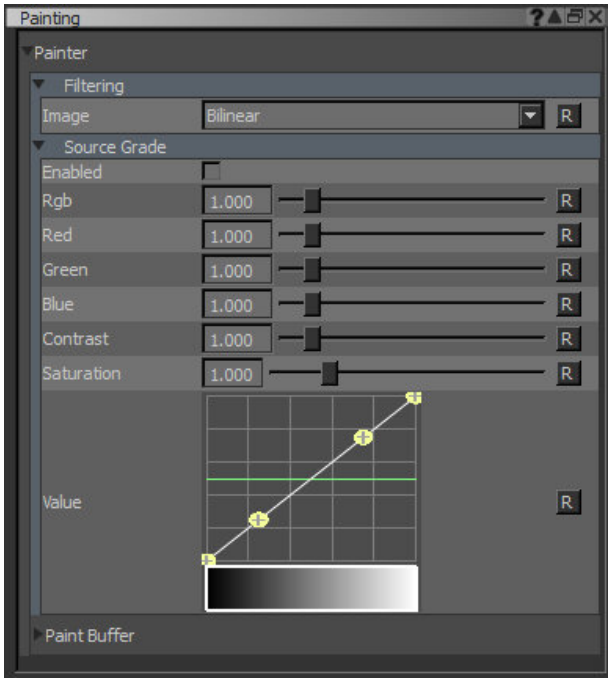
Video: Watch Mari 4.0 | Curve Editor.

In Mari 3.3, the Curve Editor is very basic.

In Mari 4.0, the Curve Editor has been entirely redesigned. A curve attribute now displays as a grayscale gradient in Mari's properties panels and a separate, scalable curve editor window allows for precise editing.

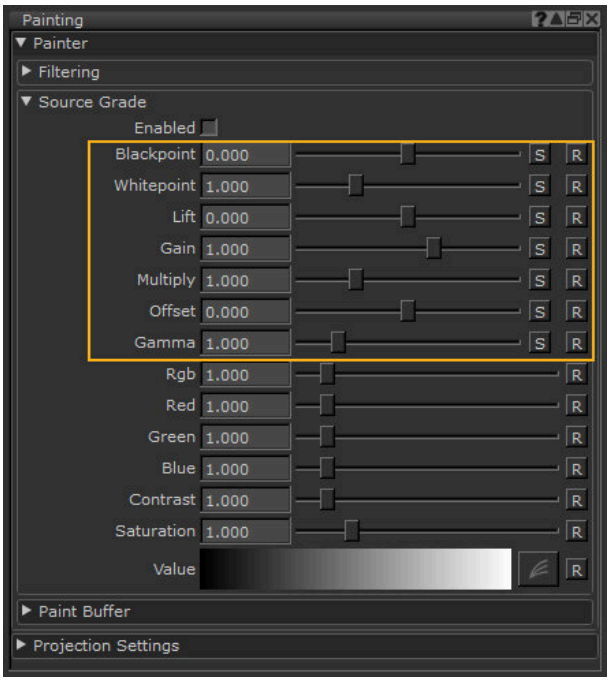
See the Types of Controls.

Controlling Source Grading



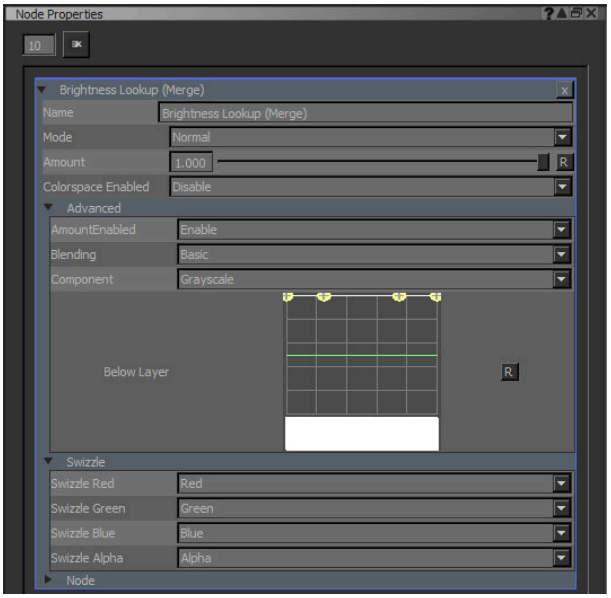
In Mari 3.3, in the **Painting** palette, the **Source Grade** section contains a limited amount of controls.

See the Painting Palette.



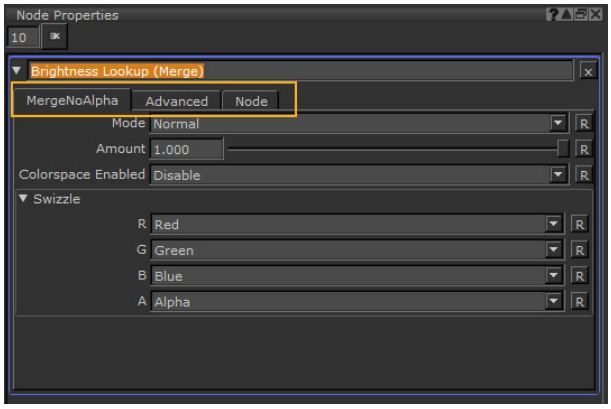
In Mari 4.0, in the **Painting** palette, the **Source Grade** section now contains the same controls as the **Grade Adjustment** layer.

Using the Node Graph



In Mari 3.3, the Node Graph contains two modes: basic and advanced.

The **Node Properties** palette stacks a node's property groups in a single pane.



In Mari 4.0, the Node Graph's basic mode has been removed and the Node Graph is now accessible in the advanced mode only.

The **Node Properties** palette displays a node's property groups in separate tabs to make it more readable.

For more information, see Node Graph.

Subdividing Objects

Video: Watch Mari 4.0 | Subdividing Objects.

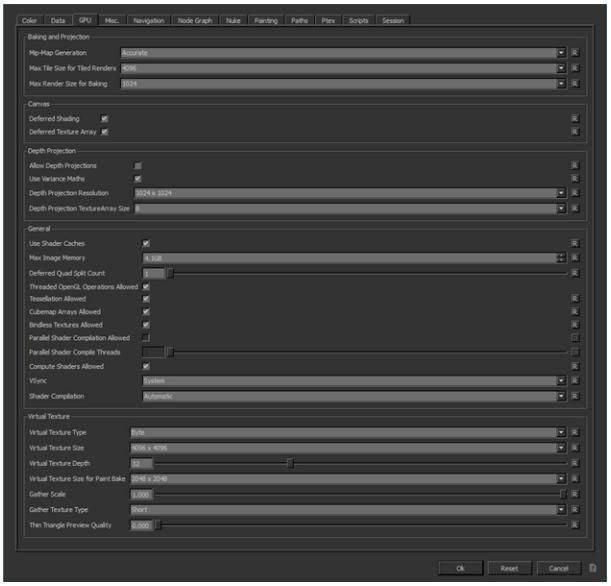
In Mari 3.3, the subdivision options are very limited.

Mari 4.0 now supports OpenSubdiv 3.1, providing you with more options to subdivide your objects.

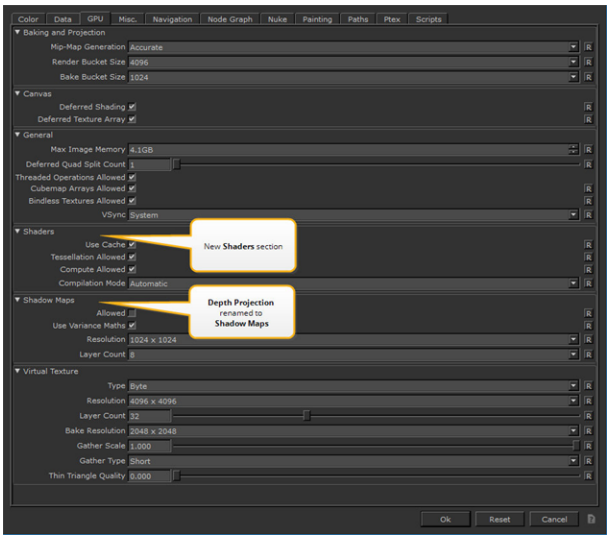
See Subdividing Objects and the Subdivide Dialog.

Interface

Project Preferences



In Mari 3.3, in the **Mari Preferences** dialog, some preferences in the **GPU** tab have confusing labels.

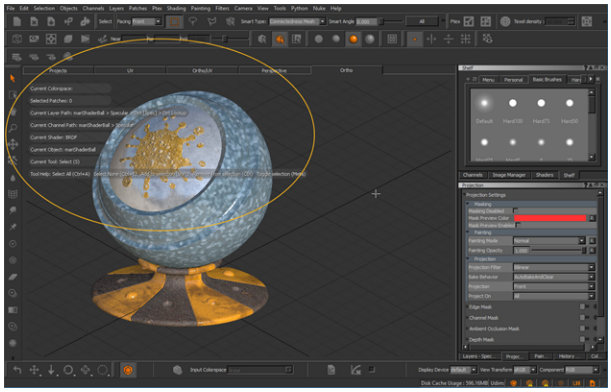


In Mari 4.0, the **GPU** tab has been reorganized and some preferences have been renamed. The **Depth Projection** section has been renamed to **Shadow Maps** and a **Shaders** section has been added to better categorize the preferences.

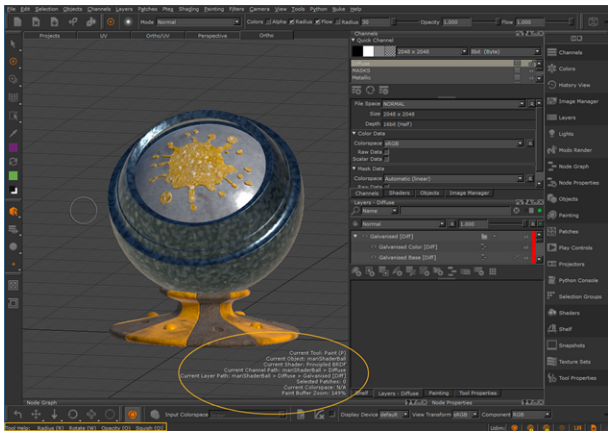
In the **Misc** tab, the **Slider Precision** preference has been moved to the top, in the **Appearance** section. The **Session** tab has been removed.

See the Mari Preferences Dialog.

Head-Up Display



In Mari 3.3, the Head-Up Display (HUD) and the tool help are located in the top-left of the canvas, overlapping with your asset while painting.



In Mari 4.0, the HUD is located in the bottom-right of the canvas and the tool help in the status bar giving you more space for your painting experience.

Exporting Channels

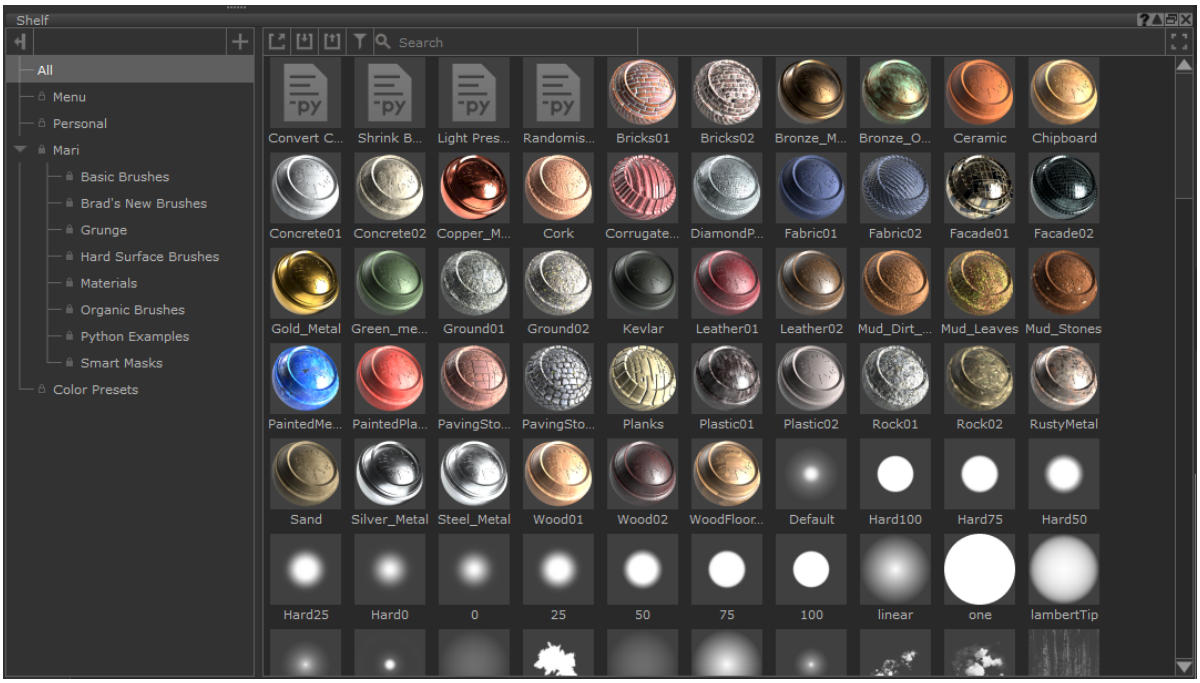
Video: Watch Mari 4.0 | Exporting Channels.

In Mari 3.3, you can only export channels one dialog at a time by navigating to **Channels** > **Export** or **Export Flattened**.

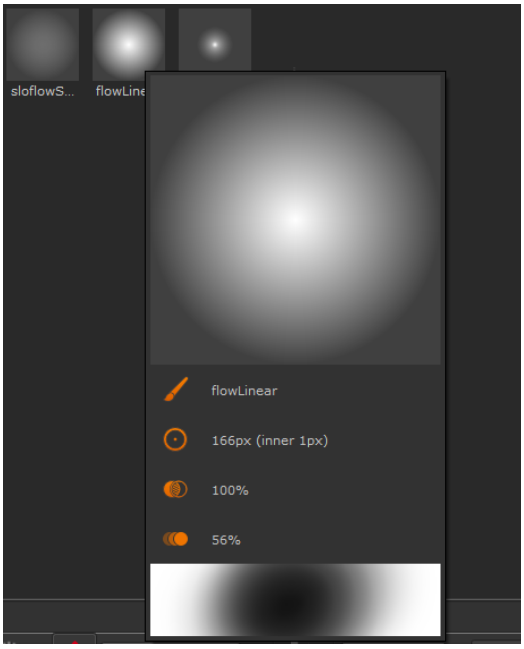
In Mari 4.0, you can export multiple channels at once using the **Export Manager**. For more information, see the Export Manager Dialog.

Ready-to-use Shelf Content

The Shelf Palette contains a range of brushes, materials, and smart masks for you to use.



Note: You can find further information about each item by hovering your mouse over the item in the Shelf. This will display a pane with key details.



Tip: Select any brush from the shelf to start painting with it.

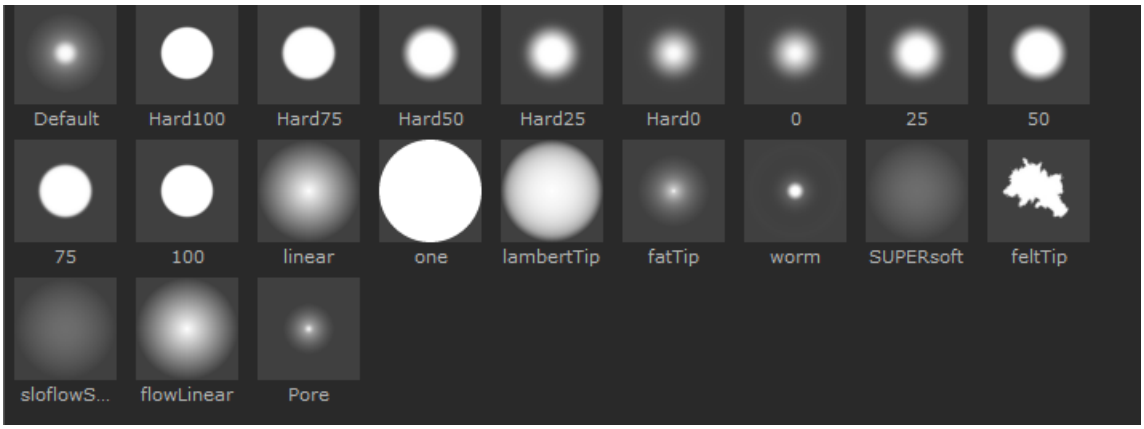
Tip: Drag materials and smart masks directly from the shelf into your node graph to start using them.

Select each heading to see what each shelf section includes:

Brushes

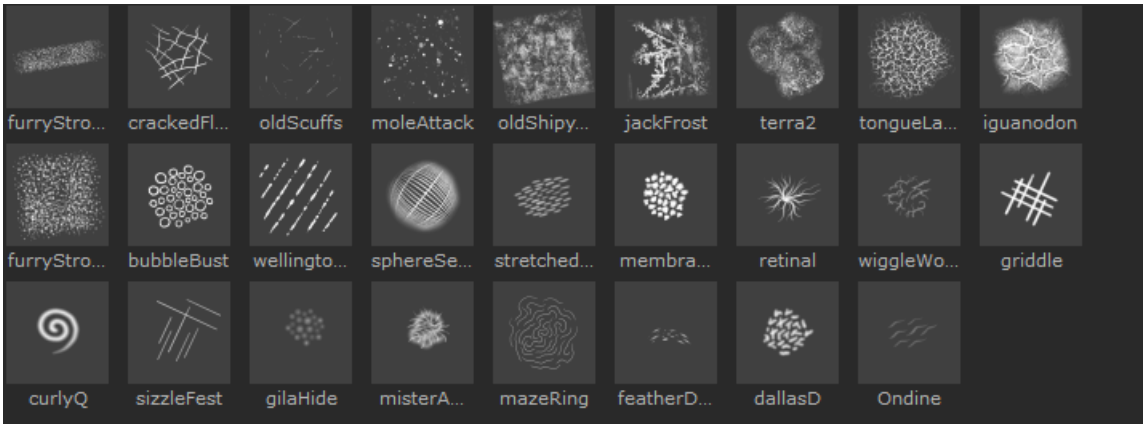
Basic Brushes

A set of predefined basic brushes that are versatile in a variety of use cases, and a good place to get started with painting.



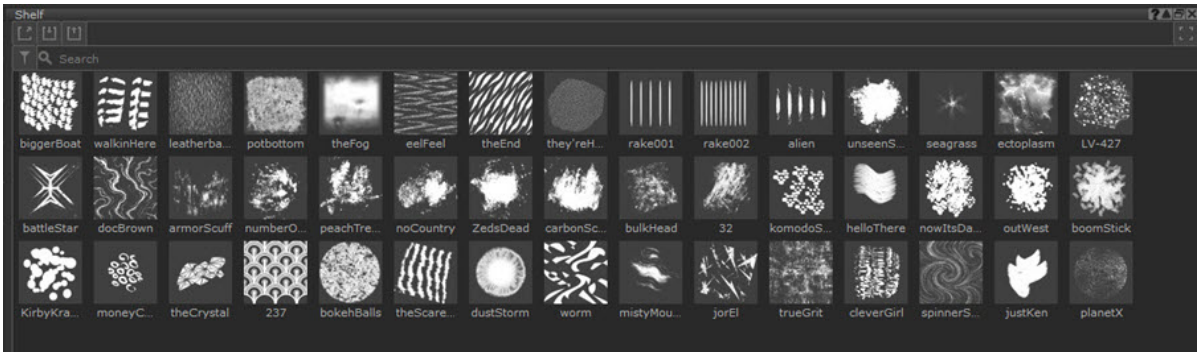
Brad's New Brushes

A set of predefined brushes created by Bradford DeCaussin, which can be used to add small detail to your textures. For example, you can use the **moleAttack** brush to add spots and blemishes to skin, and use **jackFrost** to add texture to objects in cold environments.



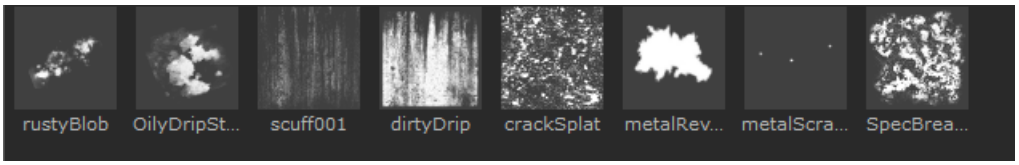
Brad's Movie Brushes

Another set of predefined brushes created by Bradford DeCaussin, which can be used to add small detail to your textures. For example, experiment with **cleverGirl** for a reptilian creature's skin.



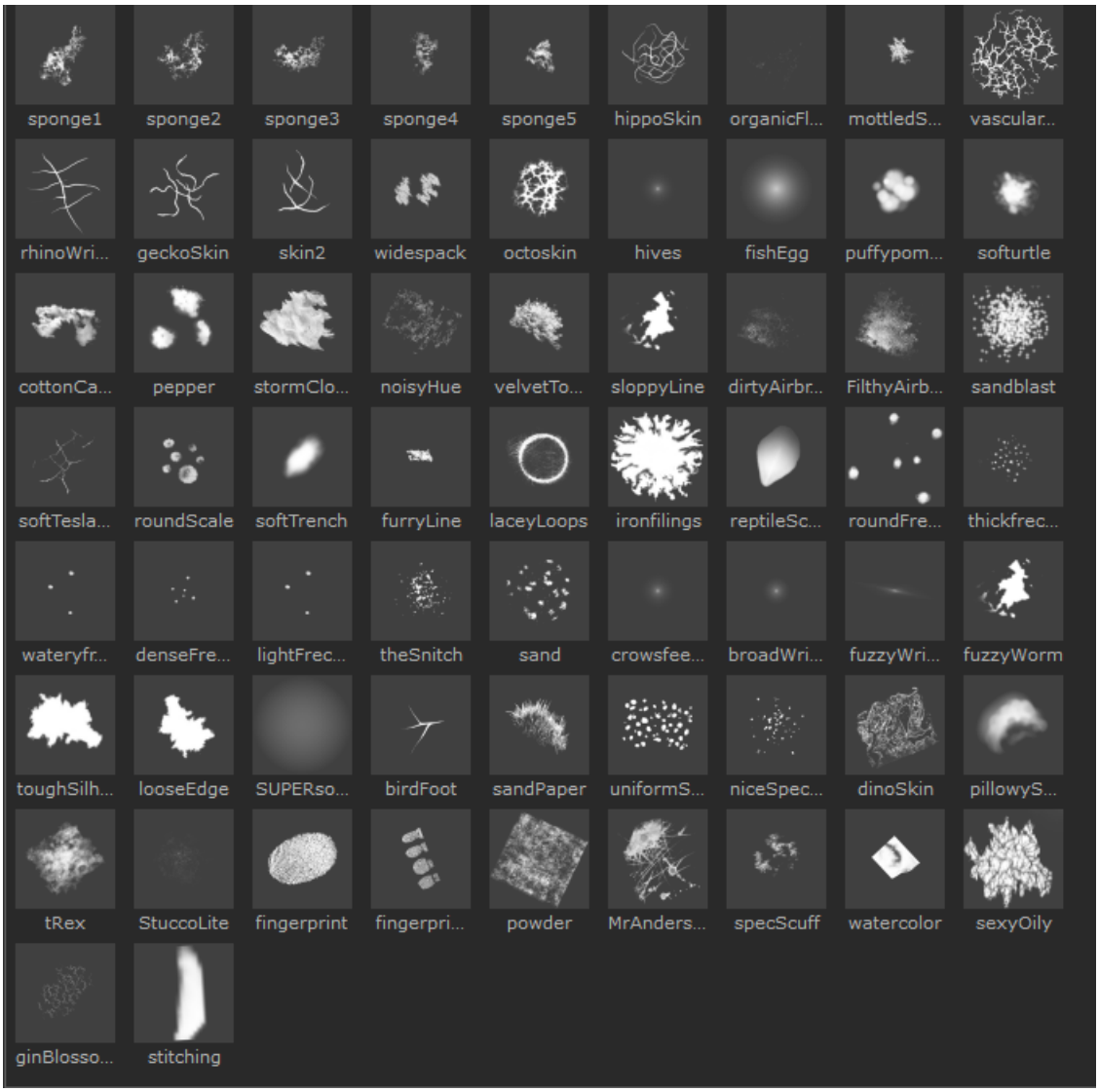
Hard Surface Brushes

A set of predefined hard surface brushes. These are particularly useful for adding detail to metal surfaces, such as scrapes and scuffs.



Organic Brushes

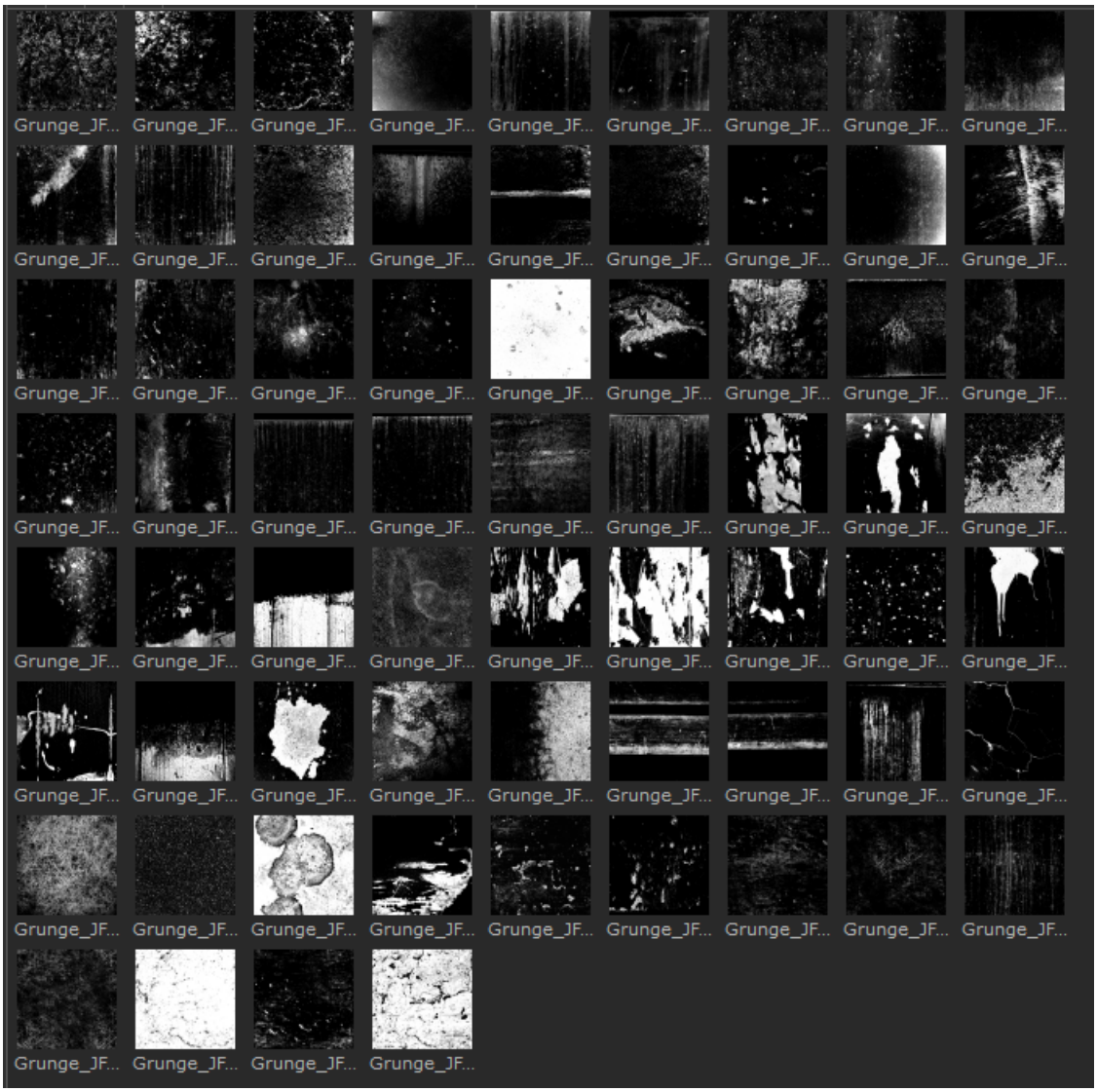
A set of predefined organic brushes, useful for creating plant and animal textures.



Grunges

Grunge

A shelf which comes loaded with high-quality grunge textures created by industry professionals. This includes all of the textures used in the new Smart Masks collection, by Stuart Ansley, plus the Grunge Textures for Mari pack released alongside Mari 7.0, by Johnny Fehr.



Materials

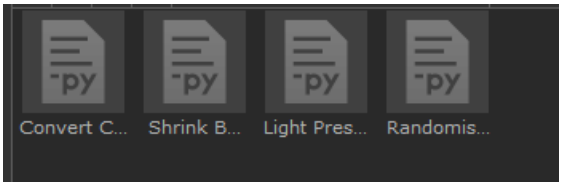
Materials

A set of predefined materials based on the Principled BRDF shader model. There are a range of manmade and organic materials for a range of use cases, from adding fabric to clothing, to creating shiny gold coins. See Managing Materials in Mari for more information.

Scripts

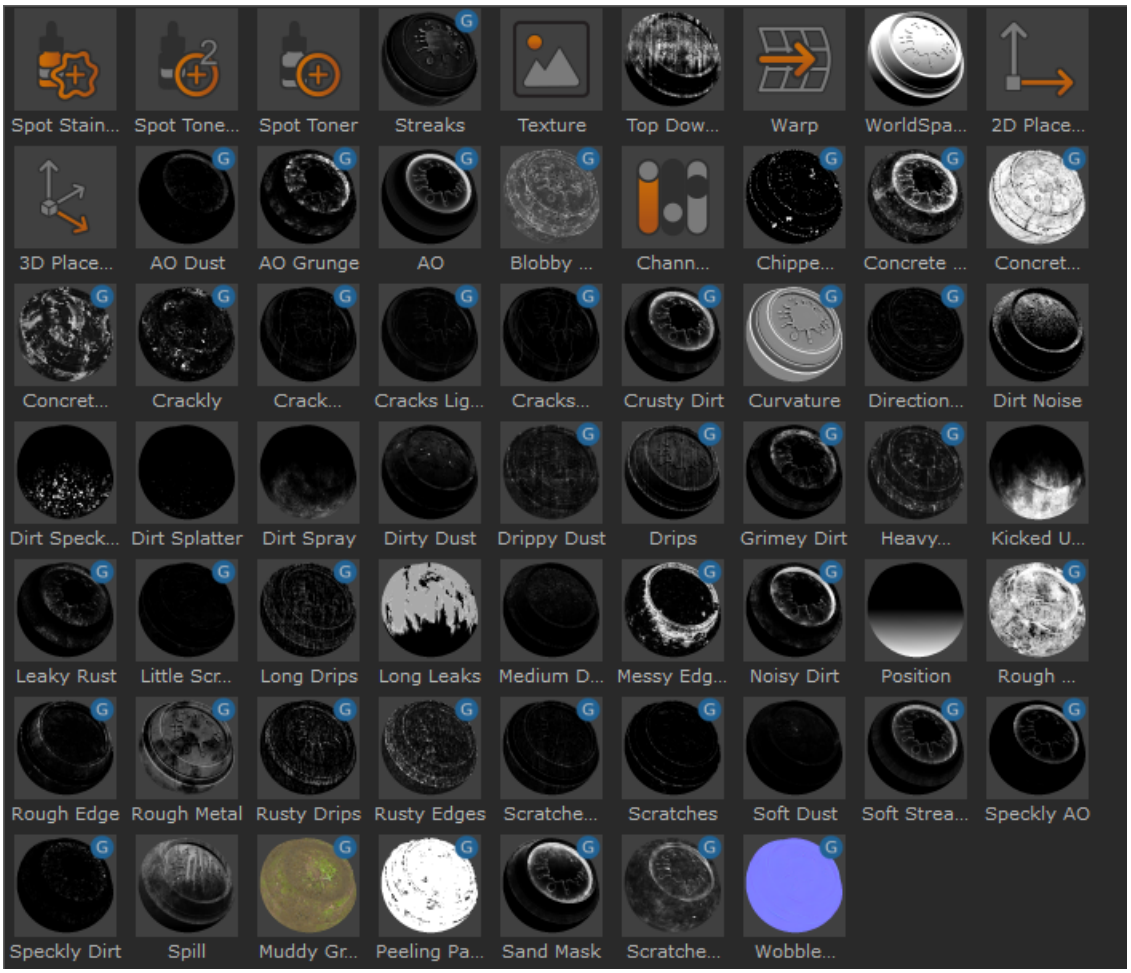
Python Examples

A set of predefined python scripts to perform certain useful functions. For example, **Shrink Backdrop** can help tidy up your node graph by shrinking a backdrop node to fit its contents.

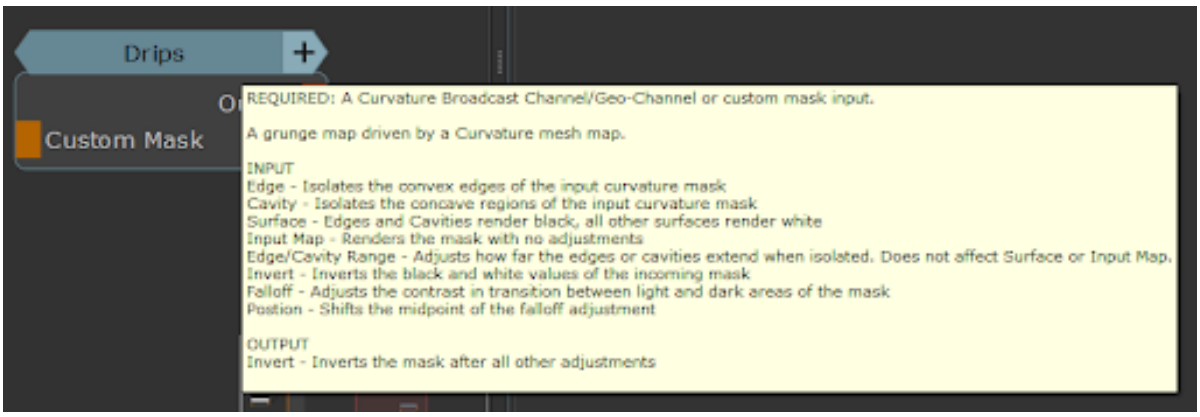


Smart Masks

Designed by industry professional, Stuart Ansley, this shelf contains over 50 **Smart Masks** and **Utilities**. These will help you in creating versatile textures with breakups and details, allowing for more realistic and interesting designs. Using the Utilities, you can now easily modify the position and placements of their textures, giving you greater control over the final look.



Tip: Once you've added them to your node graph, the Smart Masks and Utilities nodes also have more detailed tooltips which give further information about their usage and controls.



Note: The shelf includes **Basic Smart Masks**, **Grunge Maps** and **Utilities**.

Basic Smart Masks

Basic Smart Masks are controllers that hold mesh map information. For example:

- **(AO) Ambient Occlusion**: A basic controller for an Ambient Occlusion mesh map. Requires an "AO" Broadcast Channel/Geo-Channel or custom mask input.

See controls

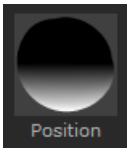
Input | **Invert**	Inverts the black and white values of the incoming mask
Input | **Falloff**	Adjusts the contrast in transition between light and dark areas of the mask
Input | **Position**	Shifts the midpoint of the incoming mask, effectively adjusting the range of the mask. Will not affect the mask if Falloff is set to 1
Output | **Invert**	Inverts the mask after all other adjustments

- **Curvature:** A basic controller for a Curvature mesh map. Requires a "Curvature" Broadcast Channel/Geo-Channel or custom mask input.

See controls

Input | **Edge**	Isolates the convex edges of the input curvature mask
Input | **Cavity**	Isolates the concave regions of the input curvature mask

Input | **Surface**	Edges and Cavities render black, all other surfaces render white
Input | **Input Map**	Renders the mask with no adjustments
Input | **Edge/Cavity Range**	Adjusts how far the edges or cavities extend when isolated. Does not affect Surface or Input Map
Input | **Invert**	Inverts the black and white values of the incoming mask
Input | **Falloff**	Adjusts the contrast in transition between light and dark areas of the mask
Input | **Position**	Shifts the midpoint of the incoming mask, effectively adjusting the range of the mask. Will not affect the mask if Falloff is set to 1
Output | **Invert**	Inverts the mask after all other adjustments



• **Position:** A basic controller for a Position gradient for the x, y, and z axis. Note that the output is clamped.

See controls

Axis | **X**	The blend amount of the x (left-right) gradient contribution
Axis | **Y**	The blend amount of the y (up-down) gradient contribution
Axis | **Z**	The blend amount of the z (front-back) gradient contribution
Output | **Invert**	Inverts the mask after all other adjustments
Debug | **Debug Enabled**	Activates Debug Mode. The output will display Debug Colors if enabled and will ignore any Falloff, Falloff Position, or Invert attributes. This

	mode can be helpful to align the full range of the Position Gradient from white to black on the 3D mesh using the Floor Offset and Broad/Fine Scale Adjustments
Debug | **Debug Colors**	Activates Debug Colors to more easily visualize the opposite ends of the gradient (pure white and pure black). A value of 1 is displayed in magenta and a value of 0 is displayed in green
Debug | **Floor Offset**	Adjusts the starting point of the gradient. With the default Y Axis active and isolated, the starting point of the gradient will move up and down in space. Lower the slider number to raise the floor starting point
Debug | **Broad/Fine Scale Adjustment**	Increases or decreases the range of the gradient. Smaller assets typically only require the Fine Scale Adjustment slider but larger assets may benefit from a combination of the Broad and Fine Scale Adjustments
Input | **Falloff**	Adjusts the contrast in transition between light and dark areas of the mask
Input | **Position**	Shifts the midpoint of the incoming mask, effectively adjusting the range of the mask. Will not affect the mask if Falloff is set to 1
Output | **Invert**	Inverts the mask after all other adjustments

- **WorldSpaceNormal**: A basic controller for a World Space Normal gradient for the x, y, and z axis. Note that the output is clamped.

See controls

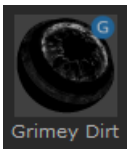
Axis | **X**	The blend amount of the x (left-right) gradient contribution

Axis | **Y**	The blend amount of the y (up-down) gradient contribution
Axis | **Z**	The blend amount of the z (front-back) gradient contribution
Input | **Invert**	Inverts the black and white values of the incoming mask
Input | **Falloff**	Adjusts the contrast in transition between light and dark areas of the mask
Input | **Position**	Shifts the midpoint of the incoming mask, effectively adjusting the range of the mask. Will not affect the mask if Falloff is set to 1
Output | **Invert**	Inverts the mask after all other adjustments

Grunge Maps

Each **Grunge Map** Smart Mask is a combination of a Basic Gradient Input and a Breakup in the form of a tileable texture map, a procedural, noise, or both. All Grunge Map Smart Masks have the same controls as those listed in the Basic Smart Masks, with a few additions.

The following is an example of a typical Grunge Map Smart Mask, driven by an AO mesh map and a tileable texture map with controls for 2D or 3D Placement. In some cases there will be additional similar controls for procedural noises.



- **Grimey Dirt**: A grunge map based off an Ambient Occlusion mesh map. Requires an “AO” Broadcast Channel/Geo-Channel or custom mask input.

See controls

Smart Mask Tab	
Input | **Invert**	Inverts the black and white values of the incoming mask

Input | **Falloff**	Adjusts the contrast in transition between light and dark areas of the mask
Input | **Position**	Shifts the midpoint of the incoming mask, effectively adjusting the range of the mask. Will not affect the mask if Falloff is set to 1
Output | **Contrast**	Adjusts the contrast of the output map
Output | **Contrast Pivot**	Adjusts the midpoint of the output contrast adjustment
Output | **Invert**	Inverts the mask after all other adjustments
Breakup Tab	
Texture |	Texture map used to provide breakup for the mesh map. Generally mixed using an Overlay, Multiply, or other blend mode with the input mesh map
Texture | **Rotation Angle (Degrees)**	Angle of the texture map
Texture | **UV Repeat**	Number of repeating tiles of the texture map
Texture | **U Repeat**	Number of horizontal texture repeats
Texture | **V Repeat**	Number of vertical texture repeats
Texture | **U Offset**	Left and right 2D texture placement offset
Texture | **U Offset**	Up and down 2D texture placement offset
Texture | **Mirror U**	Flips the texture horizontally
Texture | **Mirror V**	Flips the texture vertically
Texture | **Triplanar**	Disables 2D controls, activates Triplanar texture projection, and enables Triplanar projection controls
Texture | **Front Angle**	Adjusts the rotation of the front projection
Texture | **Top Angle**	Adjusts the rotation of the top projection

Texture | **Right Angle**	Adjusts the rotation of the side projection
Texture | **Uniform Scale**	Adjusts the world scale of the texture projection
Texture | **Scale X**	Adjusts the left-right scale of the projection
Texture | **Scale Y**	Adjusts the up-down scale of the projection
Texture | **Scale Z**	Adjusts the front-back scale of the projection
Texture | **Translate X**	Adjusts the left-right position of the projection
Texture | **Translate Y**	Adjusts the up-down position of the projection
Texture | **Translate Z**	Adjusts the front-back position of the projection

• Some nodes will have additional controls which may include the following:

Noise | **Size**	The blend amount of the x (left-right) gradient contribution
Noise | **Roughness**	The blend amount of the y (up-down) gradient contribution
Randomness | **Amount**	The blend amount of the z (front-back) gradient contribution
Randomness | **Size**	World scale of the the randomness noise
Randomness | **Roughness**	Rough breakup of the randomness noise
Leaks | **Length**	Length of the leaks based on the range of the input gradient

Note: Information on special attributes that are unique to a certain node can be found in the Node Description under the Node tab of the Smart Mask.

Utility Nodes

These tools stored in the shelf can help you create unique effects with Smart Masks.

• **Texture**

This node contains the controls found in a tiled texture node and triplanar texture node. This can help with scaling and stretching textures to get the desired effect.

• **Warp**

You can create a warp effect in both UVs and 3D, and switch to Directional mode to adjust the angle of the warp.

• **Spot Toner**

This helps add definition to your textures. When connecting a Texture as the Base and Paint node as the Over, you can then paint brightness and contrast onto your texture.

• **Spot Toner 2**

Similar to Spot Toner, but instead of working with a Multiply blend mode, it works with Gamma instead.

• **Spot Stainer**

Similar to Spot Toner 2 but a stronger effect. This utility is useful for blood or wet masks as you can change the color, depth, and falloff to make a realistic stain effect.

• **Channel mixer**

Add and mix colors to different channels, creating a variety of color effects.

• **2D Placement**

Control the rotation, position, and scale of textures in UV space.

• **3D Placement**

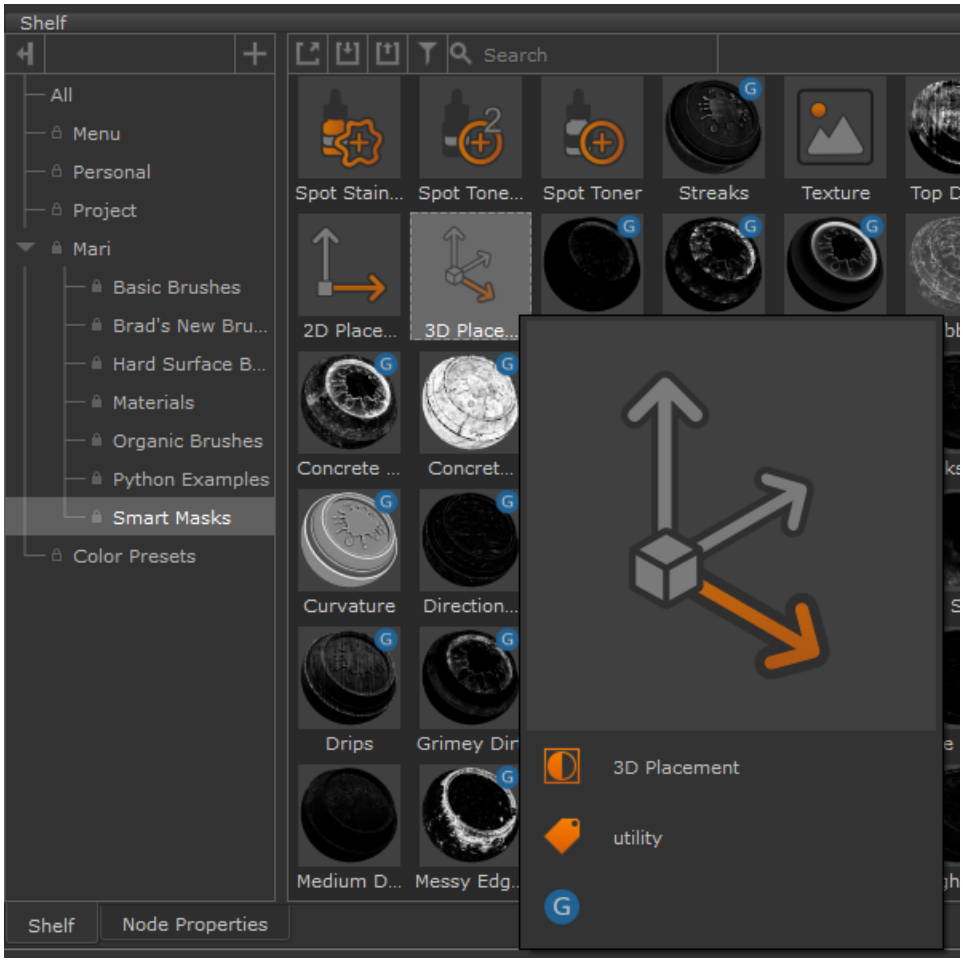
Control the rotation, position, and scale of textures in 3D space.

Note: To learn more about using Smart Masks, see Smart Mask Node.

Utilities Example: Using 3D Placement

This example shows how to use a utility node from the shelf to manipulate the placement of a procedural.

1. Add a **3D Placement** node to your node graph from the Smart Masks Shelf (you can drag it directly into the node graph).



2. Create a Cloud node (**Nodes** > **Procedural** > **Fractal** > **Cloud**) in your node graph.
3. Plug the **Output** of 3D placement in the **Position** input of the Cloud node.
4. You can now use the utility tools to manipulate the rotation/scale/position in the Node Properties of the Smart Mask.

Managing Projects

A Mari project stores your work on geometries, and any associated textures. Mari's **Project** tab holds all the projects you are working on.

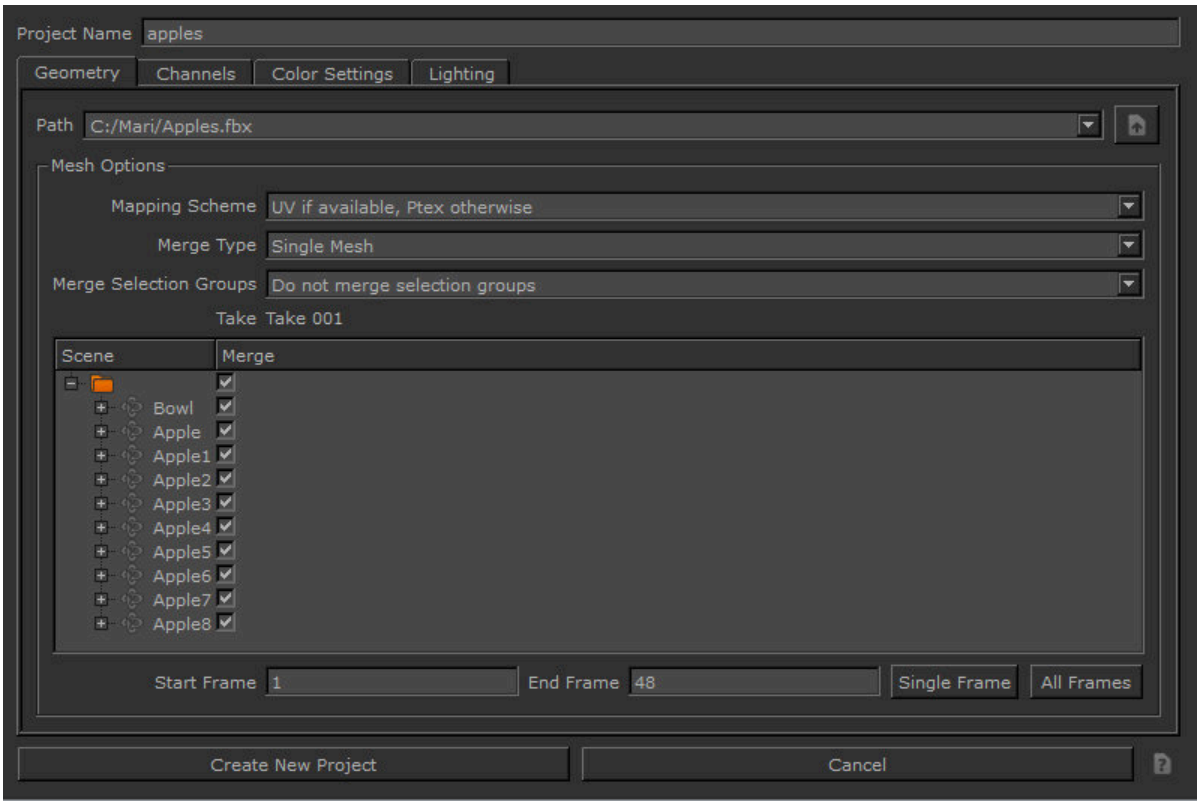
Note: For more detailed information about the **New Project** dialog, see the New Project Dialog section in the *Mari Reference Guide*.

Creating a New Project

1. Do one of the following:

Select from **menubar**	*or* click on **toolbar**	*or* click on the **Project tab**	*or* type **shortcut key**	*or* from the Project tab, right-click
File > **New**		New	**Ctrl/Cmd+N**	**New**

The **New Project** dialog displays.



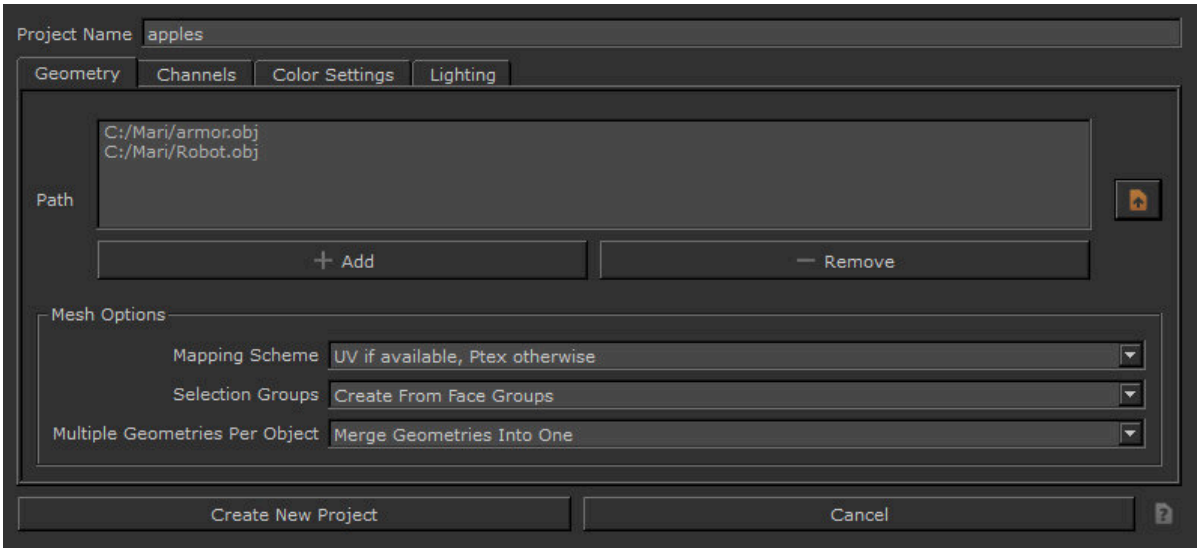
2. Set the **Project Name** (for you to identify it only - this is not a filename).
3. In the **Geometry** tab, follow these steps:
   - Select the geometry file for the project in the **Path** field. The dropdown lists the last 10 files selected, or you can click to browse to a file.

     Various mesh options display in the dialog directly under **Path**, depending on whether you open an **.obj**, **.ptx**, **.abc**, or **.fbx** file:

File formats	Mesh Options
.obj	You can select options for the **Mapping Scheme**, **Selection Groups**, and **Multiple Geometries Per Object**.
.ptx	You can select whether the mesh data in the file represents a single new object or, when there are multiple input files, if the mesh data in each file represents a separate geometry in a single new object by setting the **Merge Type** field. For more information on the **Mesh options**, see .
.abc and **.fbx**	You can select options for the **Mapping Scheme**, **Merge Type**, and **Merge Selection Groups**, as well as set the objects that are to be merged

File formats	Mesh Options
	in the object hierarchy.
.fbx	You can set the animation **Take** on project creation/object load. This binds the animation take to the object and this cannot be changed again once the animation take has been selected and loaded. For more information on the **Mesh options**, see .

Note: If you are loading an animated sequence with **.obj**, **.abc**, or **.fbx** file formats, **Mesh options** displays with the additional options **Start Frame**, **End Frame**, **Single Frame**, and **All Frames**. These provide the frame range to view either user-defined start and end frames, a single frame, or every frame. A **Frame Offset** also allows you to specify how the Alembic file is loaded into Mari. This is available for **.abc** files only.



- To create a project from multiple files, browse to a folder and select multiple files from the directory, before clicking **Open**.

  The selected files are all shown in the **Path** field. These files are stored in the history as a single project under the assigned name.

Note: It's only possible to create a project from multiple files with **.obj** and **.ptx** files. You can only load one **.abc** or **.fbx** file when creating a new project.

4. In the **Channels** tab, set the following channel options:

Channel option	Description
Root Path	Select the root path for the project. This is where Mari looks for any existing textures to import as it creates the channels.
Scan	Click **Scan** to make Mari scan the project's root path for any existing textures. If any textures exist for your channels, Mari shows a green dot next to the channel. You can select to import these textures.
Category	Selects a Mari channel preset template from the list of available shaders. The selected channel preset updates with channels.
Template	Select a channel template. Channels templates control the format of the filenames when Mari looks for existing textures on disk.
Prefix	Set a filename prefix for existing texture files, if applicable.
Create	Check the box to create a shader and connect channels to the shader on project creation. You can right-click to get a dropdown menu allowing you to easily create or import all the channels.
Import	Check the box to import existing textures into the channel after creating it. You can right-click to get a dropdown menu allowing you to easily import all the channels.
Name	This is the name of the channel to create. Mari sets a default value. As with the project name, the channel **Name** is for you to identify it only - when you export you can select a different name.
Size	Set a patch size for the new channel.
Colorspace	Set the colorspace in which the colors within your project are interpreted. Either select from the shipped config files or allow Mari to set an **Automatic** colorspace.
File Space	Set the file space used when generating vectors.
Fill	Set the **Fill** option to fill a channel with a specified color, by default.

Depth	Set the **Depth**. If you select a different **Depth** or resolution, file sizes can vary dramatically (for example, a 2k RGB "Byte" texture file is 2MB - whereas a 4k "Float" is 172MB).
Files	Specify the filename template used to pick existing textures to import into the channel. Mari sets a default value, but you can change this, or you can use the button to open a dialog allowing you to manually pick the files.

5. In the **Color Settings** tab, set your colorspace options for the project in the following fields:

> **Tip:** If you are just starting out in Mari and aren't familiar with colorspaces, you may want to leave the colorspace fields at their default.

Colorspace Option	Description
Color Management Enabled	Enable or disable the OpenColorIO color management.
OCIO Config	Select a user-defined config file or shipped OpenColorIO config file that handles colorspace.
Custom OCIO Config	Specify the filepath for the custom (non-shipped) config file.
Monitor	Set the default colorspace, which applies to thumbnails and UI elements. Typically, this should be the same colorspace as you intend to use for the canvas.
Color Picking	Set the colorspace for all color pickers, swatches, and image viewers.
8 bit Data	Set the default colorspace for 8-bit channels, and reading and writing image files with 8-bit data.
16 bit Data	Set the default colorspace for 16-bit channels, and reading and writing image files with 16-bit data.
8 bit Scalar	Set the default colorspace for masks, heights, normals, depths and, generally, any non-color image with 8-bit data.

Colorspace Option	Description
16/32 bit Float Data	Set the default colorspace for 16- or 32-bit (float) channels, and reading and writing image files with 16- or 32-bit, floating point data.

6. In the **Lighting** tab, set the following options:

Lighting Option	Description
Shader	Select a shading model.
Lighting	Select the lighting on the main mesh.
Active Light Count	Set the number of active directional lights.
Environment Map	Select an environment light.

7. Click the **Create New Project** button.
   Mari renders your selection and displays it on the canvas.

Tip: Depending on the size and complexity of textures, it can take several minutes the first time you load them into a Mari project. Once saved, however, the project should open quickly in future.

Note: Mari checks the model for any errors that might prevent it from processing; for example, touch border edges. A Mesh Sanity Check dialog displays any warnings or errors. (**Errors** cancel opening the model, but you can continue with **warnings**.)

Note: For information on model requirements in Mari, refer to Modeling Requirements.

Opening Existing Projects

1. Click on the **Projects** tab.
   This shows all the projects you have on your computer.

> **Note:** You can see your **Disk Usage** at the bottom of the Projects tab. This is a percentage indicating how much disk space remains on the drive where you installed the Mari cache.

2. Double-click on the project to open.

> **Note:** On project load, Mari looks for metadata that is over 250MB and discards anything that is over this size. This is intended to strip corrupt and problematic data, and also affects metadata added through the Python API.

Using the Command Line

You can also open Mari and a specific, existing project, as well as archives using the command line. From the **Run** dialog on your computer, use a command in the format of:

./mari /tmp/testproject.mra

By specifying the project name, UUID, folder path, or project.mri path, you open Mari and the designated project.

The command line can also be used to run scripts that contain a Python call to open a project. From the **Run** dialog on your computer, use a command in the format of:

./mari example_script.py

The **example_script** should contain the Python call to open a project.

Saving Your Project

Do one of the following:

Select from **menubar**	*or* click on **toolbar**	*or* type **shortcut key**
File > **Save**		**Ctrl**/**Cmd**+**S**

Tip: Save your project often.
-Note that while your textures remain cached locally, saving does not export them, so export often too!
-If you don't need to save (no changes since the last save), the toolbar icon is gray.

Article: Take a look at: Q100188: Improving performance in Mari and in specific Projects for more information on how to optimize your projects.

Using Automatic Project Backup & Restore

Mari has a highly optimized project backup system, which makes it easy to maintain your work and restore archives to a previous state in time.

Note: For more information see Project Settings: Settings

Note: Saves in Mari are incremental and store only the difference since the previous save, which means the backup files are often smaller than a full Mari archive.

Setting Up Your Backup System

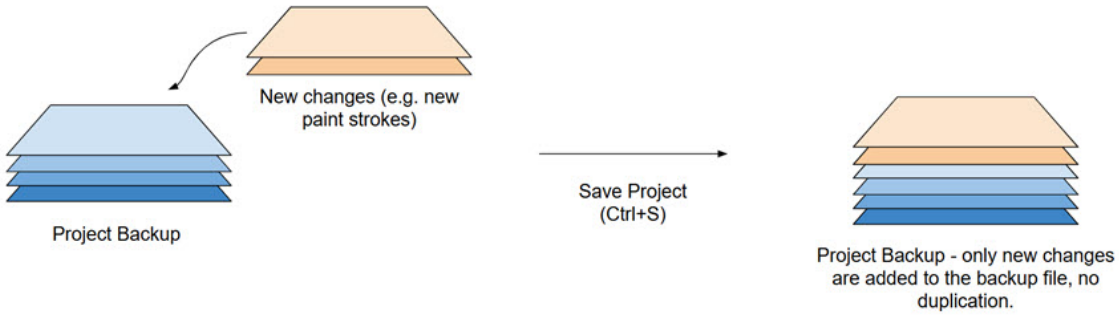
To set up the project backup system for your project:

1. Within the project, go to **File** > **Settings** > **Backup Path**.
2. Type in the backup path in the text input, or use the Add Directory button ( ) to select a backup destination on your machine/ network.
3. Each time you now save your project, Mari records the changes since your last save in this location.

Note: We recommend that your Backup Path is a separate location to the existing Project Location.

Using the Automatic Backup System

Each time you save your project, Mari records the changes since your last save. This snapshot is saved in your backup location as defined in **File > Settings > Backup Path**. This means Mari's backup system is incredibly lightweight.



The automatic backup system works asynchronously. This means that when you import, duplicate or restore a project, Mari loads the most crucial information you need first, and opens up the project with those files. You can then continue painting while your project continues to load all of its information.

> **Note:** By default, Mari will trigger the backup system when the current project is saved. To disable this behavior (e.g. if you want to manually trigger the backup system using a script), set a MARI__DONT_AUTO_TRIGGER_PROJECT_BACKUP=1 environment variable. See more at Studio Backup Integration.

Project Tagging

You can tag project backups by right-clicking the project and selecting **Tag Backup**. This will open up a text input for you to label your backup. This way you can easily identify each backup and group versions by using a common tag. You can also add tags to automatic backups in order to search for and identify them.

When you have a project open, and it has a backup location configured, you can run **File > Tag Version** (or use the **Ctrl+Shift+S** shortcut) to save the project and add a tag at the same time.

There is an option to tag your most recent backup by right-clicking the project and selecting **Tag Backup > Latest**.

Tip: Input multiple tags in the text input by separating them with a comma.

You can also add project tags if your project has a configured backup location, and this tag will be applied to the range of the project backups. This tag won't show up on the Restore options, but can be found when you hover over the Project thumbnail.

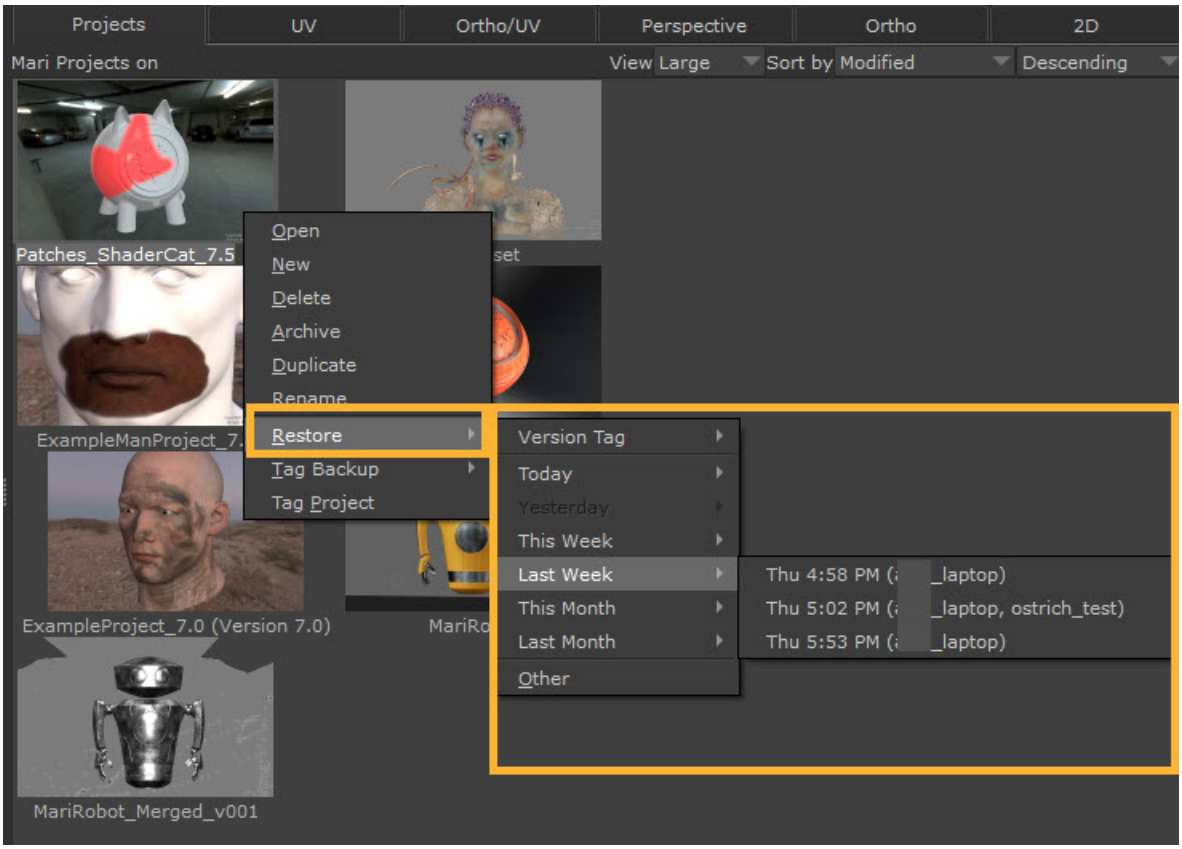
Opening Backups

The project Restore menu (Found at **Projects > Restore** or by right-clicking the project's thumbnail) is organized into date-time and tag-based sub-menus.

To see backups and restore the project to a previous version:

1. Open the **Projects** tab.
2. Right-click on a project thumbnail, and select **Restore** to see a list of backups.

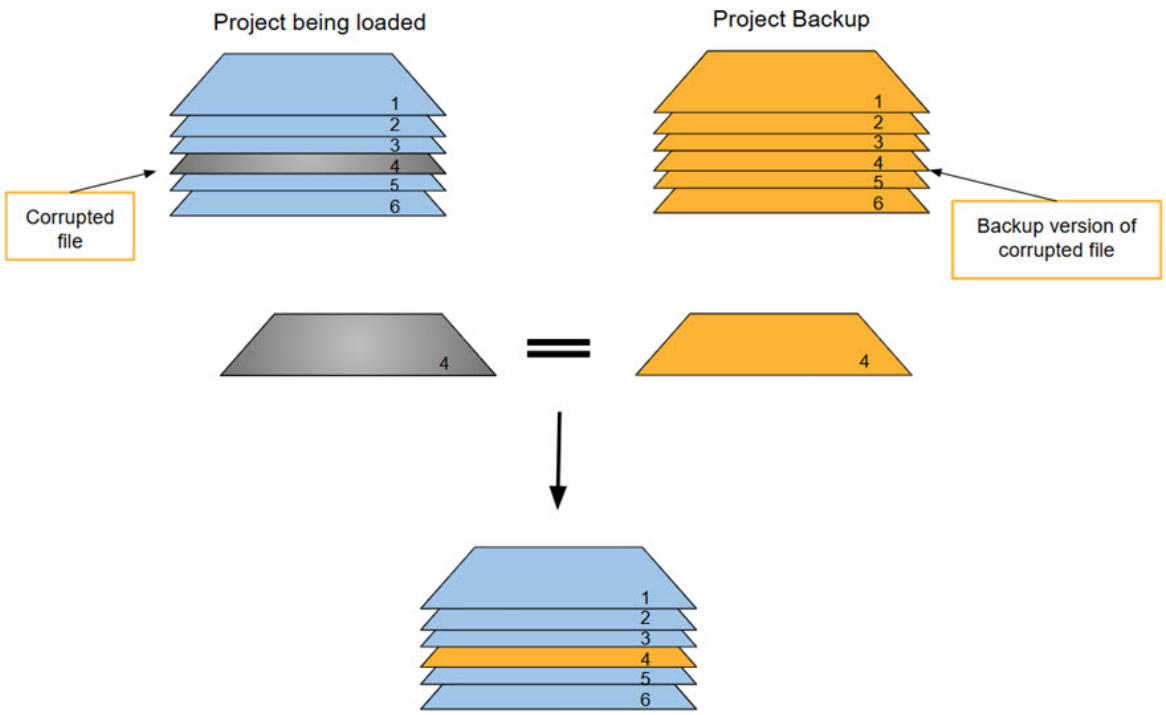
   This list is organized into date-time and tag-based sub-menus. You can search via **Version Tags**, **Today**, **Yesterday**, **This Week**, **Last Week**, **This Month**, and **Last Month**. You can still access older back ups by clicking the **Other** option in the Restore menu. This opens a browser in the backup location where you can browse all backups.
3. Roll over the backup to see its details and a thumbnail preview of the project.
4. Click the desired backup to restore the project.



To find a save point that isn't in the current backup location, use **Other** to open a file browser. You can also access this file browser by selecting a project and click the Restore button in the Projects tab.

> **Note:** Mari will autosave your project, but these won't be displayed in the Restore menu. To access autosaves, use the **Other** option in the context menu, or click the **Restore** button at the bottom of the Projects tab to open a file browser at the backup location. You can then select an autosave backup. All backups will be listed in this location as ***mri** files.

If you're loading a project with a backup and there is a corrupted file, Mari will try and match the corrupted file with another backup version to repair the corruption so you can continue working on your project as expected.



Studio Backup Integration

A backup location and Restore actions can also be set up for a studio pipeline. To run the backup system in a pipeline, you will need to stop the automatic backups. To do this, set the environment variable MARI_DONT_AUTO_TRIGGER_PROJECT_BACKUP=1.

Other manual scripts can be run with the backup once this environment variable is set by using the Python API. For example, certain tags can be restricted from the Mari UI.

To learn more, read the Python API documentation at **Help** > **SDK** > **Python** > **Documentation**.

Archiving Your Project

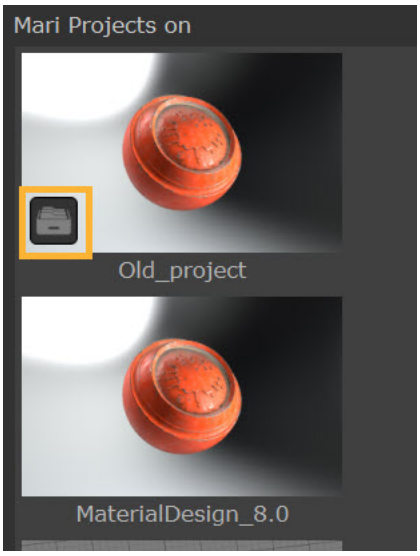
You can archive your Mari projects to store all the projects' information or to share them with other artists.

To archive your project, do the following:

1. Save and close your project before archiving it by clicking on **File** > **Save**, then **File** > **Close**.
2. In the **Projects** tab, if your project is not already selected, select your project.

3. At the bottom of the application, click **Archive** or right-click on your project and select **Archive**.
   The **Archive** dialog opens.
4. Browse to the location where you want to archive your project.
5. Click **Save**.
   Your project is archived to the location you selected as an **.mra** file.

The archived project will have an archive icon next to it in the Projects window.



You can hover over the project icon to view the project's most recent archived path and date, and whether the last archive is up to date. If the archive is no longer up to date, the icon will be removed.

Note: The only supported format for archiving projects is **.mra**.

Video: To learn more about how to archive projects, watch Archiving Mari Projects.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Article: Take a look at the Best practices for sharing Mari archives article.

Customizing the View

This section covers how Mari displays the model on screen.

You can set:

- how the model rotation works - the model movement speed, friction, and whether it uses momentum (see Appendix A: Shortcuts for model rotation shortcuts).
- the background displayed behind the model - either a color, a gradient, or an image.
- the color for the paintable area (when it displays).
- the quality settings for the display (how much detail to show).
- whether to show the wireframe model, and how to display it.
- in the UV view, whether to show gridlines or labels.

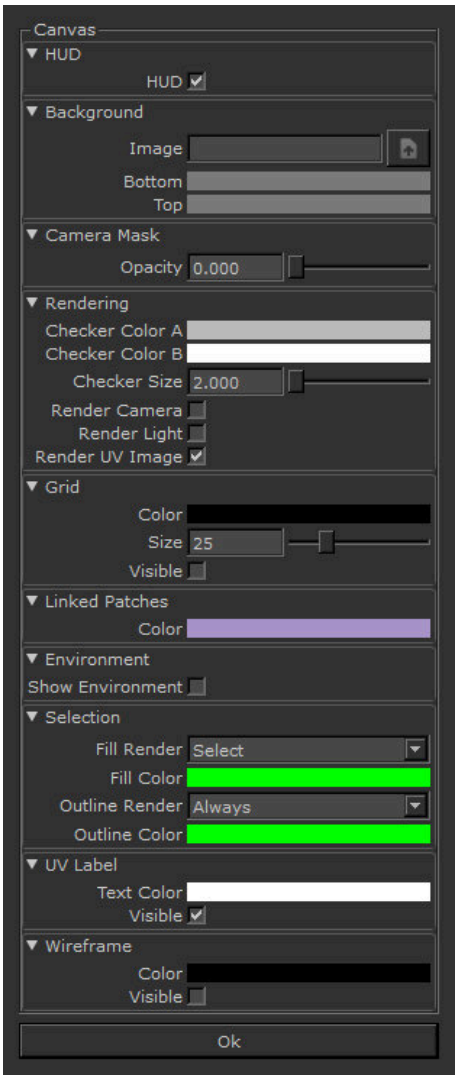
You can also adjust the focus of your canvas by hovering your cursor over the point on the model that you want to focus on and pressing **F**. The canvas centers on that part of the model at the current view size and angle. If the cursor is hovering over the canvas where no part of the model is present, pressing **F** instead brings the entire model to focus.

Setting the Main View Options

You can set the view options separately for the UV and 3D views.

1. Switch to the view you want to set the view options for.
2. Either:
   - right-click on the canvas and select **Display Properties** from the dropdown menu, or
   - from the **View** menu, select **Display Properties**.

   The **Display Properties** dialog displays.



3. To turn the onscreen display of the project information (Current Tool, Tool Help, Current Channel, and so on) off, uncheck the **HUD** box.
4. Under the **Background** section, set the window background.

To set the background to...	do this...
an image	enter the path to the image in the **Image** field. Use the button next to the field to open a dialog to find your image.
a single color	set the **Bottom** and **Top** fields to the same color. Click on the fields to set the color.

To set the background to...	do this...
a gradient between two colors	set the **Bottom** field to the color for the bottom of the window, and the **Top** field to the color for the top of the window. Click on the fields to set the color.

5. To set the global setting for the canvas **Environment** light background image, set the **Show Environment** checkbox. This control allows you to set the canvas background image from the environment light independent of the display settings within the light.
6. If you want to display a reference grid in a 3D view, set the details under the **Grid** section.

   The **Visible** switch turns the onscreen grid on and off, while the other fields let you set the **Color** and **Size** of the grid displayed.
7. To show the wireframes on the model, turn the **Visible** option on under **Wireframe**.

   You can set the wireframe color in the **Color** field.

> **Tip:** You can turn the wireframe display on/off using the button on the **Canvas** toolbar, or using the keyboard shortcut **Shift**+**W**.

> **Tip:** If the **WireframeColor** has a high alpha value, it may be hard to see onscreen. If you have the wireframe turned on but cannot see it on your model, check that the **WireframeColor** isn't too transparent.

8. If you want to adjust the **Camera Mask Opacity**, use the text field or the slider to select a value between 0 and 1.
9. To select the color that you want shown when selecting **Linked Patches**, click on the color swatch to select from the color picker. You can also select the **Linked Patch** color from within the **Patches** palette.
10. Under the **Rendering** section, you can control which additional parts of the scene display onscreen:
    - **Checker Color A** - changes color A of the transparency checkerboard to the selected swatch color.
    - **Checker Color B** - changes color B of the transparency checkerboard to the selected swatch color.
    - **Checker Size** - changes the individual transparency checkerboard squares to the selected size using either the text field or the slider. The default value is set to 10.
    - **Render Camera** - shows the camera view onscreen.
    - **Render Light** - shows the lights as globes on the main canvas.

- **Render UV Image** - if this is off, the UV view shows the patches as they appear as part of the model. If this is on, the UV view shows the patches as they appear when exported (that is, as square images, with overpaint areas around the corners of the patches).

> **Tip:** You can also turn **Render UV Image** on/off using the button on the **Canvas** toolbar.

11. If you're setting properties for the UV view, under the **UV Label** section, set the details for the text labels shown over the individual patches:
    - **Text Color** - click the swatch to set the color for the text on the labels.
    - **Visible** - uncheck this to stop showing the text labels in the UV view.
12. If you're setting properties for 3D views, under the **Selection** section, set the details for highlighting shown on selected areas:
    - **Fill Render** - when to show the highlighting in the middle of the selected areas. One of:
        - **Always** - Mari shows the highlighting the whole time the area is selected.
        - **Select** - Mari only shows the highlighting while you are actually selecting the area.
        - **Never** - Mari does not show the selection highlighting.
    - **Fill Color** - click the swatch to set the highlight color for the middle of the selected areas.
    - **Outline Render** - when to show the highlighting around the outline of the selected area. You can set this to **Always**, **Select**, or **Never**, as for **Fill Render** (above).
    - **Outline Color** - click the swatch to set the highlight color for the middle of the selected areas.
13. Click **OK** to save your settings.

Navigating the Mari Canvas

Mari includes features for 3D navigation to view your model from various angles and perspectives.

> **Video:** Watch this video to learn how to Navigating the Mari Canvas.
> This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

To...	
Zoom in and out	**Alt+Ctrl/Cmd**-click and drag, OR Right-click and drag
Pan left and right	**Alt+Shift**-click and drag, OR Middle-click and drag
Orbit the model	**Alt**+left-click and drag
Roll the camera	**Alt+Ctrl/Cmd+R**+left-click and drag

Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.

You can use key mapping to assign the mouse modifier to another key, such as the (**Super** or **Meta**) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

Tip: By default, Mari uses momentum when you are moving the view. When you let go of the mouse button, the model slowly coasts to a stop. You can turn this behavior off so that the model only moves when you are holding the mouse button down.

To do so, change the settings in the preferences by unchecking **Momentum Enabled** in **Edit** > **Preferences** > **Navigation**.

Tip: By default, **Pause Canvas Rendering** is enabled in the **Mari Preferences** > **GPU** tab. This allows for faster UI interaction by pausing canvas rendering while the mouse cursor is outside the canvas area.

You can turn this functionality off by unchecking **Pause Canvas Rendering** in **Edit** > **Preferences** > **GPU**.

The navigation toolbar also provides the ability to control panning, zooming, rolling, and orbiting, with various settings for each. You can easily access the toolbar, shown below, at any time using the following buttons:

• Reset navigation settings to default.

Article: Take a look at the Differences between 'Reset navigation settings to their default' and 'Reset Camera' article.

- Enable panning to move the model on the screen.
- Enable zooming to zoom in or out on the model.
- Enable rolling to turn on the momentum system.
- Enable orbiting mimics the orbiting system in Maya and allows you to move the camera around the model.
- Disable rotational snapping gives options to turn on and change the angle at which the camera view snaps to (45 degrees or 90 degrees).

You can also use the keys **1-6** to switch between preset views (left, right, top, bottom, front, and back).

In the UV view, you can also rotate or reset the model for easier viewing and painting. Press **1** to reset the model back to its default position. Press **2**, **3**, and **4** to rotate the model counter-clockwise in 90-degree increments from the default position.

Press **Home** to cycle between hiding all palettes, hiding docked palettes and showing hidden palettes.

Use the tabs at the top of the canvas to switch between:

- **UV** - a grid view showing the UV patches.
- **Ortho/UV** - a split screen showing the UV patches and the orthographic view.
- **Perspective** - a perspective camera view, or
- **Ortho** - an orthographic camera view.
- **2D** - 2D paint mode.

Tip: You can also use the , , and toolbar buttons or the **F10**, **F11**, and **F12** keyboard shortcuts to activate a particular view on any tab.

Configuring Your Workspace

The Mari workspace is completely configurable. You can customize how palettes display, your own personalized sets of brushes and colors, and shortcut keys.

Palettes

Mari uses a number of palettes - windows that you can move around your workspace, with specific information and tools for working on your project. For example, the **Channels** palette shows all the channels in the project, and has controls for editing the channels.

To open palettes:

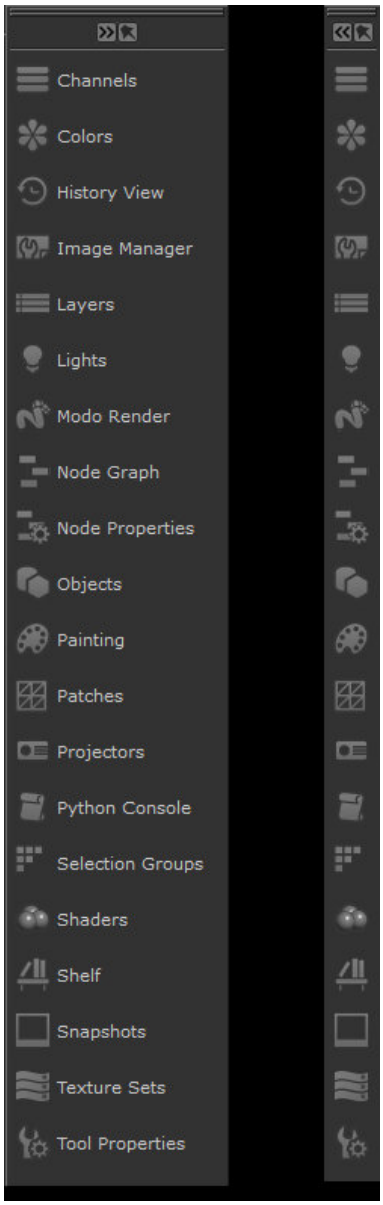
From the Palettes toolbar, select	***or* Select from menubar**	***or* right-click on the menubar or toolbar and select**
<palette name>	**View** > **Palettes** > <palette name>	<palette name>

To move, resize, and arrange palettes:

To...	**Do this...**
Undock and move a palette	Grab and drag it by its title bar to anywhere on your screen. (Or click to undock, and then drag by its title bar).
Dock a palette to the window	Move it to one of the edges of the Mari window. A dotted line displays along the edge where it can dock.
Dock a palette to another palette	Move it over another undocked palette. A dotted line displays where it can dock.
Resize a docked palette	Grab and drag the dotted bar along the bottom or side.
Resize a floating palette	Drag an edge or corner.
Stack palettes	Drag and drop them directly on top of one another. Tabs appear for selecting which of the stacked palettes to display.
Hide/show and expand	Press the **Space** bar while hovering over the canvas to toggle between showing and hiding the currently open palettes. You can also press the **Space** bar while hovering over a palette to expand it. **Note:** You can't expand a palette if you selected an item in it.

Palettes Toolbar

The **Palettes** toolbar allows quick access to your Mari palettes. The description of the **Palettes** toolbar and its controls can be found below.



Displaying the Palettes Toolbar

By default, the **Palettes** toolbar displays on the right of the application, but you can also place the **Palettes** toolbar on the left, or at the top or bottom like any other Mari toolbar. You can choose to display the **Palettes** toolbar with its icons only (see image above on the left) or the extended **Palettes** toolbar with its

icons and palette names (see image above on the far left) by toggling the Show/Hide buttons. You can display the extended **Palettes** toolbar only on the left-and right-hand side of the application.

Opening/Maximizing Palettes

Palettes open in one of two modes from the **Palettes** toolbar; Pinned or Floating. The palettes have three open states; docked, floating, and temporarily floating states. From the **Palettes** toolbar, choose in which mode to open palettes and then click on them:

- Use the Pin Palette mode to open palettes in a docked and/or floating state so that they remain visible while you work on your project. To find out how to dock or make your palettes float, see Choosing and Organizing Palettes.

  The palettes open where they were last closed in docked or floating state.

> **Note:** In Pin Palette mode, palettes do not display the Pin/Unpin Palette button as switching currently-opened, docked palettes to temporarily floating state is impossible.

- Use the Float Palette mode to open palettes in a temporarily floating state for you to edit some options before resuming painting. Your palette opens when you click on its icon or name and disappears when you move your cursor away from the palette.

  If a palette is already docked in the application, it maximizes or minimizes it. If a palette is stacked under another palette, it displays it on top of the stack.

  While in Float Palette mode, you can open a palette in a temporarily floating state and then decide to pin the palette using the Pin/Unpin Palette button.

> **Note:** Switching between Pin Palette and Float Palette modes doesn't affect the palettes that are already open.

When a palette is closed, its background in the Palettes toolbar is dark gray, and when it is open it is darker gray.

Closing/Minimizing Palettes

The behavior for closing palettes varies depending on the palette opening modes and palette open states:

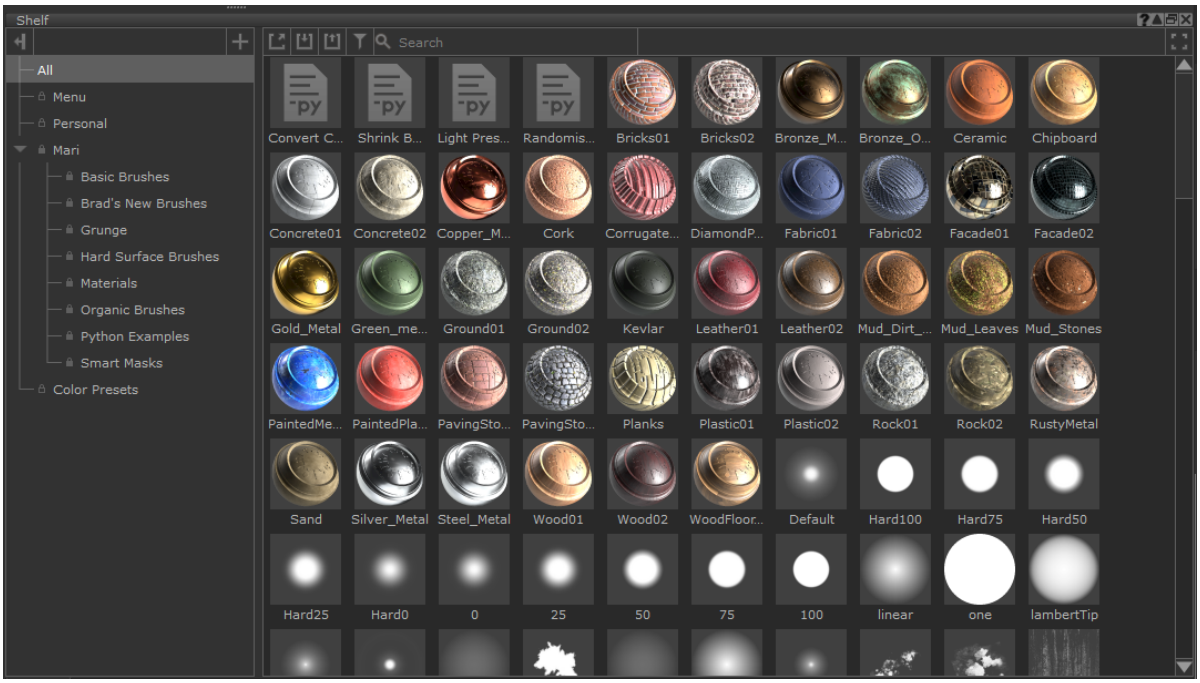
- Pin Palette mode - clicking the palettes' icon or name closes floating palettes. Clicking the palettes' icon or name minimizes the docked palettes.

- Float Palette mode - clicking the palettes' icon or name closes floating and temporarily floating palettes. Clicking the palettes' icon or name minimizes the docked palettes.

> **Note:** To know more about organizing your palettes, see Choosing and Organizing Palettes. For more details see the *Palettes Toolbar*.

Shelves

The **Shelf** palette lets you store customized "shelves" of brushes, colors, and images. You can also share them with others.



Mari has the following kinds of shelves:

- The **Menu** shelf stores seven items available from the F9 on-screen menu. These items must be numbered from 1-7 to appear.
- The **Personal** shelf stores items you select and configure for all projects.
- The **Project** shelf stores items you select and configure for the current project.
- The **Basic Brushes** shelf stores a set of predefined basic brushes.
- **Brad's New Brushes** stores a set of predefined brushes created by Bradford DeCaussin, which can be used to add small detail to your textures. For example, use the **moleAttack** brush to add spots and blemishes to skin.

- **Brad's Movie Brushes** is another set of predefined brushes by Brad DeCaussin, which can be used to add detail to your textures. For example, experiment with **cleverGirl** for a reptilian creature's skin.
- **Grunge** - a set of high-quality grunge textures created by industry professionals.
- The **Hard Surface Brushes** shelf stores a set of predefined hard surface brushes.
- The **Materials** shelf stores a set of predefined materials based on the Principled BRDF shader model. See Managing Materials in Mari for more information.
- The **Organic Brushes** shelf stores a set of predefined organic brushes.
- The **Python Examples** shelf stores predefined python scripts to perform certain useful functions.
- The **Smart Masks** shelf stores ready-to-use masks and grunge maps, and a set of utilities to edit them. See Smart Mask Node for more information.
  Note: this shelf includes items which used to be called Custom Procedurals.
- The **Color Presets** shelf stores colors that you have saved.
- Customized shelves you create store items you select and configure for all projects.

Tip: See Ready-to-use Shelf Content to learn more about what is available in the Shelf.

Video: To display the Shelf menu when painting, press and hold **K**. Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Customizing Shelves

To add an item (brush, color, or image) to a shelf, drag it onto the shelf.

Tip: You can also add nodes to the shelf by packaging them and importing them. See Adding Nodes to a Shelf.

Configuring the Lighting

Mari has five lights; four basic point lights, which appear on the drawing canvas, and one environment light.

Lighting Modes

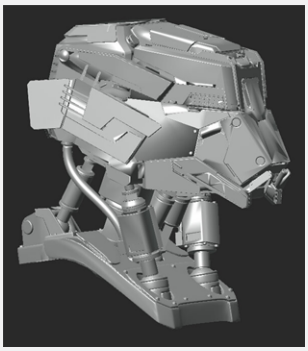
Mari comes with five lighting modes for **lighting** the model:

- **Flat** - a flat, uniform, directionless light.
- **Basic** - basic diffuse lighting without specular.
- **Basic with Shadows** - basic lighting with diffuse and specular lighting, as well as shadows.
- **Full** - full lighting with diffuse and specular lighting.
- **Full with Shadows** - full lighting with diffuse and specular lighting, as well as shadows.

Video: To learn how to control the lighting used to texture Mari assets, watch Understanding Lighting in Mari.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

You can change the lighting on the main mesh by clicking on the **Project Controls** toolbar:

Flat	Basic	Basic with Shadows

Flat	Basic	Basic with Shadows
Full	**Full with Shadows**	
		

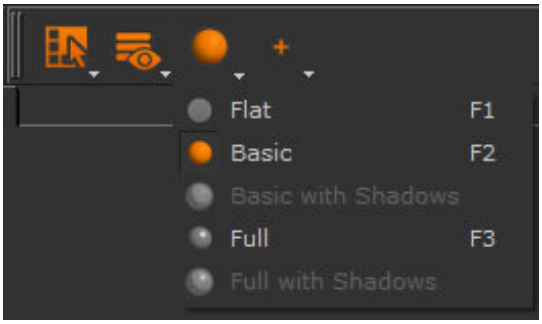
In **Basic** and **Full** modes, you can set basic point lights or an environment light. This lets you control the intensity and direction of the lighting from different directions relative to the model.

Mari only stores custom light settings as part of the project (they cannot be independently saved to a file).

To enable shadow processing, in the **Mari Preferences Dialog** (see the Mari Preferences Dialog) in the **GPU** tab, enable the **Shadow Maps** > **Allowed** checkbox. By default this preference is disabled. For more information on shadows, please refer to Shadow Settings in the Working with Objects section.

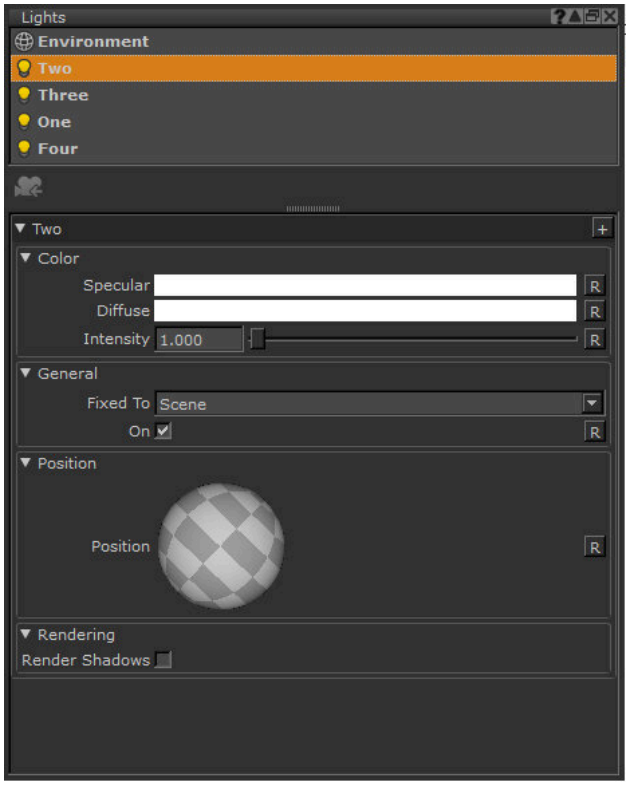
To Configure the Lighting

1. From the **Project Controls** toolbar, switch between **Flat**, **Basic**, **Basic with Shadows**, **Full**, and **Full with Shadows** lighting:



Note: If the **GPU** > **Shadow Maps** > **Allowed** option is turned off in the **Mari Preferences Dialog** (see the Mari Preferences Dialog), the **Shadows** icon cannot be toggled and is turned off in the **Lighting** toolbar.

2. To set individual lighting directions, make sure you're in full lighting mode. If it's not already visible, open the **Lights** palette by clicking **View** > **Palettes** > **Lights**:



The top half displays the lights available, and the bottom shows the details of the selected light. The lights currently in use have a “lit bulb” icon.

Tip: Mari has four basic point lights and an environment light. You cannot add or delete these lights. If you don't want a particular light, turn it off. As Mari saves the lights with the project, if you need the same lights in another project, you need to set them up separately.

3. To turn individual lights on or off, click on the icon in the list, or disable the **General** > **On** checkbox.

The basic lights work a little differently to the environment light in the **Lights** palette. To configure the basic lights, follow the steps below:

4. To change a light, first select it. In the bottom half of the palette, set the light's:

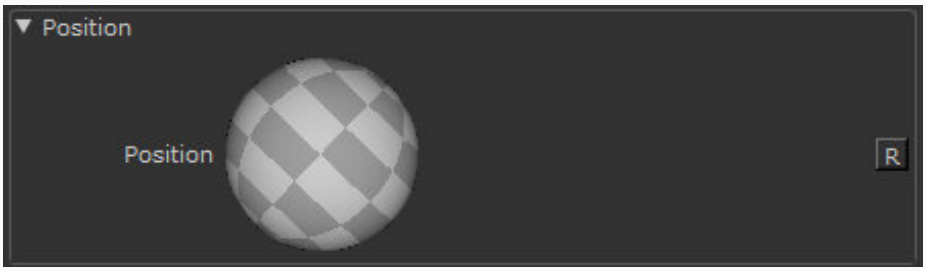
- **Color** - you can set separate values for the **Specular** (shiny highlights) of the light and **Diffuse** (the color the light shines on surfaces), and change the **Intensity** value for the light.
- **Fixed To** location - whether the camera is fixed relative to the **Scene** or the **Camera**.

This...	means...	Example
Scene	The light always shines on the same part of the model, no matter how you move the view.	If you only have one light on, pointing at the left side of the model, the right side is always in darkness, no matter how you move the view.
Camera	The light always shines from the same direction relative to the camera view. This means that it illuminates different parts of the model depending on how the model appears in the camera.	If you only have one light on, pointing from the left of the camera, it illuminates whatever part of the model(s) is visible on the left side of the current view.

Video: Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Moving a Light Around

Select...	then...
The **Transform Selected Objects** tool	click on the light and drag it around the canvas, or use the transform handles to adjust it.
The sphere control in the **Lights** palette	click and pull the sphere around to move the light.

Select...	then...
	
The **Move to Camera Position** icon	the light is repositioned to your current camera position.

You may need to zoom out from your object in order to see the lights.

Note: The **Move to Camera Position** icon can be used in both perspective and orthographic view; however, be aware that this feature does not account for zooming of the camera in the **Ortho** view.

Rendering Shadows

1. To enable casting shadows from the selected light, in the **Lights** palette, click the **Rendering** > **Render Shadows** checkbox. This does not impact the light itself, but the model casts shadows depending on the light's position.
2. More options for shadows and depth projection settings are located in the **Mari Preferences Dialog** (see the Mari Preferences Dialog) under **GPU** > **Shadow Maps**. The settings in the **Preferences** allows you to trade off the quality of the rendered shadows and the amount of GPU memory and processing required. Increasing the resources available for shadow rendering decreases the resources available for other operations, which may decrease Mari's performance.

Tip: Switch between the different lighting modes. Then try customizing a preset light. Play with its color settings and orientation. Try using the position sphere to control the light position. Notice the effects on your display?

Environment Light

The **Environment** light uses an image from the Mari Environment library, the **Image Manager** palette, or an imported image file to set environment lighting on the object(s) on the canvas. To configure the **Environment** light:

1. Specify a texture to use for the environment. You can either:
   - select an image from the Mari Environment library by clicking the blank image space under **Environment** > **Texture** > **Image**
   - click either of the arrow keys under **Environment** > **Texture** > **Image**,
   - select an image in the **Open an Image** dialog by clicking the load icon under **Environment** > **Texture** > **Image**,
   - drag an image from the Image Manager to the blank image space under **Environment** > **Texture** > **Image**, or
   - drag an image from a system file manager to the blank image space under **Environment** > **Texture** > **Image**.
2. To adjust the **Environment** light, set the light's:
   - **Color** - you can set the **Intensity** value for the light.
   - **Fixed To** location - whether the camera is fixed relative to the **Scene** or the **Camera**.

This...	means...	Example
Scene	The light always shines on the same part of the model, no matter how you move the view.	If you have an environment image that is bright on one side, that side of the model is brightly lit and the other side is dark, no matter how you move the view.
Camera	The light always shines from the same direction relative to the camera view. This means that it illuminates different parts of the model depending on how the model appears in the camera.	If you have an environment image that is bright on one side, whatever part of the model visible on that side in the current view is illuminated.

   - **Resolution** - this adjusts the resolution of the light to be either **64**, **128**, or **256**. This only applies to the light and not the background (canvas) image.
3. To move a light around, set the **Up Axis** to **Axis X**, **Axis Y**, or **Axis Z**, to adjust which part of the environment image is interpreted as up. This adjusts the background (canvas) image as well. This is

especially useful if the environment image is read into Mari wrong-side up and you want to adjust the direction of the environment lighting.

4. To specify whether you want the environment light image to show on the canvas click on the dropdown menu under **Environment** > **Texture** > **Background** and select **Show** or **Hide**.
5. To set how the background (canvas) image appears, adjust the following under **Environment** > **Texture**:
   - **Rotation** - rotates the image on a 360 degree scale so that you can adjust how the background image appears on the canvas against the object(s).
   - **Cubemap Type** - choose from **None**, **LatLong**, or **Cross** to set how 2D image is converted to 3D. Mari attempts to automatically interpret the **Cubemap Type** but if the image is a **.dds**, it isn't necessary to set this option.

> **Note:** When exporting **.dds** files, they should be encoded as 8.8.8.8 ARGB files to avoid any complications when bringing them into Mari.

   - the **Blur** - this adjust how much the background (canvas) image is blurred, if at all. For low resolution textures, the blur may not appear as smoothly as you might like. Try using a higher resolution texture to improve this.
6. To set the texture animation for the background (canvas) image, adjust the following under **Environment** > **Texture Animation**:
   - **Animation** - whether the background (canvas) image is **Static** or can **Rotate**.
   - **Speed** - sets the speed of the animation rotation of the background (canvas) image.

> **Note:** If the **Background** is set to **Hide**, or the **Speed** is set to 0.000, it doesn't matter if the **Animation** is set to **Rotate**, it won't appear to be moving. Ensure that both of these controls are set correctly in order to see texture animation.

Setting the Perspective Camera Details

For the perspective you can set the near and far clipping planes. Anything closer than the near clipping plane or farther than the far plane is not displayed.

You can also set the field of view. This controls the amount of distortion applied to generate the perspective.

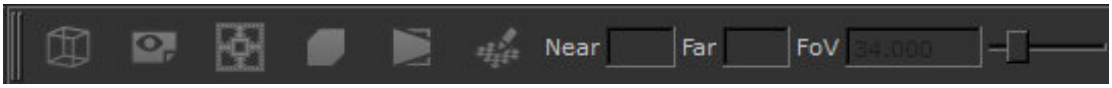
Field of view set to 20

Field of view set to 120

Tip: You can also have black bars appear in the Canvas to indicate the aspect ratio of the perspective
camera. Select **View** > **DisplayProperties** and set **Camera Mask Opacity** to anything other than 0.

To set the perspective camera details:

1. Click on the **Perspective** tab to switch to the perspective camera.
2. If it's not already visible, open the **Canvas** toolbar.



   To open the toolbar, right-click in the toolbars area at the top of the screen and select **Canvas** from the dropdown menu.
3. Enter values for the **Near** and **Far** clipping planes.

   Mari only displays parts of the scene that fall between these two planes. Anything closer than the Near plane, or further than the Far plane, is not shown.
4. Set the value for the field of vision in the **FoV** field. This controls how much Mari distorts the view when applying perspective. At 0, the perspective camera gives exactly the same view as the Ortho camera. As the value increases, the distortion increases.

Tip: You can also change view mode by clicking the **Ortho**, **Perspective**, and **UV** buttons, or pressing **F10**, **F11**, and **F12** respectively.

Configuring Navigation

The navigation has preferences that allow you to change, among other aspects:

- momentum
- flick speed
- friction
- rolling
- orbiting

See Appendix A: Shortcuts for model rotation shortcuts. To configure navigation settings:

1. Go to **Edit** > **Preferences**, and select the **Navigation** tab.
2. To set the controls, select one of the options below, from the **Control Type** dropdown menu:
   - 3ds Max
   - Houdini
   - Lightwave
   - Mari
   - Maya
   - Modo
   - Nuke

   These controls allow you to set what type of control options, shortcuts and preferences you want to use.
3. To manually change how the model movement and navigation works, set the following:
   - **Momentum Enabled** - if this is enabled, the model has momentum, so when you let go of the mouse button, the model slowly coasts to a stop. If this is off, the model only moves when you are holding the mouse button down.
   - **Minimum Flick Speed** - the minimum speed at which you can move your model around. The lower this is set, the more sensitive the movement controls are.
   - **Friction** - affects the momentum of your model as you move it around the canvas. Decreasing this value allows movement to retain its momentum for longer. Increasing the value arrests momentum faster.
   - **Lock To World Up** - if this is enabled, the navigation is locked with the y axis in place so that the "world up" view cannot be modified.
   - **Center Mode** - a dropdown menu that allows you to select the orbit view center: Look At, Objects, Origin, or Selection.

- **Circular Motion Enabled** - if this is enabled, you can roll by making a circular motion around the center of the canvas. If disabled, rolling is achieved by simply making a horizontal motion across the canvas.

Trackball configuration

Mari supports trackball configuration. To set this up, navigate to **Edit** > **Shortcuts**, and use the **Search Actions** field to locate the **Trackball** setting.

Bind your preferred key combination by double-clicking on the **Shortcut** column and type your preferred keyboard shortcut for trackball. If you already have a function assigned to that shortcut, a dialog appears to confirm whether you want to overwrite the existing configuration.

To configure your trackball settings (such as trackball speed and size), go to **Edit** > **Preferences** > **Navigation**.

Note: A mouse button input is required for the shortcut to be successfully registered.

Customizing your splash screen

Mari 8 comes with an alternative splash screen for those who would like to change the image from the spider model.

1. In your file browser, go into the Mari program folder and navigate to **Media** > **Logos**.
2. Locate the file **mari_splashscreen_alt.png**.
3. Rename this file so it is called **mari_splashscreen.png**. You can delete or rename the original spider splash screen image.

The next time you open Mari, the splash screen will be the alternative image.

The alternative Mari 8.0 splash screen image.

Using Shortcuts

To optimize your workflow, you can set up shortcuts for common actions or settings in Mari.

As you have realized, there are usually two or three ways of performing any Mari function. These include:

- **keyboard shortcuts** - pressing one or more keys.
- **mouse shortcuts** (also called "gestures") - holding a key and moving the mouse in a particular way.

As well as the default shortcuts that come with Mari, you can customize your own keyboard shortcuts for any Mari function. Plus, for each project, you can create custom shortcuts for the items and sets within that project or assign multiple shortcuts for the same action.

> **Tip:** You can also change shortcuts through custom configuration files. See the Defining Custom Shortcuts section for details on how to do this.

Some actions can also be canceled once they have been started. To cancel an action, press **Escape** or click the **x** button on the status bar. Canceling actions is not immediate and depends on the complexity of the action. Canceling is limited to:

- applying filters
- ambient occlusion
- autosave
- projecting
    - from the paint buffer
    - through projectors
- turntable rendering
- paint projection.

> **Tip:** Hover over a tool to display its tooltip, including the tool's shortcut if any.

Viewing the Default Shortcuts and Gestures

The list of shortcuts and mouse gestures is in Appendix A: Shortcuts.

The appendix is formatted so you can print out the pages and use them as a reference card. All the default shortcuts in the **Manage Keyboard Shortcuts** dialog box are listed in the appendix.

For the keyboard shortcuts, a plus sign (**+**) means to hold down two keys at the same time. For example, "**Ctrl**/**Cmd**+**C**" means: press **Ctrl** (for Windows or Linux) or **Cmd** (for Mac), and keep holding it down while you then press **C**.

For the mouse gestures, press and hold down the backslash key (****), while drawing the shape illustrated, in the direction indicated - from the ball start point (●) to the arrow.

Tip: The shortcuts for the **Color Picker** and **Select** tools work differently from the other tools. By default, the keyboard shortcuts for these tools only switch to the tool while you hold down the key, and then switch back to the original tool when you release the key. This is to make it easier to quickly pick a color or select a patch while painting with another tool.

If you would prefer, you can set Mari so that quickly tapping the shortcut key for these tools switches "permanently" to the tool, while holding down the shortcut key switches it temporarily (until you let go of the key). To set this, navigate to the Mari **Preferences** dialog box (from the **Edit** menu, select **Preferences**), and on the **Misc** tab, turn off **Tool On Key Held**.

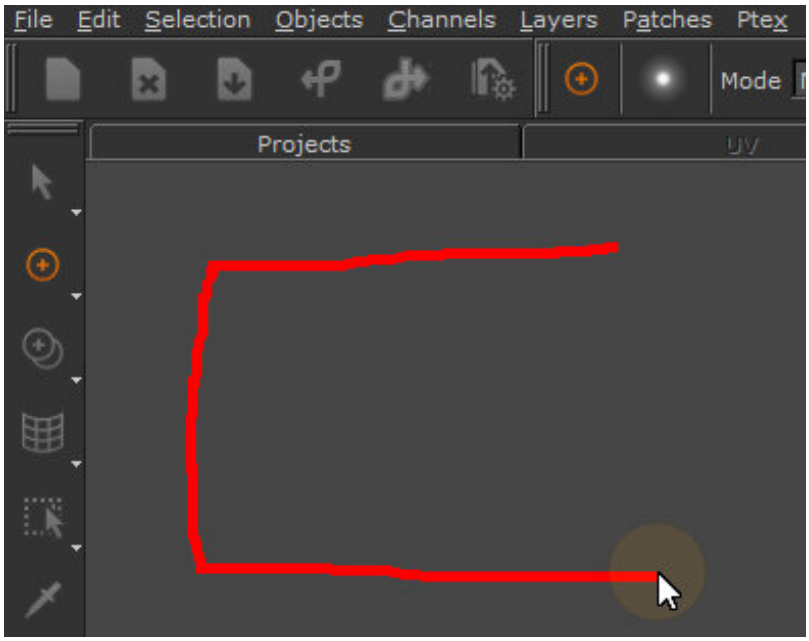
Experiment: Before looking at the mouse gestures, try several of the keyboard shortcuts. You can look up which controls have assigned shortcuts by searching in the **Manage Keyboard Shortcuts** dialog or referring to Appendix A: Shortcuts.

Tip: Hover over a tool to display its tooltip, including the tool's shortcut if any.

Trying the Mouse Gestures

To use a mouse gesture, press and hold the backslash key (****), then click and hold the left mouse button, and move the mouse in the direction indicated.

A red line traces your mouse movement on the canvas.



When you release the mouse button, the tracing disappears and Mari performs the function you selected (in the illustration, the **Clear Painting** function).

Tip: Try several of the mouse gestures. (You may have to paint something before you can try some of them!) The mouse gesture shortcuts are not shown in the **Manage Keyboard Shortcuts** dialog, but can be located in Appendix A: Shortcuts.

Customizing General Keyboard Shortcuts

To add and change the general keyboard shortcuts:

1. From the **Edit** menu, select **Shortcuts**.

   The **Manage Keyboard Shortcuts** dialog box displays many categories of Mari functions.

> **Note:** To know more about this dialog, see the *Manage Keyboard Shortcuts Dialog* section in the *Mari Reference Guide*.

2. Click [+] to expand a category.

   Entries display in columns for the **Name**, **Description**, and **Shortcut** key for each function in that category.

Tip: Drag the edges of column headings to make them wider (so you can see the full contents of the **Description** column, for example). Similarly, drag the edges of the entire dialog box to re-size it as necessary.

3. To add or change a shortcut key, double-click in the **Shortcut** column, and type over the new key. Before you click **OK**, click somewhere to take the focus outside the **Shortcut** column you're editing.

   If the new key you entered is already used, a "warning" icon displays. In this case, the newly entered key doesn't work. This includes parts of any other shortcut that conflicts with a previously set shortcut.

   For example, a new shortcut **Ctrl**/**Cmd**+**R+**left-click can't be set if a shortcut already exists with the definition **Ctrl**/**Cmd**+**R.**
4. If there are no conflicts, press **OK** (otherwise select a different key).

Experiment: Expand and browse through the different categories of Mari functions and their shortcut keys. Try adding shortcuts for functions that don't yet have a key assigned - and then try the shortcut key you've added.

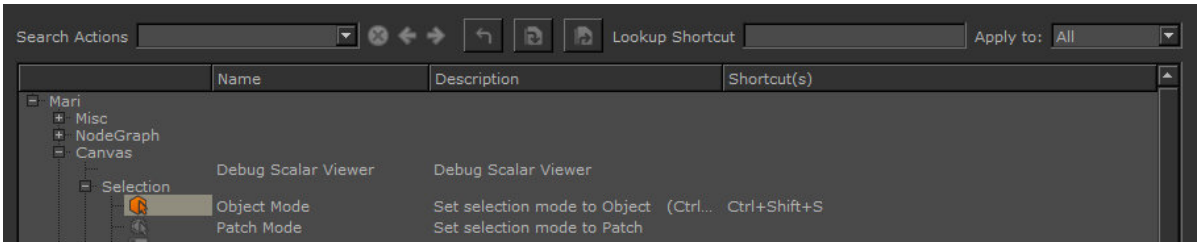
Tip: Hover over a tool to display its tooltip, including the tool's shortcut if any.

Setting Keyboard Shortcuts for Project Items

To set shortcuts for a project item:

1. In the palette showing the item, right-click on the item and select **Assign Shortcut**.

   The **Assign Keyboard Shortcut** dialog box displays, with the item selected in the list.



   Once you have set a shortcut for a project item, it displays on palettes to the right of the item.

2. To remove the shortcut from the item, right-click on it again and select **Assign Shortcut** from the dropdown menu.

   The **Assign Keyboard Shortcut** dialog box opens again, with focus on your item. Double-click on the shortcut key for the item and delete the key.

Quick Palettes and Pie Menu

While painting, you can instantly access the **Shelf**, **Colors**, **Paint Target**, and **Image Manager** quick palettes, as well as the **Pie** menu. You can open these quick palettes by pressing and holding the keys listed below. The quick palettes display under your mouse cursor. As soon as you release the keys, the quick palettes disappear for you to resume painting. This allows you to have easy access to brushes, images, and so on. It also gives you the ability to select, among others, your objects, channels, and paint targets.

Mari contains the following quick palettes:

- The **Pie** menu - a configurable on-screen menu that selects common brush tips or color settings. By default, the **Pie** menu only contains the **Paint Buffer Eraser** tool.
  For more information, see Using the Pie Selection Control.
- The **Shelf** quick palette - stores customized shelves of brushes, colors, and images. You can also share them with others.
  For more information, see the Shelf Palette.

Video: Watch the demo.

- The **Image Manager** quick palette - stores and works with images in your project, including using them for brushes and painting through them onto your model.
  For more information, see the Image Manager Palette.
- The **Colors** quick palette - lets you view and change colors. The bar on the right shows the selected color (without a border - so you can compare directly with what's already painted). You can also select a color and fill your selection(s) by dragging-and-dropping any color swatch on the canvas. The **Colors** quick palette contains four tabs: **Picker**, **Sliders**, **Values**, and **Gray**.
  For more information, see the Colors Palette.
- The **Paint Target** quick palette - quickly selects objects, channels and their paint targets including layers and individual layer masks.
  For more information, see Using the Paint Target Quick Palette.

Displaying the Quick Palettes

To display the quick palettes while painting, use the following shortcuts:

To display the following ...	... With the mouse over the canvas, press and hold
Pie menu	**F9**
Shelf quick palette	**K**
Image Manager quick palette	**L**
Colors quick palette	**J**
Paint target quick palette	**I**

Using the Pie Selection Control

Mari includes a configurable on-screen "pie" control menu that allows you to select common brush tips or color settings. You can add up to seven brush tips or colors to the "pie" control, as well as being able to quickly switch between the **Paint** and **Paint Buffer Eraser** tools. To use pie selection:

1. With the mouse over the canvas, press and hold **F9**.

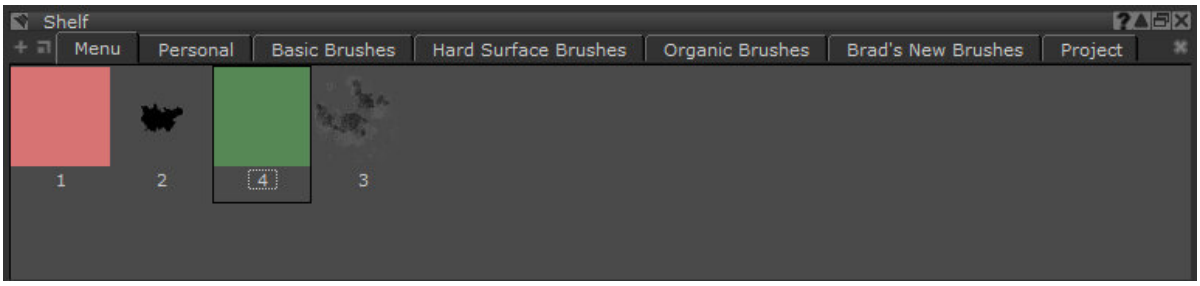
   The pie selection menu displays:

2. To select a setting or tool from the menu, move the mouse over your selection and let go of **F9**.

   You don't need to click – Mari picks up what your mouse is hovering over.

To configure the pie selection control:

1. Add your items (colors and brushes) to the **Menu** shelf.
2. Rename them as 1 through 7.



Mari ignores any other items on the **Menu** shelf, and only displays the items named 1-7.

Tip: For details on how to add items to the **Menu** shelf, see Storing Resources Using Shelves.

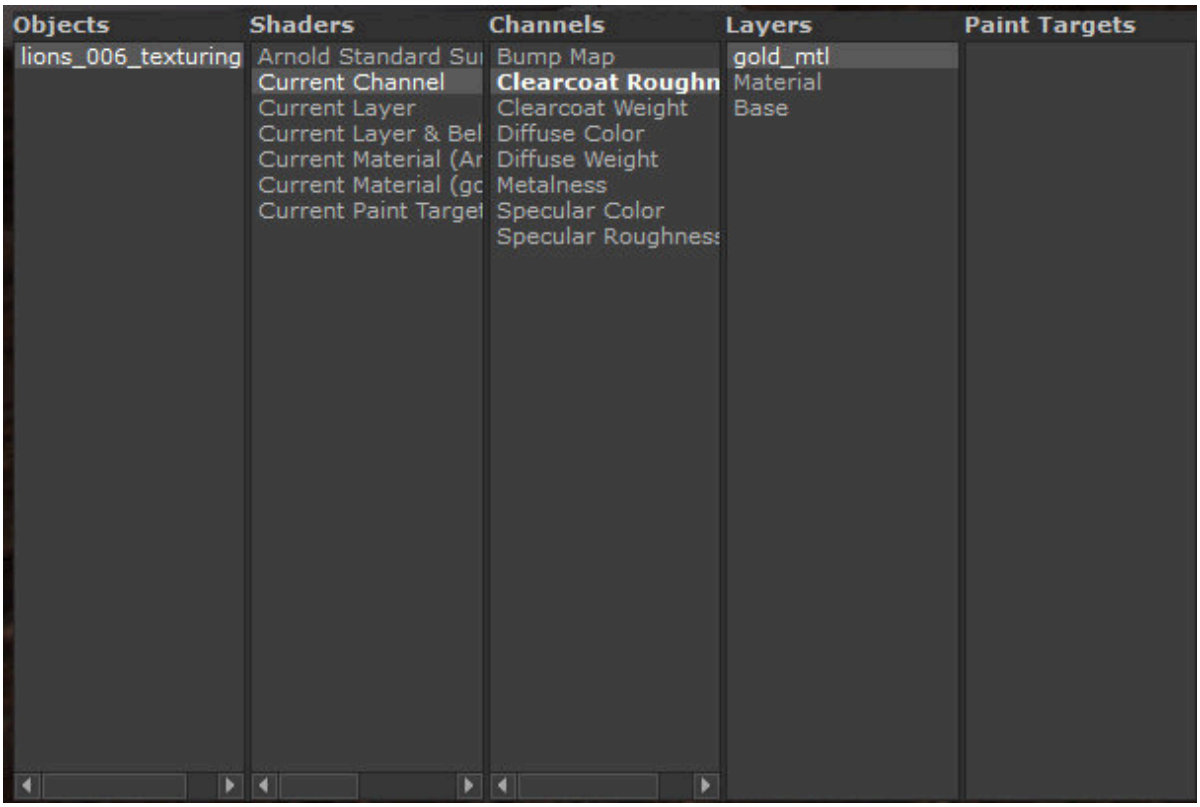
Using the Paint Target Quick Palette

Mari includes a **Paint Target** quick palette that allows you to select your objects, shaders, channels, layers, and paint targets (paintable layers or masks) all at once without having to go in their respective palettes.

To use the **Paint Target** quick palette:

1. With the mouse over the canvas, press and hold the **I** key.

   The **Paint Target** quick palette displays:



2. To select an object, shader, channel, and so on from the quick palette, click the required item and let go of the **I** key.

   Your selection takes effect immediately and the changes are reflected in the respective palettes.

> **Note:** Layers and masks only display if the current selected layer is paintable or has a paintable mask.

Unassigning Shortcuts

You can unassign previously configured shortcuts at any time by following these steps:

1. Go to **Edit** > **Shortcuts**.
2. Double-click on the shortcut you want to unassign and press **Backspace** to unassign a single command.

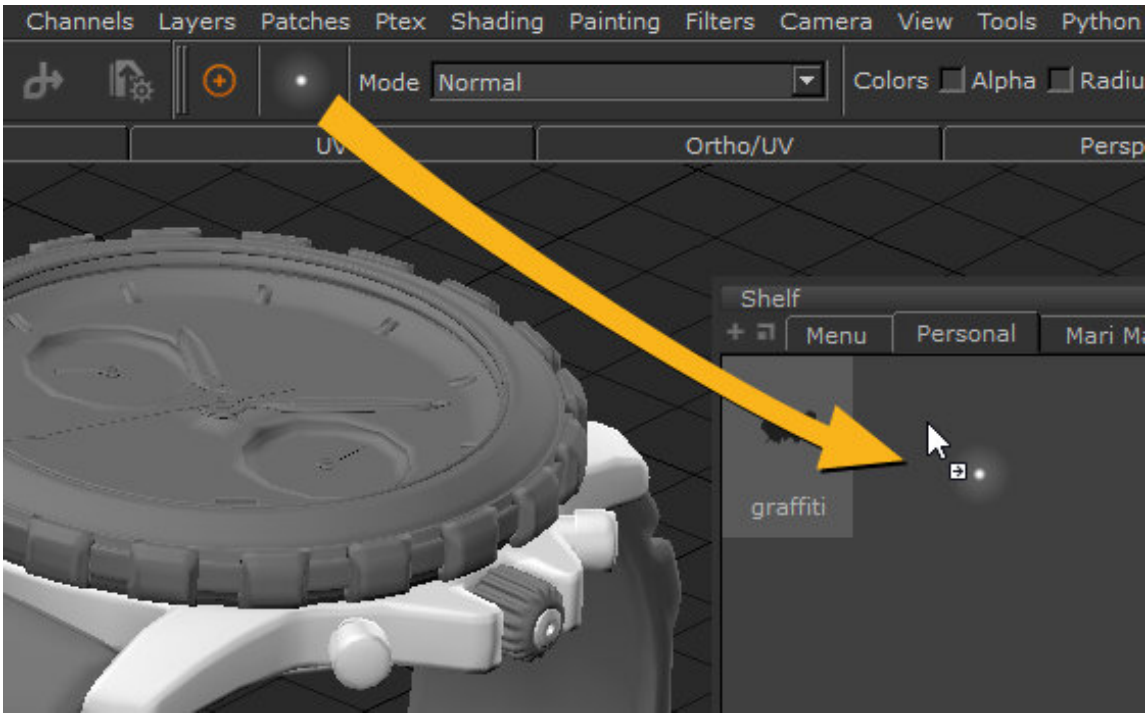
   OR

   Press **Delete** to unassign all commands for that shortcut.

Configuring Brushes

The Mari **Shelf** and **Tool Properties** palettes let you select and customize a wide array of preset brushes, with a wide variety of brush properties you can edit. You can then drag a modified brush from Mari's top toolbar to a shelf for your ongoing personal use, or for a particular project only.

Note: You must drag modified brushes from the toolbar. If you drag a modified brush from an existing shelf, such as **Basic Brushes**, any changes you made are discarded.



Brush properties generally apply to a **splat** (also sometimes called a “stamp”) - one application of the brush tip onto the canvas. A **stroke** is one continuous application of splats. For some properties you can specify **jitter** - a random variation, to approximate how an actual (as opposed to digital) paintbrush might work. See Changing Brush Color Dynamically for information on changing brush color by stroke, splat, or pen pressure.

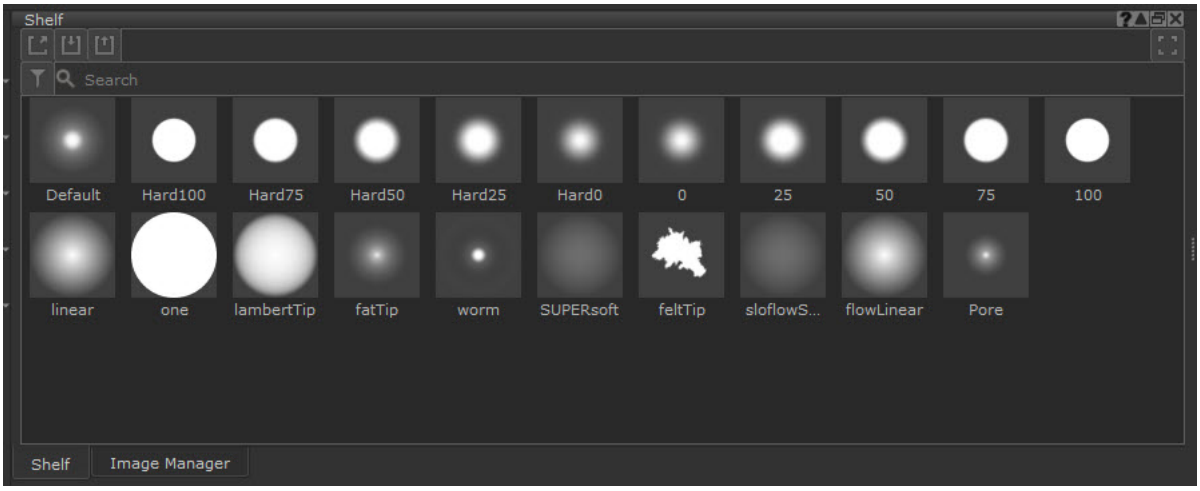
You can also export and import brushes to share with other Mari users, and you can use keyboard-and-mouse shortcuts to modify your brush tips dynamically. See Changing Brush Properties “On the Fly” for more information.

Reviewing the Preset Brushes

Mari comes with a set of preset brushes. To access them:

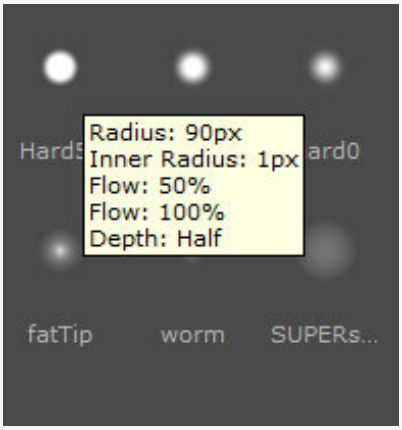
1. Open the **Shelf** palette.

   The preset brushes display and are collected into tabs.



2. Click through each tab and familiarize yourself with the different brushes that are available.
3. On the **Tools** toolbar, select a painting tool such as **Paint**, **Blur**, **Vector Paint**, **Paint Through**, **Gradient**, and **Clone Stamp**.
4. Open the **Tool Properties** palette. See Brush Properties for more information.
5. Try selecting some brushes from the **Shelf** palette and painting with them in the scratch pad of the **Tool Properties** palette.

Tip: When you hover the mouse over a brush, tooltips display key properties:



Tip: Select one of the preset brushes, and play around with the **Radius**, **Opacity**, **Profile**, and **Jitter** settings. Notice how each affects your brush strokes on the canvas or in the **Tool Properties** palette's scratch pad.

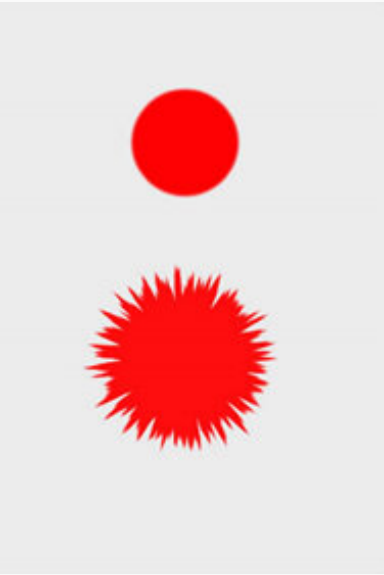
Brush Properties

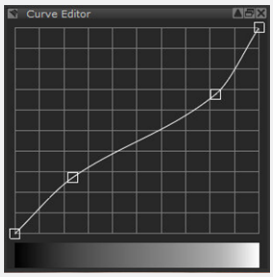
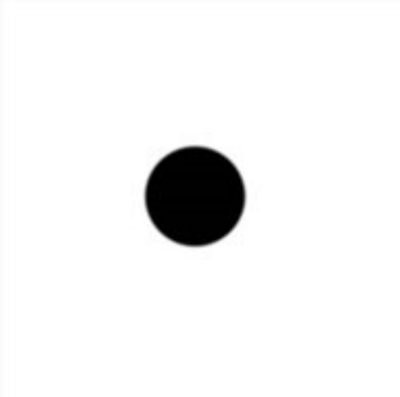
For each brush, Mari specifies several properties that you can view and change.

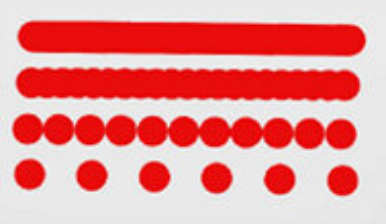
To view the properties of a brush, select a painting tool such as **Paint**, **Blur**, **Vector Paint**, **Paint Through**, **Gradient**, and **Clone Stamp**, then select a brush in the **Shelf** palette, and open the **Tool Properties** palette. The following properties display.

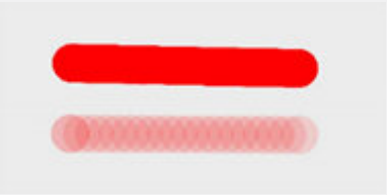
Property	Explanation	Example
Brush Tip		
Bitmap		
Use Painting	Click to use the painting in the paint buffer as a brush tip. Mari prompts you to save it either as an **.exr** or a **.tif** file (which file format is used depends on the support in your graphics card). **Note:** EXR files are 16-bit - most other bitmaps are 8-bit.	
Path	Type the path or click the button to select an existing bitmap to use as the brush tip.	
Source	When **Type** is set to **Bitmap**, select the color information channel from a specified bitmap image. Select from: • Red	If you're using **RGBA** as the **Source**, ensure the correct color space is assigned to your bitmap image using the **Bitmap Color Space** controls.

Property	Explanation	Example
	• Blue • Green • Luminance • Alpha • RGBA	
Invert Alpha	When enabled, any alpha in the brush **Source** is inverted.	
Type	To use an image as the brush tip, you need to set **Type** to **Bitmap**.	
Bitmap Color Space		
Colorspace	Assigns a colorspace to the bitmap of your brush tip.	This control works for bitmaps with **Source** set to **RGBA**.
Raw Data	Enables/disables the use of raw data in a specified bitmap.	Disabled by default.
Scalar Data	Enables/disables the use of scalar data in a specified bitmap. See Color Data and Scalar Data for more information.	Disabled by default.
Geometry		
Vertical Shear	How much to distort the brush tip diagonally, top to bottom (as if pulling opposite corners up and down). **Range:** -1.00 to 1.00, **Reset:** 0.00	
Horizontal Shear	How much to distort the brush tip diagonally, right to left (as if pulling opposite corners right and left). **Range:** -1.00 to 1.00, **Reset:** 0.00	
Vertical Scale	How much to distort the brush tip vertically.	

Property	Explanation	Example
	Range: 0.00 to 2.00, **Reset:** to .50	
Horizontal Scale	How much to distort the brush tip horizontally. **Range:** 0.00 to 2.00, **Reset:** to .50	
Noise		
Octaves	Number of applications of noise. **Range:** 1 to 8, **Reset:** 4	The same brush with the default and maximum noise: 
Frequency	How many spikes per octave. **Range:** 0.00 to 100.00, reset to 1.00	
Scale	How big the spikes are, as a proportion of the radius (1.00 = twice as big as the radius). **Range:** 0.00 to 1.00, **Reset:** 0.00	
Segments	Lines making up the brush outline, to which noise is applied (4=square, 360=circle). **Range:** 4 to 360, **Reset:** 360	
Profile		
Profile	For rendered brushes (see **Brush Properties** above), alpha profile of the brush tip from center to edge - determines how “hard” or “soft” it is.	Hard and soft brush tip profiles:

Property	Explanation	Example
	 In the Curve Editor, to move values along a curve, you can drag any of the points on the curve in any direction, and see the effects in the preview bar below the curve. Click to add a new point, and right-click to invert, select presets, and export and import settings. To remove a point, click on it while holding **Ctrl**/**Cmd**.	
Bit Depth	**Byte** (8-bit), **Half** (16-bit), or **Float** (32-bit) It is generally a good idea to use a brush with the same bit depth as the texture you're painting, to avoid stepping. Note that if you use an 8-bit bitmap, it still paints in 8-bit even if you set the bit depth to **Half**.	
Resolution		
Height Pixels	Set the height resolution of the brush tip.	
Width Pixels	Set the width resolution of the brush tip.	
Brush Settings		
Noise	How much noise to add to each splat, to soften the appearance and reduce	

Property	Explanation	Example
	banding. Lower values give a harder brush, higher values a softer brush. (You can also add noise to the brush tip itself - under the **Noise** section below.) **Range:** 0.00 to 100.00, **Reset:** 0	
Spacing	The space between splats, as a proportion of the width of a splat. 1.00 = side-by-side (so if the tip is a circle, it would look like a string of beads.) .02 = .02 x the width of a splat. **Range:** 0.02 to 5.00, **Reset:** 0.02	A stroke from the same brush with spacing set to .02, .25, .50, and 1.0: 
Clear Brush Value Settings	Click to reset all **Opacity**, **Flow**, **Color**, **Hue**, **Saturation**, and **Value** properties to their default settings.	
Clear Brush Shape Settings	Click to reset all **Position**, **Rotation**, **Radius**, **Steady Stroke**, and **Tilt** properties to their default settings.	
Opacity		
Opacity	How transparent a stroke appears, where 1.00 is totally opaque. **Range:** 0.01 to 1.00, **Reset:** 1.00	A stroke from the same brush with opacity set to 1.00 and 0.50 with no jitter, and 1.00 with maximum jitter:
Pressure Blend	When enabled, more pressure on the graphics tablet increases the opacity of the splat (as calibrated for your tablet).	
Jitter Opacity	Whether to randomly vary the opacity (to the degree specified in **Jitter Opacity Max**).	
Jitter Opacity	How much to randomly vary the opacity	

Property	Explanation	Example
Max	(if **JitterOpacity** is enabled). **Range:** 0.00 to 1.00, **Reset:** 0.00	
Flow		
Flow	Mimics how quickly paint is applied, by setting the maximum opacity in a splat. **Range:** 0.01 to 1.00, **Reset:** 1.00	A completely hard brush tip, full opacity, with spacing of 0.15 and flow at 1.00 and 0.30: 
Pressure Blend	When enabled, more pressure on the graphics tablet increases the flow of splats (as calibrated for your tablet).	
Color		
Pressure Blend	When enabled, more pressure on the graphics tablet paints in the foreground color at the higher end of the **Range** specified (as calibrated for your tablet).	
Range	Sets the range between which the background and foreground colors are blended. The minimum range is pure background color and the maximum is pure foreground color. • **Jitter enabled** - paint in a random blend between the minimum and maximum **Range** values. • **Pressure Blend enabled** - light pressure paints at the lower end of the **Range** and heavier pressure toward the high end of the **Range**. • **Jitter and Pressure Blend enabled** -	

Property	Explanation	Example
	light pressure paints at the lower end of the **Range** and heavier pressure paints in a random blend between the minimum and maximum **Range**.	
Random Jitter	Sets the type of color jitter applied to splats: • **No Jitter** • **Jitter Per Stroke** • **Jitter Per Tip**	See Changing Brush Color Dynamically for more information.
Use Legacy Coloring	When enabled, color jittering is controlled by the alpha channel color jittering available in versions of Mari 4, and older.	
Hue		
Pressure Blend	When enabled, more pressure on the graphics tablet paints in the **Hue** at the higher end of the **Range** specified (as calibrated for your tablet).	
Range	Sets the range between which the **Hue** is adjusted. • **Jitter enabled** - adjusts the **Hue** by a random amount between the minimum and maximum **Range** values. • **Pressure Blend enabled** - light pressure adjusts the **Hue** by the amount at the lower end of the **Range** and heavier pressure by the amount at the high end of the **Range**. • **Jitter and Pressure Blend enabled** - light pressure adjusts the **Hue** by the amount at the lower end of the **Range** and heavier pressure adjust the hue by a random amount between the	

Property	Explanation	Example
	minimum and maximum **Range**.	
Random Jitter	Sets the type of hue jitter applied to splats: • **No Jitter** • **Jitter Per Stroke** • **Jitter Per Tip**	See Changing Brush Color Dynamically for more information.
Saturation		
Pressure Blend	When enabled, more pressure on the graphics tablet paints with the **Saturation** at the higher end of the **Range** specified (as calibrated for your tablet).	
Range	Sets the range between which the **Saturation** is adjusted. • **Jitter enabled** - adjusts the **Saturation** by a random amount between the minimum and maximum **Range** values. • **Pressure Blend enabled** - light pressure adjusts the **Saturation** by the amount at the lower end of the **Range** and heavier pressure by the amount at the high end of the **Range**. • **Jitter and Pressure Blend enabled** - light pressure adjusts the **Saturation** by the amount at the lower end of the **Range** and heavier pressure adjust the saturation by a random amount between the minimum and maximum **Range**.	
Random Jitter	Sets the type of **Saturation** jitter applied to splats: • **No Jitter**	See Changing Brush Color Dynamically for more information.

Property	Explanation	Example
	• **Jitter Per Stroke** • **Jitter Per Tip**	
Value		
Pressure Blend	When enabled, more pressure on the graphics tablet paints with the **Value** at the higher end of the **Range** specified (as calibrated for your tablet).	
Range	Sets the range between which the **Value** is adjusted. • **Jitter enabled** - adjusts the **Value** by a random amount between the minimum and maximum **Range** values. • **Pressure Blend enabled** - light pressure adjusts the **Value** by the amount at the lower end of the **Range** and heavier pressure by the amount at the high end of the **Range**. • **Jitter and Pressure Blend enabled** - light pressure adjusts the **Value** by the amount at the lower end of the **Range** and heavier pressure adjust the saturation by a random amount between the minimum and maximum **Range**.	
Random Jitter	Sets the type of **Value** jitter applied to splats: • **No Jitter** • **Jitter Per Stroke** • **Jitter Per Tip**	See Changing Brush Color Dynamically for more information.
Position		
Jitter Position	Whether to randomly vary displacement from the line of the brush stroke (to the	

Property	Explanation	Example
	degree specified in **Jitter Position Max**).	
Jitter Position Max	How many pixels at most to randomly vary displacement from the line of the brush stroke (if **JitterPosition** is enabled). **Range:** 0 to 1000, **Reset:** 20	
Rotation		
Rotation	The degree to which the brush rotates to match the direction you're painting (if **Align to Stroke** is enabled). **Range:** 0.00 to 360.00, **Reset:** 0.00	
Align to Stroke	Whether the brush tip rotates along with the direction you're painting. (With calligraphy, for example, the tip does *not* align.)	
Jitter Rotation	Whether to randomly vary the rotation.	
Jitter Rotation Max	How many degrees at most to randomly vary the rotation (if **Jitter Rotation** is enabled). **Range:** 1 to 360, **Reset:** 360	
Radius		
Radius	How many pixels at most the radius of splat can be (if **JitterRadius** and/or **RadiusPressure** are enabled). **Range:** 1 to 1200, **Reset:** 100	A stroke with and without radius jitter, where the jitter is equal to the whole radius of the brush:
Squish	How much to flatten the radius. If the brush tip is a circle, 1.0 is a circle, .5 is an oval half as tall as it is wide.	

Property	Explanation	Example
	Range: 0.05 to 1.00, **Reset:** 1.00	
Pressure Blend	When enabled, more pressure on the graphics tablet affects the radius of the splat (as calibrated for your tablet).	
Jitter Radius	Whether to randomly vary the radius (to the degree specified for the **Radius** and **InnerRadius**).	
Inner Radius	How few pixels the radius of a splat can be (if **JitterRadius** and/or **Radius Pressure** are enabled). **Range:** 1 to 1200, **Reset:** 1	
Steady Stroke		
Mode	Whether to use the **Off**, **Distance**, or **Smoothing** mode. **Reset**: Distance	
Delay	Defines how much delay is used for the brush stroke. In **Distance** mode, the **Delay** uses the specified amount of screen pixels, keeping a fixed distance, to define the brush stroke. In **Smoothing** mode, the **Delay** uses the moving average of the cursor movement to define the brush stroke. **Note:** When you move the mouse very fast, Mari gets fewer samples, so the delay looks longer.	

Property	Explanation	Example
	Range: 0 to 200. **Reset**: 0.00	
Tail	Whether to use the **No Tail** or **Meet** options for the finishing strokes. • **Meet** - The brush stroke catches up to the cursor's position when releasing the mouse or pen. • **No Tail** - The brush stroke stops as soon as the mouse or pen is released, with no catch-up. **Reset**: No Tail	
Tilt		
Y Stretch	Stretches the brush splat vertically (Y direction) based on tablet pen tilt.	
X Stretch	Stretches the brush splat horizontally (X direction) based on tablet pen tilt.	
Scratch pad - shows a squiggle painted with the selected brush. You can also paint directly in the scratch pad (the default squiggle disappears while you enter a stroke).		

Note: The texture properties vary from one painting tool to the other and some tools do not have texture properties. Only the tools containing texture properties are listed below.

Property	Explanation	Example
Texture		
Blur Tool		
Blur	Specifies the blur value.	
Paint Through Tool		
Description	Displays the tool function.	

Property	Explanation	Example
Texture		
Name	Displays the tool name.	
Image		
Filename	Sets the filepath of the image file to project as a texture.	
Info		
Description	Displays the colorspace information of the image used to project as a texture.	
Height	Displays the image height.	
Width	Displays the image width.	
Preview		
Preview Alpha	Specifies the transparency of the image previewed over the canvas.	Range: 0.10 to 1.00
Image	Displays a preview of the image.	
Transform		
Image Scale	Sets the scale of the image on the x and y axes. This scales the image relative to the image's size. The button locks the current **Image Scale** values so that any changes to scale are applied proportionately to both axes.	
Pan Lock	Locks the image position relative to the model. If you pan the model, the image that you're painting through pans too.	
Scale Lock	Locks the image size relative to the model. If you zoom the model in or out, the image that you're painting through zooms in or out too.	

Property	Explanation	Example
Texture		
Reset Size	Resets to: • **ToOriginalSize** • **RelativeToPaintBuffer**	
Auto Reset	Sets the **Auto Reset** option to: • **Reset** • **DoNotReset**	
Reset	Resets any transformations applied to the image using the **Image Scale**, **Rotation**, **Scale** or **Translation** options. This **Reset** option is dependent on what you set in the **Reset Size** and **Auto Reset** options.	
Rotation	Sets the rotation of the image used by the **Paint Through** tool.	
Scale	Sets the scale of the image on the x and y axes. This scales the image relative to the canvas size.	
Translation	Sets the translated xy position of the image used by the **Paint Through** tool.	
Gradient Tool		
Type	Selects between **Linear** or **Radial** gradient.	
Out Point	Specifies how far through the gradient the final color reaches.	0.00 to 1.00, higher values imply the end color is reached later.
In Point	Specifies how far through the gradient the start color lasts for (before it starts graduating into the end color).	0.00 to 1.00, higher values imply the start color finishes later.

Property	Explanation	Example
Texture		
End	Sets the final color for the gradient (color that the gradient ends up as).	
Start	Sets the first color for the gradient (color the gradient starts out as).	
Preview		
Preview Alpha	Selects the alpha value displayed by the preview of the gradient floating over the main canvas.	Range: 0.10 to 1.00
Image	Displays a preview version of the gradient.	
Transform		
Pan Lock	Locks the image position relative to the model. If you pan the model, the gradient that you're painting pans too.	
Scale Lock	Locks the image size relative to the model. If you zoom the model in or out, the gradient that you're painting zooms in or out too.	
Reset Size	Resets to: • **ToOriginalSize** • **RelativeToPaintBuffer**	
Auto Reset	Sets the **Auto Reset** option to: • **Reset** • **DoNotReset**	
Reset	Resets any transformations applied to the image using the **Image Scale**, **Rotation**, **Scale** or **Translation** options. This **Reset** option is dependent on what	

Property	Explanation	Example
Texture		
	you set in the **Reset Size** and **Auto Reset** options.	
Rotation	Sets the rotation of the image used by the **Gradient** tool's preview.	
Scale	Sets the scale of the image used by the **Gradient** tool's preview.	
Translation	Sets the translated xy position of the image used by the **Gradient** tool's preview.	
Clone Stamp		
Description	Displays the tool function.	
Name	Displays the tool name.	
Image		
Relative	When enabled, clones from an offset point relative to the source point. When disabled, clones from the source point position.	
Clone From	Sets values (0 to 1) to clone from the image on the x and y axes. The button locks the current **Clone From** values so that any changes to **Clone From** are applied proportionately to both axes.	
Filename	Sets the filepath of the image file used for clone stamping.	
Info		
Description	Displays the bit depth and the image	

Property	Explanation	Example
Texture		
	channels of the image used for clone stamping.	
Height	Displays the image height.	
Width	Displays the image width.	
Preview		
Onscreen Preview	When enabled, displays the image on the canvas.	
Preview Alpha	Specifies the level of transparency of the image preview displayed over the canvas.	Range: 0.100 to 1.000
Image	Displays a preview of the image used for clone stamping.	
Transform		
Image Scale	Sets the scale of the image on the x and y axes. This scales the image relative to the image's size. The button locks the current **Image Scale** values so that any changes to scale are applied proportionately to both axes.	
Pan Lock	Locks the image position relative to the model. If you pan the model, the image that you're painting pans too.	
Scale Lock	Locks the image size relative to the model. If you zoom the model in or out, the image that you're painting zooms in or out too.	
Reset Size	Resets to: • **ToOriginalSize**	

Property	Explanation	Example
Texture		
	• **RelativeToPaintBuffer**	
Auto Reset	Sets the **Auto Reset** option to: • **Reset** • **DoNotReset**	
Reset	Resets any transformations applied to the image using the **Image Scale**, **Rotation**, **Scale** or **Translation** options. This **Reset** option is dependent on what you set in the **Reset Size** and **Auto Reset** options.	
Rotation	Sets the rotation of the image used by the **Clone Stamp** tool's preview.	
Scale	Sets the scale of the image on the x and y axes. This scales the image relative to the canvas size.	
Translation	Sets the translated xy position of the image used by the **Clone Stamp** tool's preview.	
Scratch pad - shows a squiggle painted with the selected brush. You can also paint directly in the scratch pad (the default squiggle disappears while you enter a stroke).		

Changing Brush Properties “On the Fly”

Mari also includes default keyboard-and-mouse shortcuts to change the relative **Radius**, **Rotation**, **Opacity**, and **Squash** of the brush you're using.

1. To change the **RADIUS** of your brush, press and hold **R**, and click and hold the left or right button while moving the cursor to increase or decrease the outer radius, and up or down to increase or decrease the inner radius.
2. To change the **ROTATION** of your brush, press and hold **W**, and click and hold the left or right button while moving the cursor to rotate clockwise or counterclockwise.
3. To change the **OPACITY** of your brush, press and hold **O**, and click and hold the left or right button while moving the cursor to increase or decrease the opacity.
4. To change the **SQUASH** of your brush, press and hold **Q**, and click and hold the left or right button while moving the cursor.

Video: Watch a quick demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

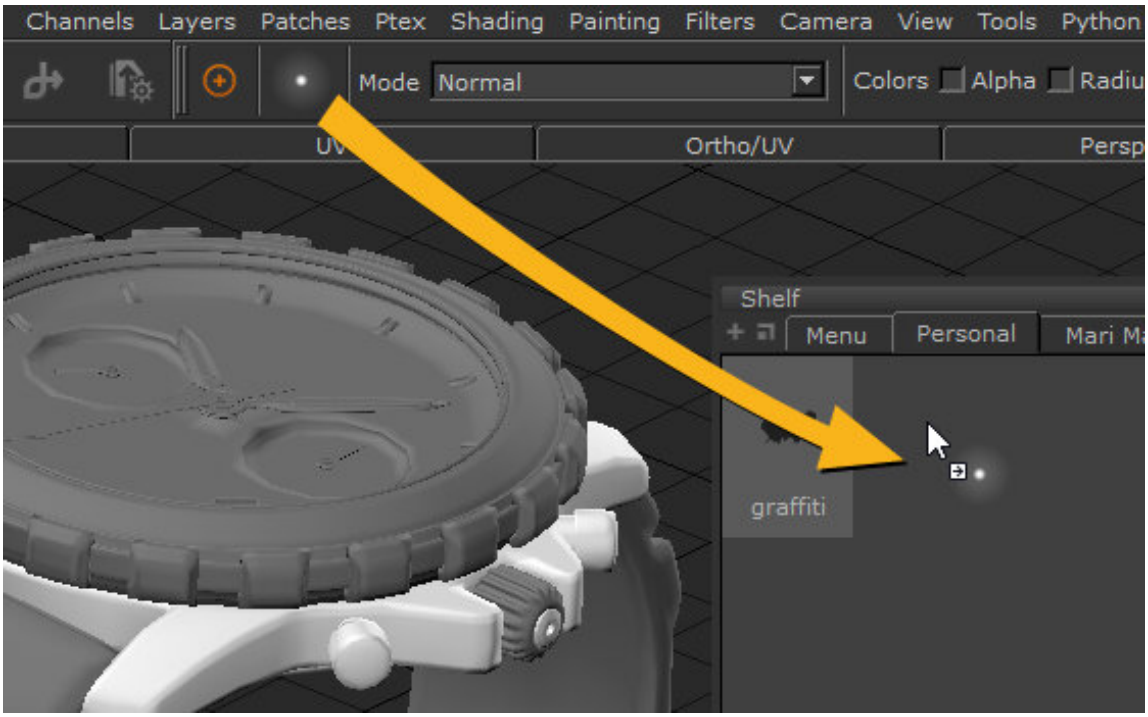
Tip: To save any changes you make on the fly, see the procedure to Customizing a Brush.

Experiment: Play around with the **R**, **O**, **W**, and **Q** keyboard/mouse shortcuts for changing your brush tip on the fly.

Customizing a Brush

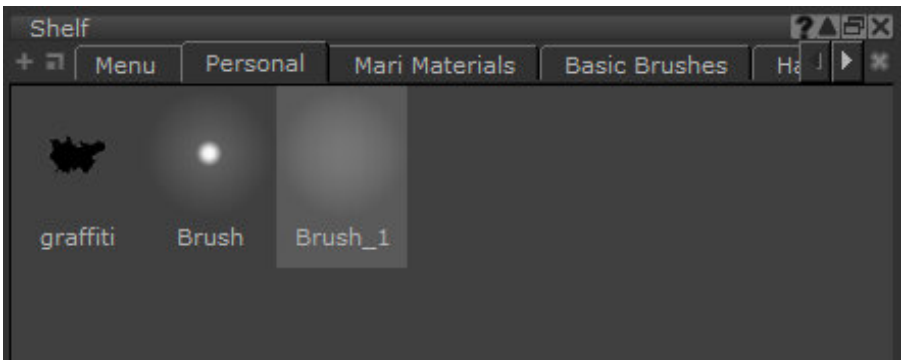
Customizing brushes involves three main steps: selecting a preset, changing its properties, and saving it to a shelf.

1. On the left-hand **Tools** toolbar, select a painting tool such as **Paint**, **Blur**, **Vector Paint**, **Paint Through**, **Gradient**, or **Clone Stamp**.
2. Open the **Shelf** palette, select a shelf for a category of preset brushes, and select a brush.
3. Open the **Tool Properties** palette and change the Brush Properties.
4. In the **Shelf** palette, select the **Menu** shelf, **Personal** shelf, **Project** shelf, or any other custom shelf you may have created.
5. Drag the customized brush from Mari's top toolbar to the **Shelf** palette's shelf on which you want to store the brush.



Note: You must drag modified brushes from the toolbar. If you drag a modified brush from an existing shelf, such as **Basic Brushes**, any changes you made are discarded.

The modified brush displays in the selected shelf. By default it is named **Brush** (and then **Brush_1**, **Brush_2**, and so on.



6. To rename a brush, double-click its name, type a new name, and press **Enter**.
7. To save your new customized brush, save your project. The **Project** shelf only stores brushes per project.

Note: Any brush tip that you select is shown as an outline for the brush cursor design with any tool that allows you to apply paint to the model, for example, **Paint**, **Paint Through**, and **Clone Stamp**, among others. This cursor outline scales according to the radius of the brush tip, but does not reflect other options, such as jitter or opacity.

Tip: To a delete a custom brush, select the brush then right-click and select **Delete Item** from the dropdown menu. You can also press **Backspace** or **Delete**.

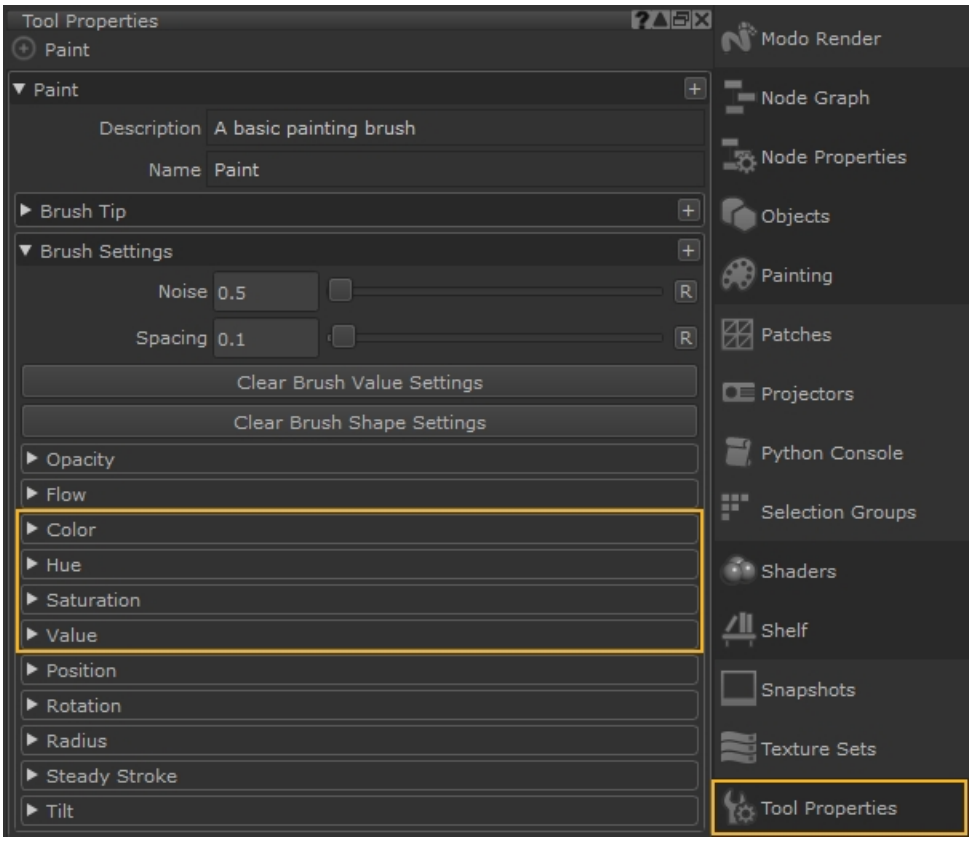
Tip: Try creating three custom brushes you might actually use and test them out on a model. If it doesn't turn out as you expected, try customizing it and re-saving it to your **Personal** shelf.

Changing Brush Color Dynamically

In Mari the **Color**, **Hue**, **Saturation**, and **Value** properties of a brush can be "jittered" on a per stroke, per paint splat or pen pressure basis. Jittering picks random **Color**, **Hue**, **Saturation**, and **Value** levels from within a given range to alter the brush dynamically as you paint.

Note: This feature is available in Mari 4.8, and later.

Brush color is controlled from the **Tool Properties** palette under **Brush Settings**.



Jittering Color by Stroke, Splat, and Pen Pressure

The **Color**, **Hue**, **Saturation**, and **Value** slider controls can be configured to jitter by:

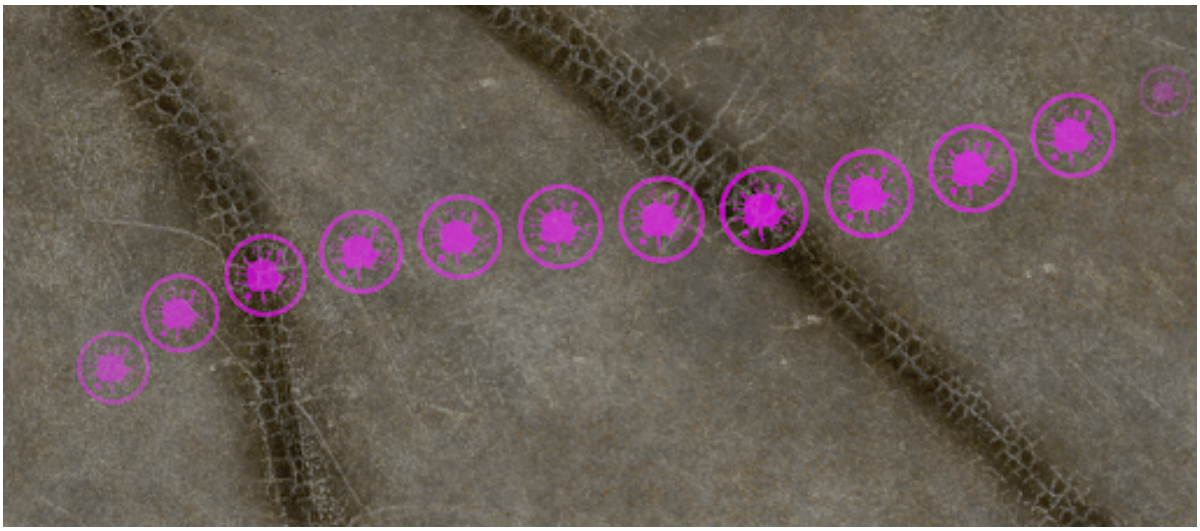
- **Jitter by Stroke** - the **Color**, **Hue**, **Saturation**, and **Value** property of a brush varies on a stroke by stroke basis.
- **Jitter by Tip Splat** - the **Color**, **Hue**, **Saturation**, and **Value** property of a brush varies per tip splat of the brush stroke.
- **Jitter by Pen Pressure** - when enabled, the stroke or tip splat jitters in accordance with how much pressure is applied to the pen of your graphics tablet.

Choosing the Jitter Color

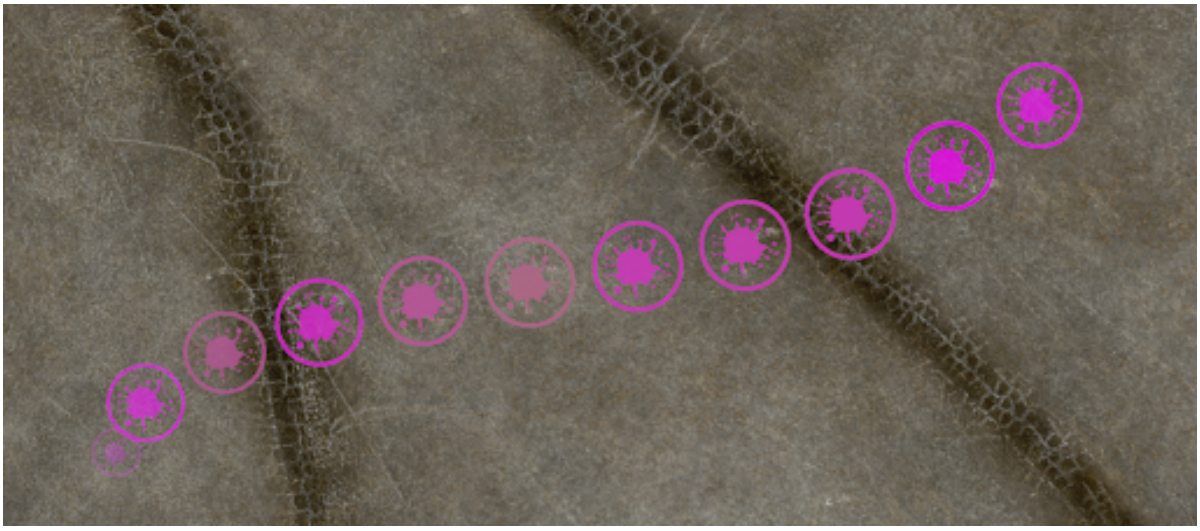
The color range controller allows you to precisely specify how the jittering between the foreground and background color occurs.

The range slider for the color of a brush tip operates from 0.0-1.0, where 1 represents a given background color, and 0 represents a given foreground color.

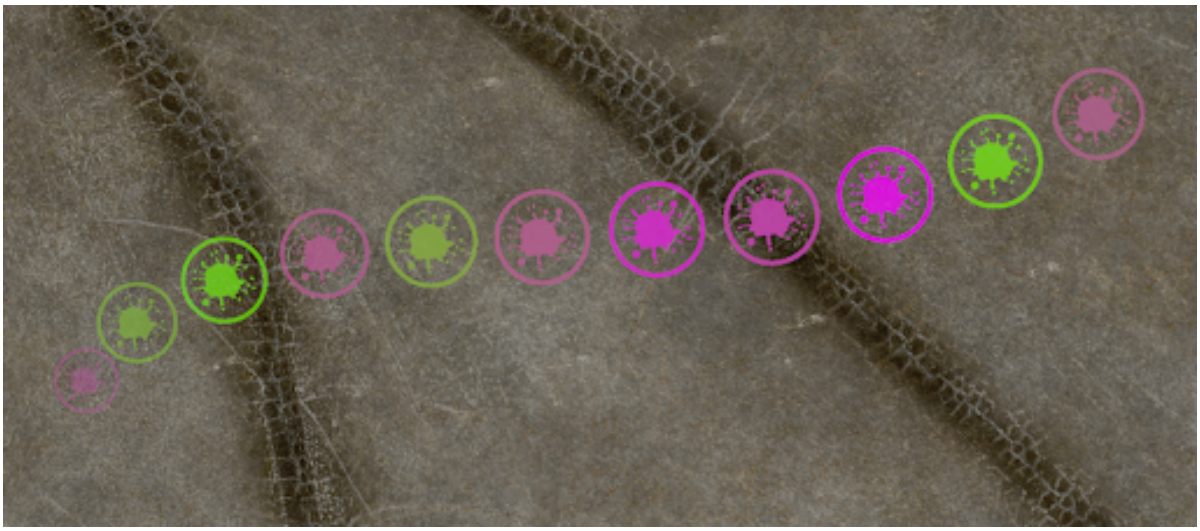
Note: The **Use Legacy Coloring** checkbox allows you to enable the alpha channel controlled color jittering feature available in versions of Mari 4 or older.



A brush stroke with no color jitter per tip splat.



A brush stroke with the color jitter per tip splat range set from 0.5 to 1.0.



A brush stroke with the color jitter per tip splat range set from 0.0 to 1.0.

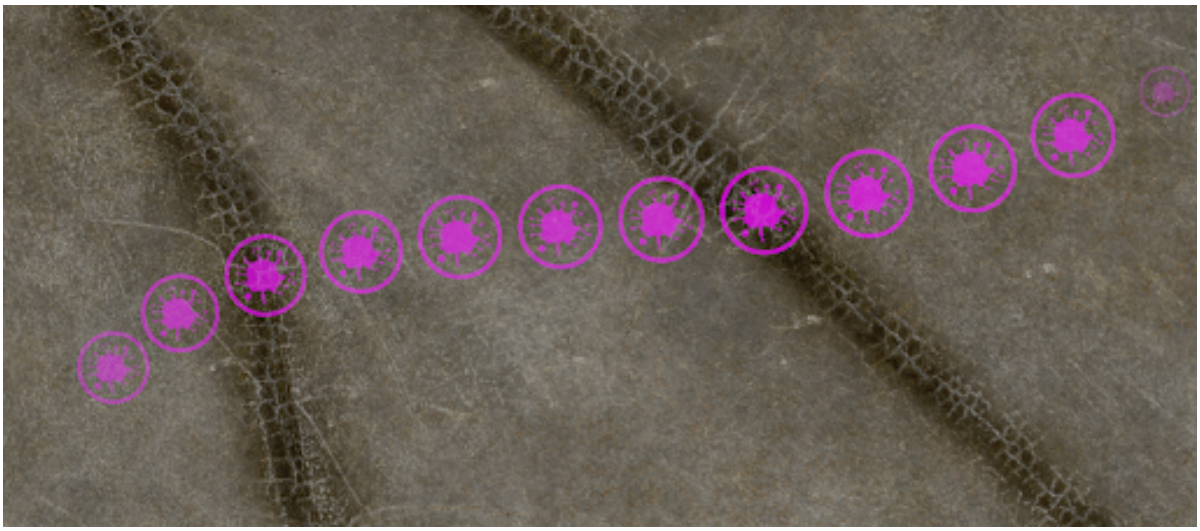
Choosing the Jitter Hue

The hue range controller allows you to precisely specify how the hue of your current selected color will jitter.

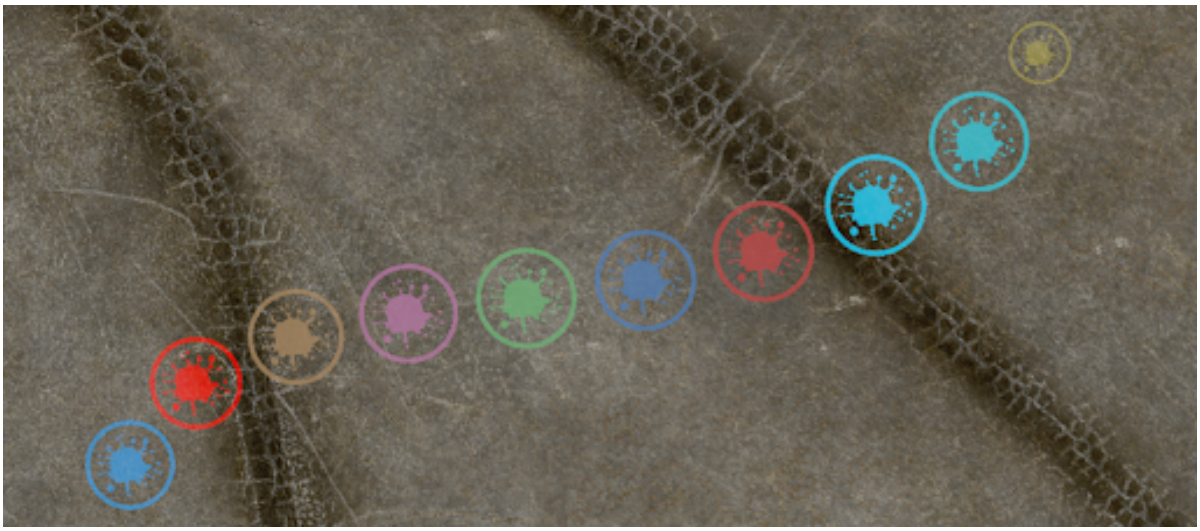
Hue shift jitters on a positive or negative basis, with the range slider operating from -1 to +1, where 0 specifies no hue shift jittering.

For example, a negative value on the range slider such as -1 equates to -360 degrees of hue shift. A positive value such as +1 represents +360 degrees of hue shift.

The larger the range between the two values, the more hue shifting occurs in the jitter.



A brush stroke with no hue jitter per tip splat.



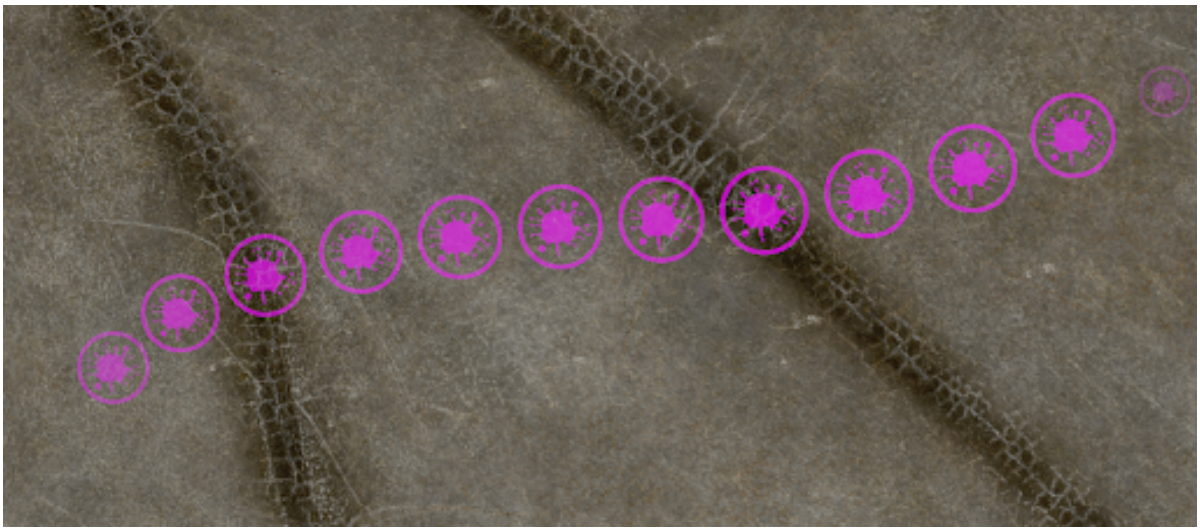
A brush stroke with the hue jitter per tip splat range set from -1.0 to 1.0.

A brush stroke with the hue jitter per tip splat range set from -1.0 to 0.0.

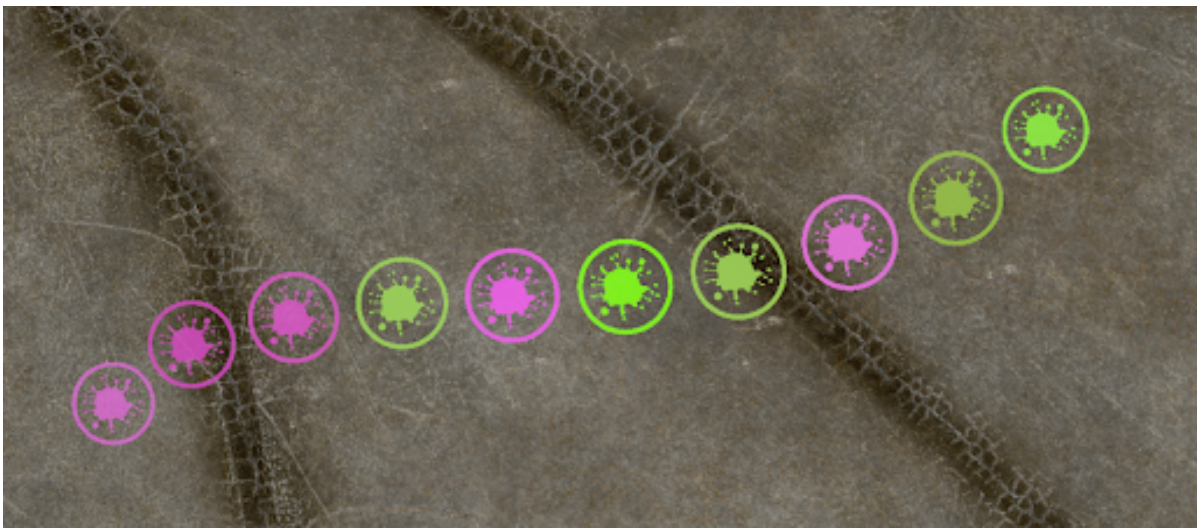
Choosing the Jitter Saturation

The saturation jitter slider specifies the range of saturation jittering that occurs in a brush stroke.

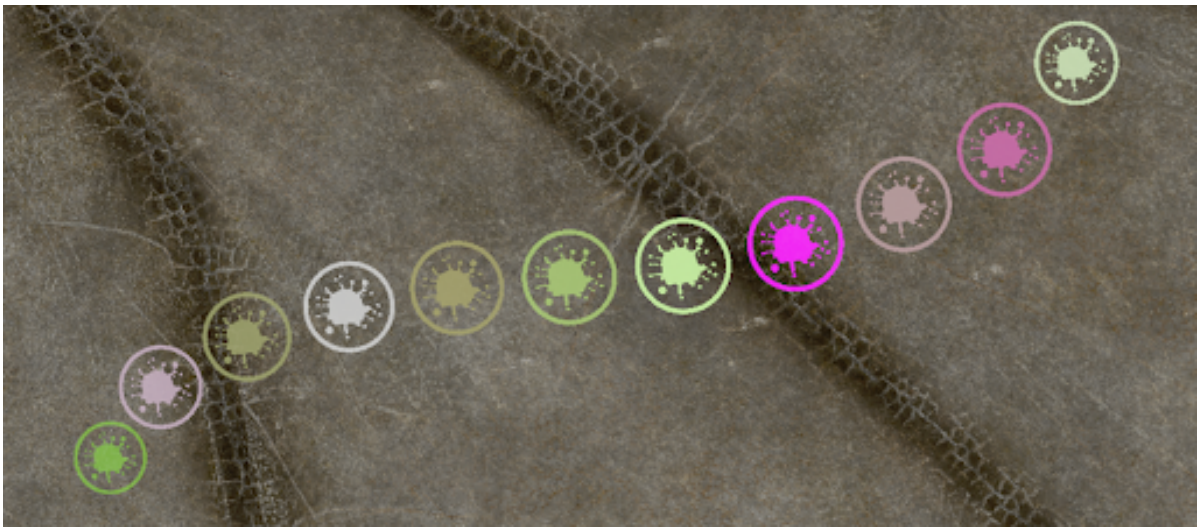
The range slider for the saturation of a brush tip operates from 0.0-1.0, where 1 represents a given foreground color, and 0 represents the least saturated version of the given foreground color.



A brush stroke with no saturation jitter per tip splat.



A brush stroke with the saturation jitter per tip splat range set from 0.5 to 1.0.

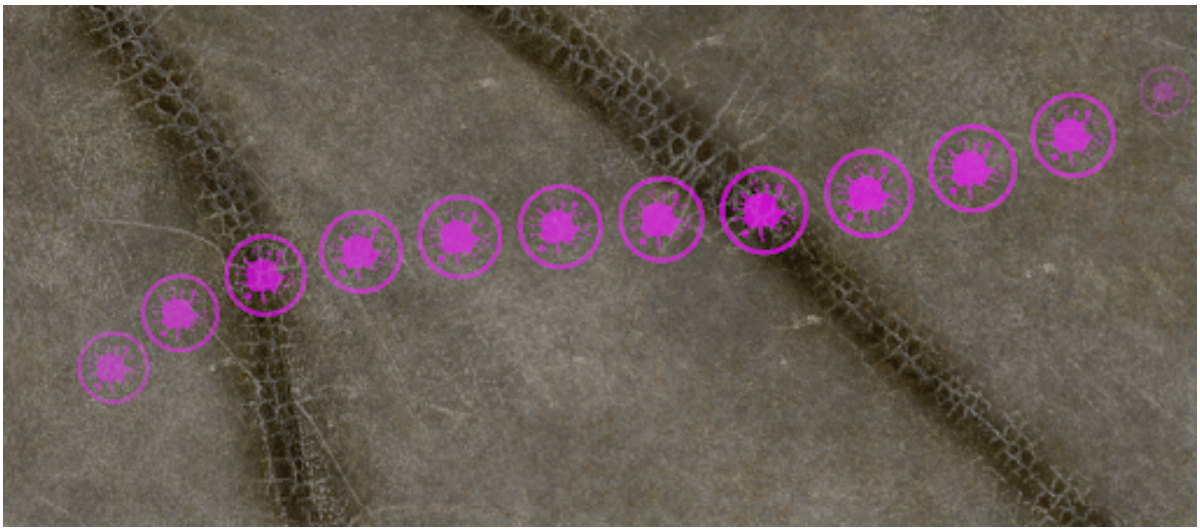


A brush stroke with the saturation jitter per tip splat range set from 0.0 to 1.0.

Choosing the Jitter Value

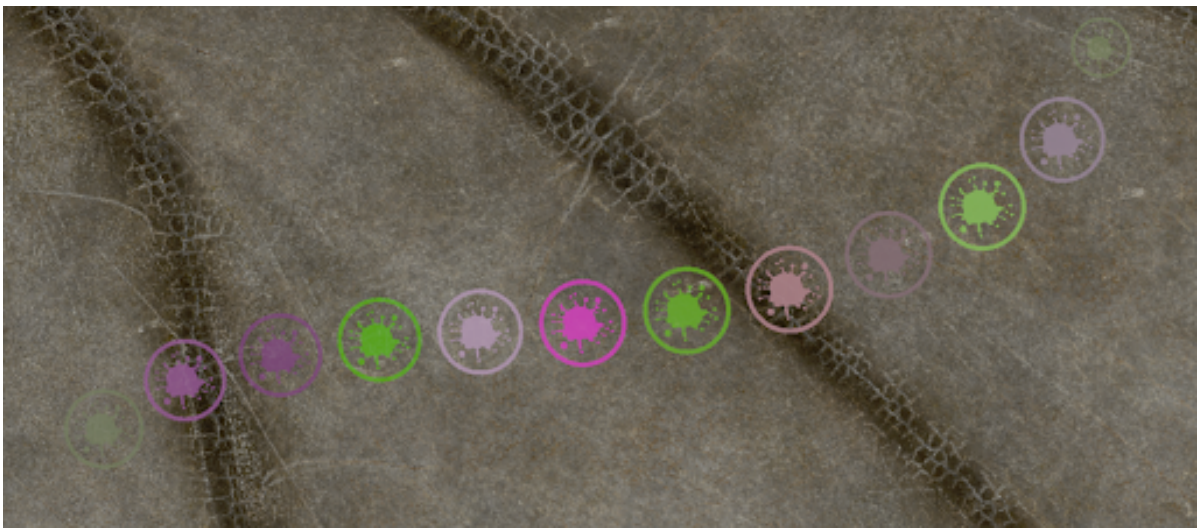
The value of the chosen foreground color can also be jittered.

The range slider for the value of a brush tip operates from values of 0.0 - 1.0, where 1 represents a given foreground color, and 0 represents the given foreground color at its lowest value.



A brush stroke with no value jitter per tip splat.

A brush stroke with the value jitter per tip splat range set from 0.5 to 1.0.



A brush stroke with the value jitter per tip splat range set from 0.0 to 1.0.

Exporting and Importing Brushes

You can also export and import your brushes. This is useful, for example if you want to share brushes with other users.

1. To export a custom brush, in the **Shelf** palette, in the **Personal** shelf, **Project** shelf, or any other custom shelf you may have created, right-click a brush and select **Save Item**.

   The **Save Item** dialog box prompts you to save the brush, as an **.msi** file.

Tip: The first time you save or open items in Mari, it defaults to your home directory. Thereafter, Mari remembers the last folder you navigated to in that dialog box.

2. Type a filename, and click **Save**.

   Mari exports the brush.
3. To import a brush, right-click in the **Personal**, **Project** or custom shelf and select **Load Item**.

   The **Import Item** dialog box prompts you to select an **.msi** file.
4. Navigate to and select the **.msi** file for the brush you want to import, and click **Open**.

   Mari imports the brush.

Tip: You can also import and export entire shelves (see Storing Resources Using Shelves).

Experiment: Try creating a custom brush, exporting it, then deleting it from the shelf, and finally (re)importing it.

Importing a Custom Brush

Custom-made Photoshop brushes can be imported into Mari and stored in a custom shelf. These brushes are "stamp" brushes that do not take into consideration any variable size or spacing. The shape (or shapes) of the brush are imported into a custom shelf where you can set the preferences for the brush manually.

To import custom brushes:

1. Navigate to *Menubar* | **Tools** > **Import Brushes**.

   This opens an **Import Brushes** dialog.
2. Select the Photoshop brush (**.abr** format) from the file location it was saved to, and click **Open**.

   A new shelf is created and populated with the brush or brush parts. If there are multiple "stamp" images, you may need to set the preferences for the brush.

Note: See Shelf Palette for more information about managing your brushes and items in Mari.

Note: Brushes imported from Photoshop must be custom-made. If you try to import Photoshop's default brushes, or any invalid brush, an error message displays and the brush fails to be imported.

Video: Watch this video for more about how to import our Photoshop brushes into Mari. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Storing Resources Using Shelves

You can store just about anything you can grab from within Mari in **Shelves**. Most artists like to store their favorite brushes, colors, and images (for clone stamping or painting through). Power users can also add any preset Mari function to a shelf.

By default, the Mari **Shelf** palette comes with a number of shelves, including a few preset shelves:

- **Menu** — items that you can select from the **F9** pie selection control menu.
- **Personal** — selected items you use regularly.
- **Project** — items just for the current project.
- **Basic Brushes** — a set of predefined basic brushes.
- **Brad's New Brushes** — a set of predefined brushes.
- **Brad's Movie Brushes** — another set of predefined brushes by Brad DeCaussin.
- **Grunge** — a set of high-quality grunge textures created by industry professionals.
- **Hard Surface Brushes** — a set of predefined hard surface brushes.
- **Materials** — a set of predefined materials based on the **Principled BRDF** shader model. See the *Principled BRDF* section in the *Mari Reference Guide*.
- **Organic Brushes** — a set of predefined organic brushes.
- **Python Examples** — predefined python scripts that perform certain useful functions.
- **Smart Masks** — ready-to-use masks and grunge maps, and a set of utilities to edit them.
  Note: this shelf includes items which used to be called Custom Procedurals.
- **Color Presets** — stores colors that you have saved.

You can also create any number of customized shelves.

Your shelves display in the **Shelf** palette, see Configuring Brushes. You can also make any shelf into a separate, custom palette. Mari also lets you export and import entire shelves and items within shelves, for example to share with other users.

Tip: For an overview of all the ready-to-use content in the shelf, see Ready-to-use Shelf Content.

Note: To load Mari without the **Mari Materials**, **Basic Brushes**, **Hard Surface Brushes**, **Organic Brushes**, and **Brad's New Brushes** preset shelves, disable the **Edit** > **Preferences** > **Misc.** > **Shelf** > **Shelves Include Presets** preference. See the *Mari Preference Dialog* section in the *Mari Reference Guide*.

Video: To display the **Shelf** quick palette when painting, press and hold **K**. See the *Shelf Palette* section in the *Mari Reference Guide*. Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Tip: For details on using the pie selection control menu, see Using the Pie Selection Control.

Searching your Shelves

Use the Shelf search box to find specific items across your shelves. You can also use the following operations to filter through results:

- Enter any required search terms and separate them using spaces.
- Terms separated by **|** (vertical line) will return any results with either term in the name.
- Prefix terms with **-** (dash) to exclude them from your search.
- For exact matches of terms, enclose terms in quotes (**“”**).

For example, searching for **bricks | plaster -cracked** matches any item with **bricks** OR **plaster** in the name, but excludes any with **cracked** in the name.

Searching **“bricks plaster”** will only return results with the exact term **bricks plaster** in them. If you search **bricks plaster** without the quotes, it will return terms with both **bricks** and **plaster** anywhere in the result name.

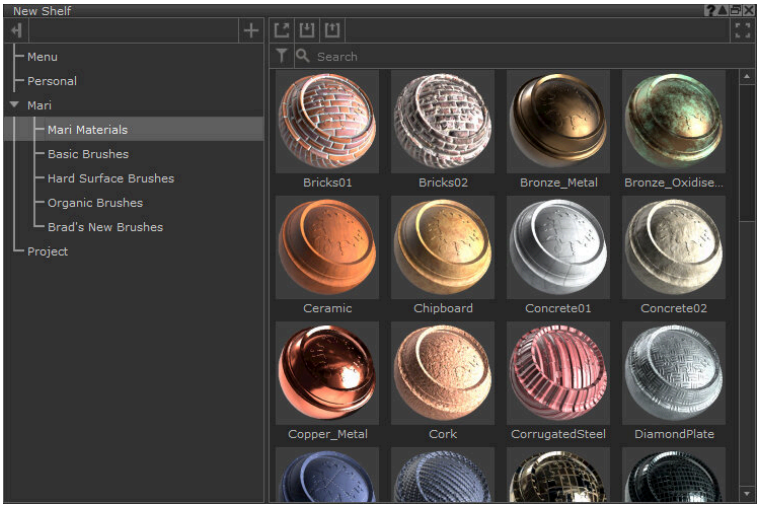
Use **#** to search for Shelf Items that have been assigned a specific Tag.

Opening Shelves

1. Start Mari and open your project.
2. If the **Shelf** palette is not already open:

- from the **View** menu, select **Palettes** > **Shelf**, or
- right-click in the toolbar area and select **Shelf** from the dropdown menu.

The **Shelf** palette displays.



The Mari **Shelf** palette.

Video: To display the **Shelf** quick palette when painting, press and hold **K**. See the *Shelf Palette* section in the *Mari Reference Guide*. Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Tip: The default layout in Mari has the **Shelf** palette already open.

As described above, the **Shelf** starts with seven default shelves: **Menu**, **Personal**, **Basic Brushes**, **Hard Surface Brushes**, **Organic Brushes**, **Brad's New Brushes**, and **Project**. The **Menu** and **Personal** shelves are empty when you first start Mari. The **Project** shelf only displays when you have a project open (and is empty when you start a new project).

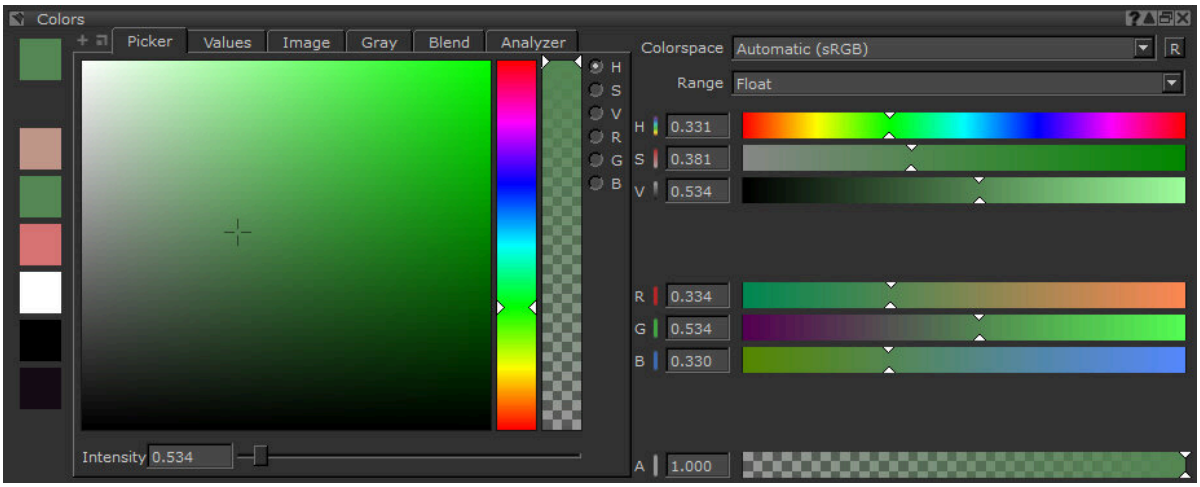
Adding a Color to a Shelf

To add a color to a shelf:

1. If the **Colors** palette is not already open:
   - from the **View** menu, select **Palettes** > **Colors**, or

• right-click in the toolbar area and select **Colors** from the dropdown menu.

The **Colors** palette displays.



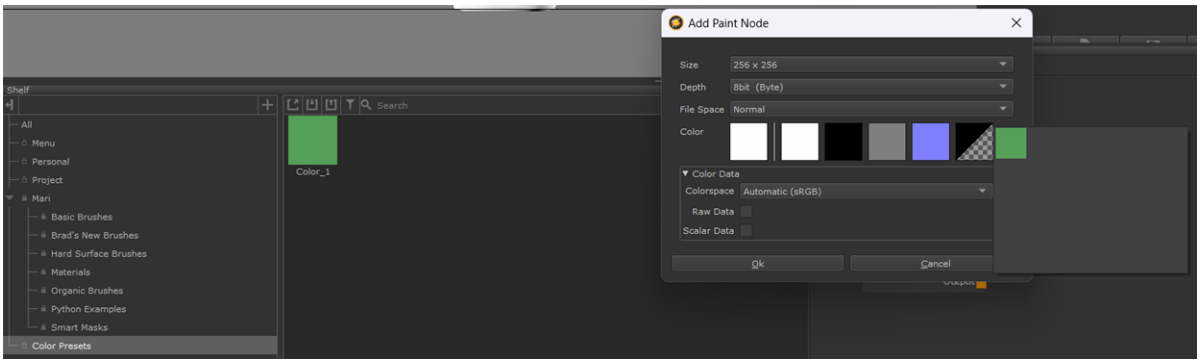
Tip: The default layout in Mari has the **Colors** palette already open.

2. Open the shelf in the **Shelf** palette you want to copy the color to. You can use the **Color Presets** shelf for this, as this will then automatically make your saved colors available in the Add Paint Node / Add Bake Point / Add Channel dialogs as well. Or, you can add the color to other shelves such as **Personal**, **Project**, or Creating a Custom Shelf.
3. When the color you want to copy is in the **Foreground** or **Background** color swatch, drag and drop it from the **Colors** palette to the shelf where you want to store it.

Tip: The swatch on the shelf only holds the color values, not the alpha. Swatches on shelves always have an alpha of 1.0. You can also drag and drop color swatches from anywhere else they appear in Mari (such as the **Properties** palette or **Select Color** dialog box).

You can display a large array of color swatches in a shelf by minimizing the icon size: right-click anywhere in the shelf, and select **Toggle Icon Size** from the dropdown menu. Icons switch from large to small (or the other way around).

4. If you use the **Color Presets** shelf, this color can then be accessed directly from both the shelf and from the Add Paint Node / Add Bake Point / Add Channel dialogs when clicking .

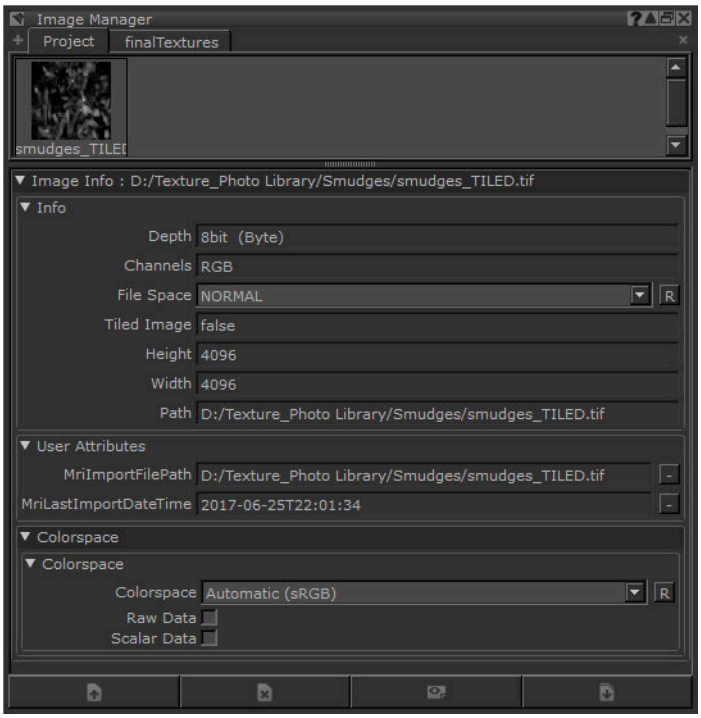


Adding an Image to a Shelf

To add an image to a shelf (for example, for painting through or clone stamping):

1. If the **Image Manager** palette is not already open:
   - from the **View** menu, select **Palettes** > **Image Manager**, or
   - right-click in the toolbar area and select **Image Manager** from the dropdown menu.

   The **Image Manager** palette displays.



Tip: The default layout in Mari has the **Image Manager** palette already open. (Click to give it focus.) See Managing Images for instructions on loading images into the **Image Manager**.

2. Click the shelf in the **Shelf** palette you want to copy the color to (**Personal**, **Project**, or Creating a Custom Shelf).
3. Drag and drop an image from the **Image Manager** to the shelf where you want to store it.

Tip: You cannot copy temporary, cropped images from the **Image Manager** to a shelf - unless you save them first.

Tip: You can rename an item in a shelf, by double-clicking its name and typing over. You can also drag any item from any shelf to the **Personal**, **Project**, or custom shelves.

Adding Materials to a Shelf

Materials are created from multiple texture files, procedurals, or plain colors that can be used to accurately represent how light interacts with real-world materials. This method of shading is called Physically Based Rendering (PBR).

You can apply these materials to your assets for a more realistic result as well as establishing a base line look for your asset before adding more bespoke paint and crafting details.

Mari provides a set of predefined materials (presets) in the **Mari Materials** tab of the **Shelf** palette but you can also add your own materials to the **Menu**, **Personal**, and **Project** shelves or any custom shelves.

Note: You can create materials manually using the Node Graph (see Working with Materials in the Node Graph) or you can use the Material Ingest Tool (see the *Material Ingest Tool Dialog* in the *Mari Reference Guide*).

To add your own materials to a shelf, drag one or more **.mma** files from your file browser into a shelf.

Alternatively, right-click in the shelf, select **Load Items** then browse to the **.mma** file(s), and click **Open**.

This creates material thumbnails in the shelf.

Note: Hover over materials in the shelf to see a larger thumbnail display with information about the material.

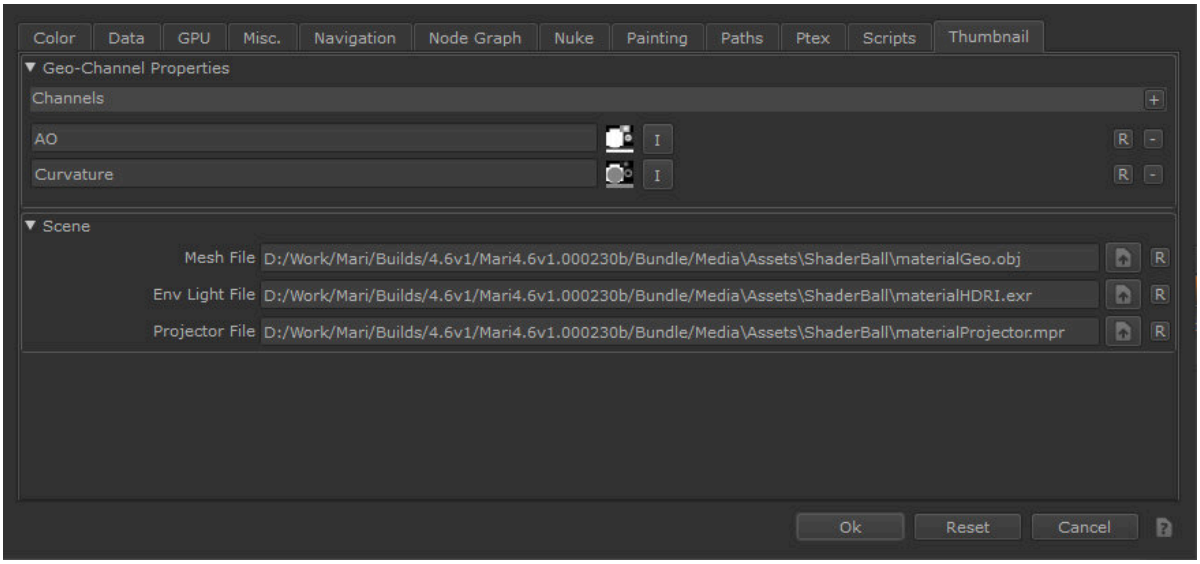
To delete a material from a shelf, select one or more materials and press **Delete** or right-click and select **Delete Item**.

Note: You can also drag predefined materials from the **Mari Materials** tab to your own shelves.

Customizing Material Thumbnails in the Shelf

When you export a material as a **.mma** file, you have the ability to define the object, environment, and projector that appear in the thumbnail for the material. The options to do this are found in the **Preferences** dialog, under the **Thumbnail** tab.

Note: This only affects materials exported after you've changed the **Thumbnail** preferences.



The **Thumbnail** tab in the **Preferences**.

To Change the Mesh, Environment, and Projector Which Appear in the Exported Material Thumbnail:

1. Select a mesh file using in the **Mesh File** property.

2. Select an environment light file using in the **Env Light File** property.
3. Select a projector file using in the **Projector File** property.
   For more information on Projectors, see Creating the Projector.

You can also add Geo-Channels to the thumbnail mesh object, which can be useful in instances where your materials make use of Geo-Channels. For more information on Geo-Channels see Adding Geo-Channels to Objects and Geo-Channel Nodes.

Adding or Changing Geo-Channels for the Object in your Material Thumbnails

1. Click on **Channels** to add a new Geo-Channel.
2. Click in the Geo-Channel name field to give the Geo-Channel a new name.
3. Click on the Geo-Channel to add an imageset to the Geo-Channel.

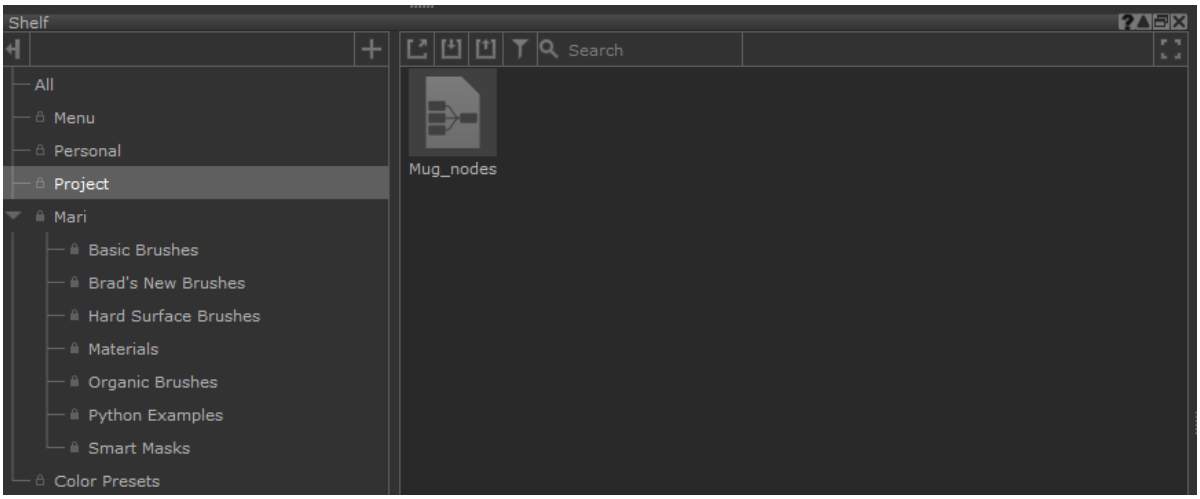
When **.mma** files are exported and placed in your **Shelf**, they appear with the mesh and associated geo-channels, environment, and projector you selected.

Adding Nodes to a Shelf

You can export part of the project's Node Graph as **Node Packages**. Similarly to Materials and Smart Masks, Node Packages bundle up the node network, along with any source images, into a single file. This provides easier portability of premade Node Graphs between artists by removing the need to re-import images that cannot be found at their original paths.

To add Nodes to a Shelf

1. In the node graph, select the nodes you wish to save. Click and drag over them to select multiple nodes.
2. Right click on the nodes and select **File** > **Package Nodes**.
3. A new window will appear, allowing you to save a node archive file (**.mna**).
4. Once the file is saved, open the Shelf Palette and find the section you would like the nodes to appear in.
5. Select the Import Items button:
6. This opens a file browser window - you can then choose the .mna file you saved in step 3.
7. The nodes now appear in your shelf.



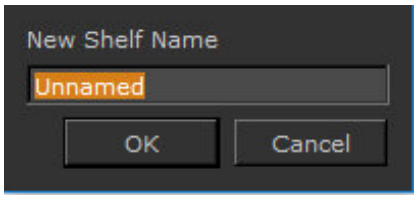
8. You can then click and drag this file into your node graph to import them into your node graph.

Creating a Custom Shelf

To create custom shelves (for example particular sets of color swatches, images, or brushes):

1. In the **Shelf** palette, click .

   The **Create New Shelf** dialog box displays.



2. Type a name for the shelf and click **OK**.

   The new shelf displays at right in the **Shelf** palette.

Adding any Function to a Shelf

To add any Mari function to the **Personal**, **Project** or a custom shelf:

1. From the **Edit** menu, select **Shortcuts**.

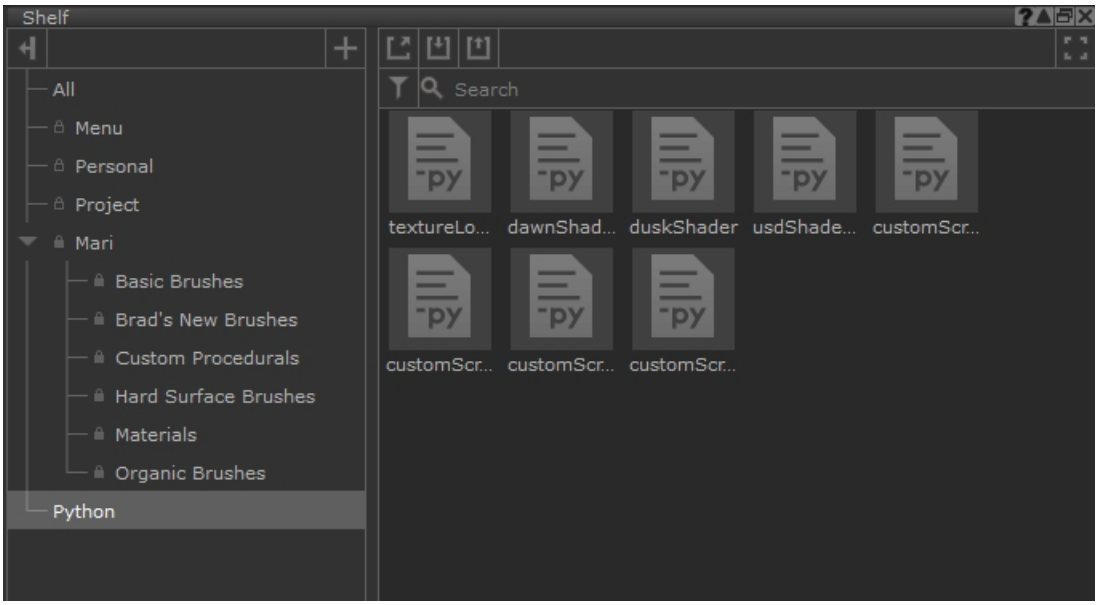
   The **Manage Keyboard Shortcuts** dialog box displays.

Tip: Resize columns and the dialog box to better view the names and descriptions of Mari functions.

2. Click to expand the hierarchical list of Mari functions (actions, tools, palettes, and so on.)
3. Click to open the shelf you want to drag a function to.
4. Drag and drop the function from the **Manage Keyboard Shortcuts** dialog box to the shelf.

Run Custom Python Scripts Directly from Shelves

Sharing Python scripts created through Mari's powerful Python API encourages collaboration between artists and storing these scripts in shelves means you can execute a script by double-clicking it in the shelf. For example, a colleague may have created a Python script that automatically imports certain textures for a show you're working on. By importing this script to a Shelf Item, you no longer need to install this script manually. Instead, simply run the script each time you start on a new asset for the show - keeping the look of the show consistent and reducing project setup time.



You can edit scripts directly in the Shelf using the **Edit Script** widget or drag and drop the script from the shelf to the **Python Console**. You can also change the default shelf script thumbnail to describe its function at a glance.

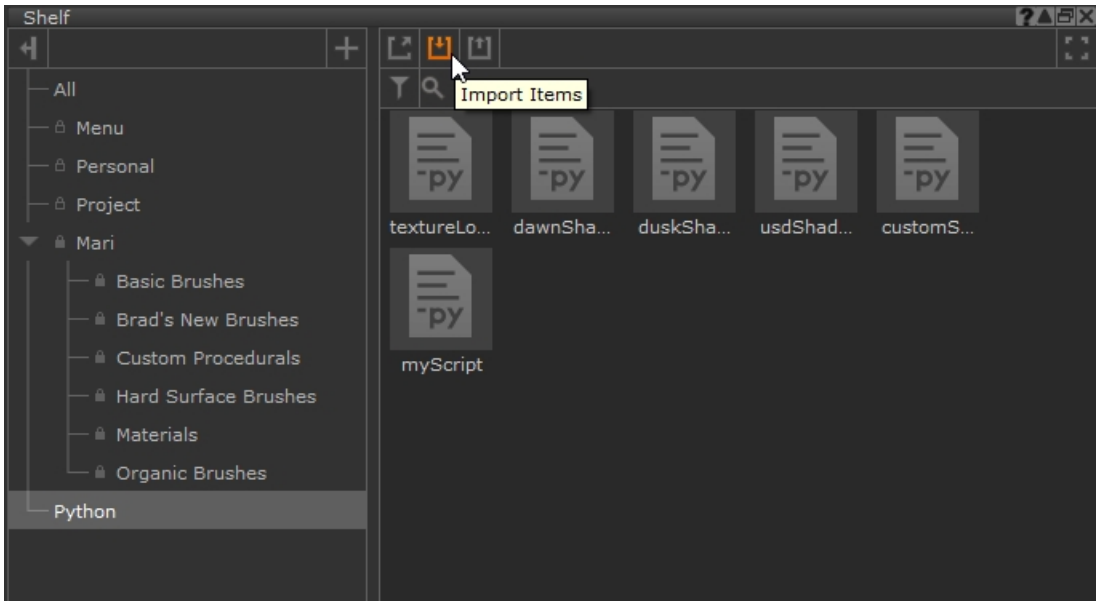
Adding and Removing Python Scripts in Shelves

Python scripts are added to shelves by importing **.py** files from disk or by dragging code from the Python Console in Mari.

> **Note:** You can't edit the shelves that ship with Mari, so you'll have to create a custom shelf to hold your Python Scripts. See Creating a Custom Shelf for more information.

To create shelf items by importing Python scripts from disk:

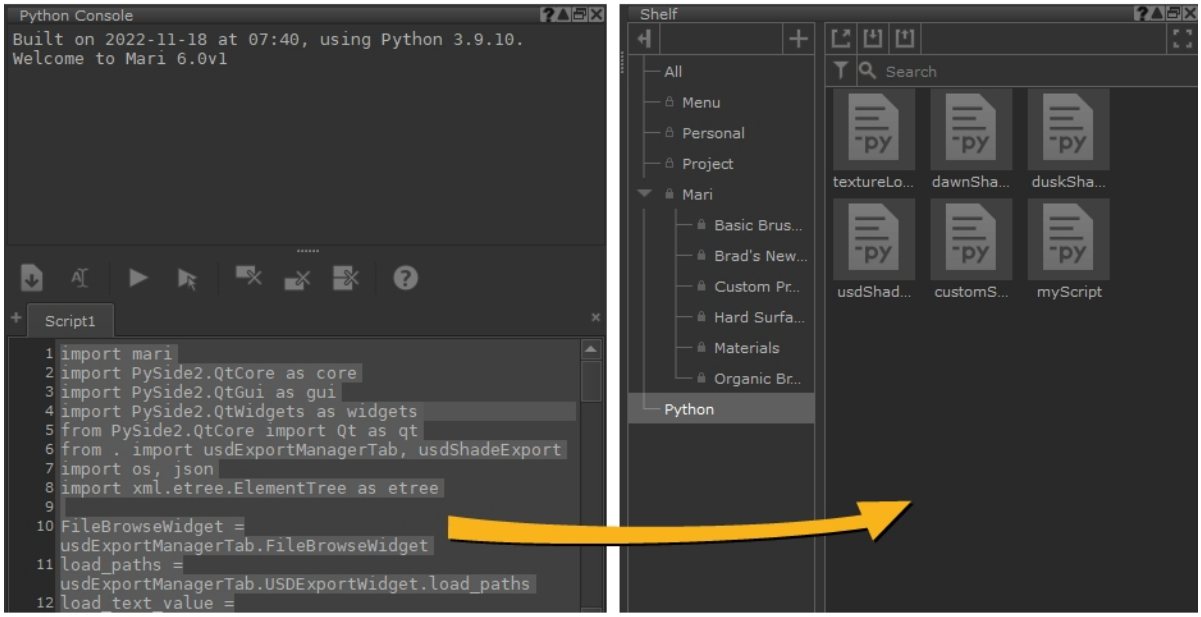
1. Open the **Shelf** palette by clicking **Shelf** in the right-hand toolbar.
2. Select the shelf you want to use for script storage.
3. Click the Import Items button and then browse to the location of your script.



4. Select the script and click **Open** to add the script to your selected shelf.

To create shelf items from the Python Console:

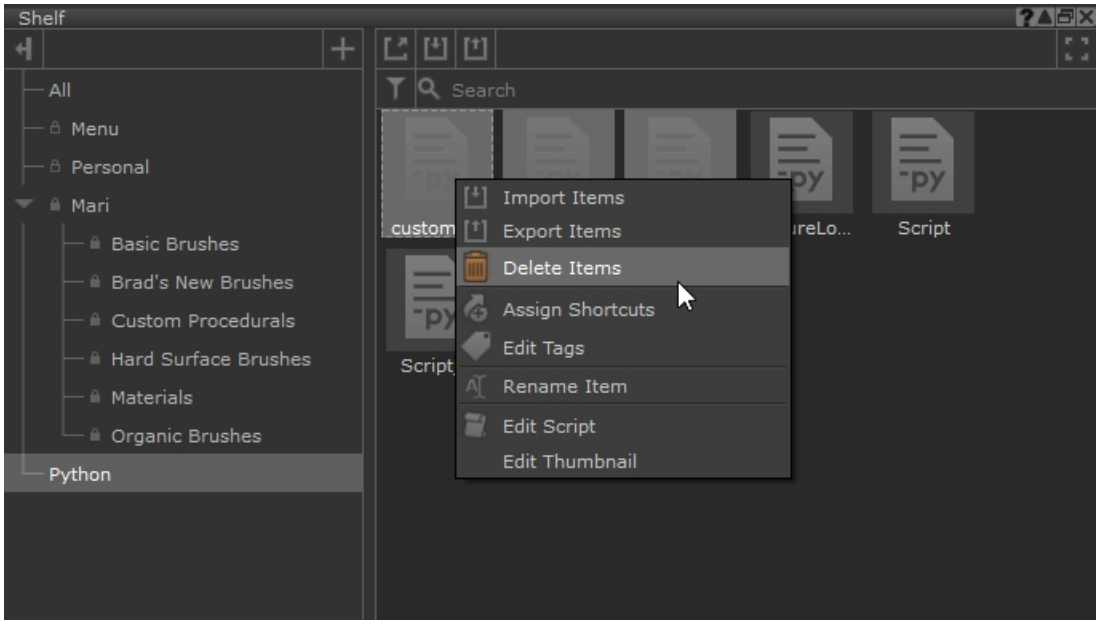
1. Enter the Python code you want to save as a shelf item manually or import a script from disk into the console.
2. Select the entire script in the console editor and drag and drop the text into the **Shelf** palette in the shelf you want to use.



The new Python shelf item takes its name from the **Python Console** default, **Script1**, or from the Python file you imported.

To delete a shelf item:

1. Select the shelf you're using for script storage.
2. Select the script or scripts you want to delete.
3. Press **Backspace** on your keyboard or right-click a selected script and click **Delete Items**.



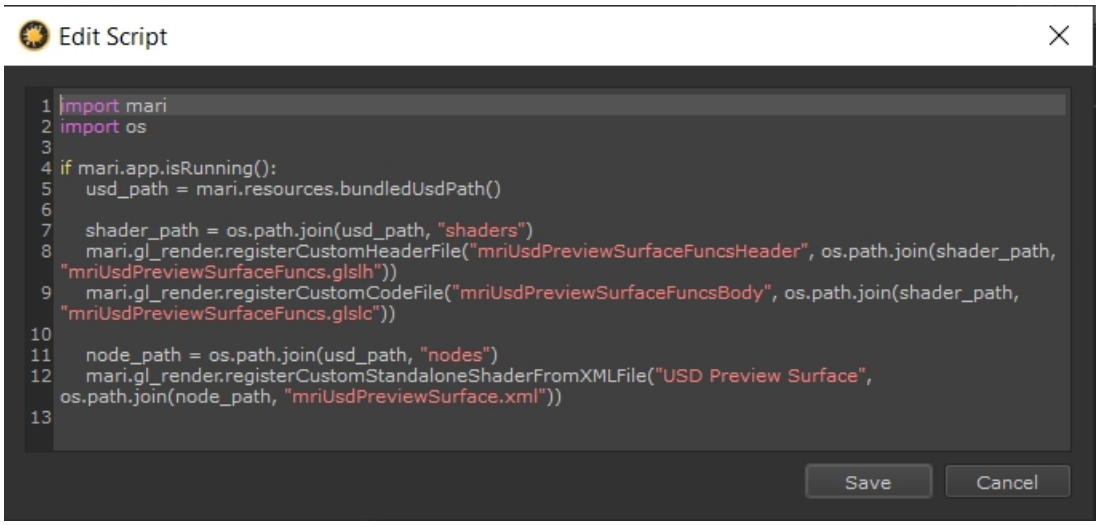
Editing Python Scripts in Shelves

Mari's **Shelf** palette includes an editing widget that saves you time if you want to edit a Python script in a shelf. You can drag and drop a script into the Python Console, but for small changes, using the editor means you don't have to open another palette to make an edit.

To edit a script in a shelf:

1. Select the shelf you're using for script storage.
2. Select the script you want to edit.
3. Right-click the selected script and click **Edit Script**.

   The **Edit Script** widget is displayed.



4. Edit the script as required and then click **Save** to update the script in the shelf.

> **Tip:** You can change the default shelf script thumbnail to describe a scripts function at a glance. Right-click a script in the shelf and select **Edit Thumbnail** to browse to an image file on disk.

Executing a Python Script from a Shelf

Once you've imported a shelf script or created your own, you can run a script by double-clicking it in the shelf or by dragging the script to the **Python Console** and running it from there.

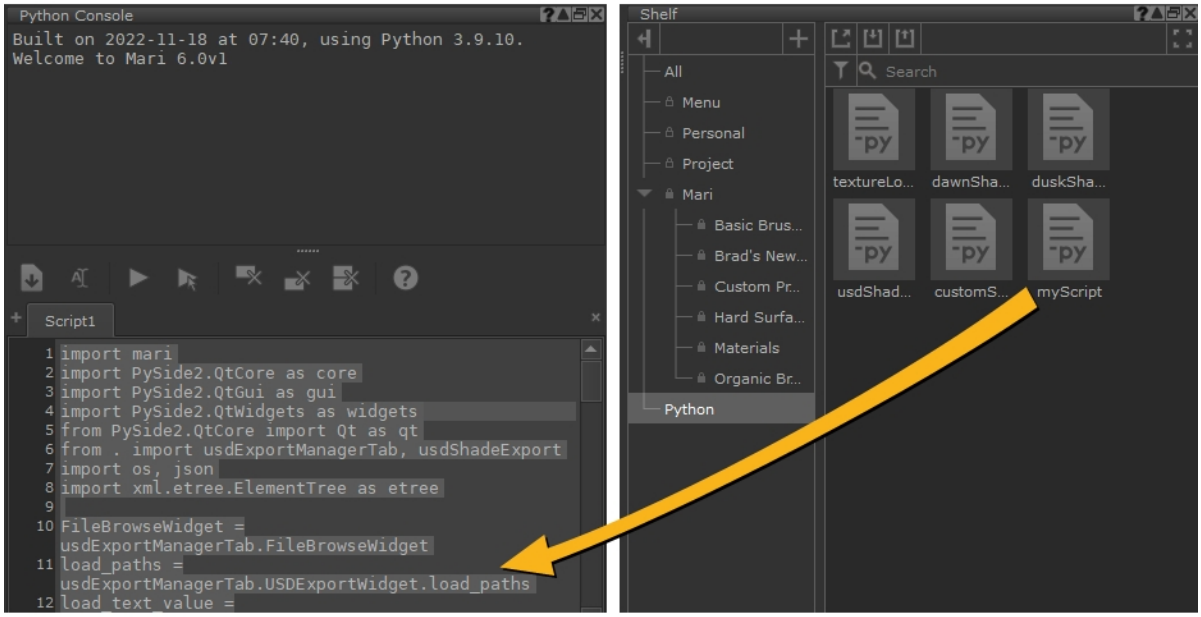
To run a script from the shelf:

1. Select the shelf you're using for script storage.
2. Double-click the script you want to run.

   Mari executes the script automatically, you don't need to do anything else.

To run a script from the Python Console:

1. Select the shelf you're using for script storage.
2. Drag the script into the **Python Console**.



3. Click the **Evaluate** button or press **Ctrl**+**Enter** to execute the script.

Grouping Shelves in the Shelf

In the Shelf palette, you can group shelves together under a parent shelf so you can organize all your stored presets for better viewing and access.

> **Note:** To learn more about the **Shelf**, see Shelf Palette and Storing Resources Using Shelves.

To group shelves under a parent shelf:

1. Click **Shelf** on the palette toolbar to open the Shelf palette.
2. From the shelf panel, right-click the shelf you would like to group under a parent shelf and click **Group**.
   A **Group Shelf** dialog is opened.
3. Select an existing shelf from the dropdown OR,
   enter a new group name into the text field.

> **Note:** If a shelf already exists that you would like to group the selected shelf under, type in that shelf name.

4. Click **OK**.

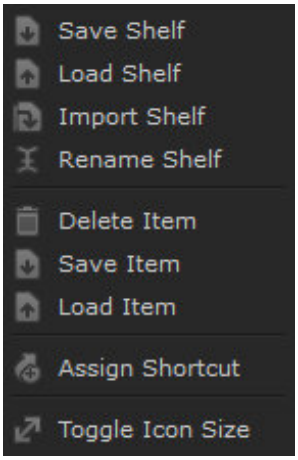
To ungroup shelves:

1. Right-click the shelf in the shelf panel that you would like to ungroup.
2. Click **Ungroup Shelves**.

   The shelf is removed from the group hierarchy.

Modifying Custom Shelves and Their Contents

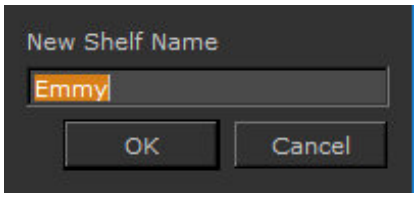
The **Shelf** palette includes a right-click dropdown menu with options to rename, delete items from, and assign shortcut keys to items in the **Personal**, **Project**, or custom shelves. (It also includes features to Importing and Exporting Shelves and Items.)



Tip: Some of the customization features are not available for some of the preset brushes shelves (their name and contents cannot be changed).

1. To rename a shelf, right-click anywhere in the shelf and select **Rename Shelf**.

   The **Rename Shelf** dialog box displays.



2. Type the new name, and click **OK**.

Mari renames the shelf.

3. To delete an item from a shelf, right-click it and select **Delete Item**.

   Mari removes that item from the shelf.

4. To assign a shortcut key to an item in a shelf, right-click it and select **Assign Shortcut**.

   The **Assign Keyboard Shortcut** dialog box displays.

> **Tip:** Resize columns and the dialog box to better view shelf names and the **Shortcut** column.

5. Double-click in the **Shortcut** column for the selected item, type a shortcut key, and click **OK**.

   You can now use that shortcut to select that item.

6. To delete a shelf, open it, and click at right. Mari asks you to confirm deleting the shelf and all its contents.

7. Click **OK**.

   Mari deletes the shelf.

> **Tip:** As a learning exercise, try this: create a custom shelf, rename it, add an item to it, rename the item, delete the item, then delete the shelf.

Importing and Exporting Shelves and Items

You can import and export shelves, and items in shelves, for example to share with other Mari users.

> **Tip:** You can “Import” or “Load” all items from an exported shelf into a selected shelf. “Import” means to **add** the items you're importing to those already there, and “Load” means to **replace** all items there with the ones you're loading.

1. To export a shelf, right-click anywhere in the shelf and select **Save Shelf**.

   The **Save Shelf** dialog box displays.

2. Navigate to where you want to save the shelf, give it a name, and click **Save**.

   Mari saves the shelf as an **.msh** (Mari shelf) file.
3. To import all items from a shelf that's been exported, into the **Personal**, **Project** or a custom shelf, right-click anywhere in that shelf and select **Import Shelf**.

   The **Import Shelf** dialog box displays.
4. Navigate to the folder that contains the **.msh** file you want to import, select it and click **Open**.

   Mari adds that shelf's items to the selected shelf.
5. To replace all items in the **Personal**, **Project** or a custom shelf, with items from a shelf that's been exported, right-click anywhere in that shelf and select **Load Shelf**.

   Mari warns you that all items in the current shelf will be replaced.
6. Click **OK**.

   The **Load Shelf** dialog box displays.
7. Navigate to the folder that contains the **.msh** file you want to load, select it and click **Open**.

   Mari replaces all items in the selected shelf with those from the shelf you loaded.
8. To export an item from a shelf, right-click it and select **Save Item**.

   The **Save Item** dialog box displays.
9. Navigate to the folder where you want to save the item, give it a name and click **Save**.

   Mari saves the item as an **.msi** (Mari shelf item) file.
10. To import an item that's been exported, to the **Personal**, **Project** or a custom shelf, right-click anywhere in that shelf, and select **Load Item**.

    The **Import Shelf** dialog box displays.
11. Navigate to the folder that contains the **.msi** file you want to import, select it and click **Open**.

    Mari adds that item to the selected shelf.

Note: Importing custom Photoshop brushes automatically displays them in Mari in a custom shelf. For details on importing custom brushes into a new shelf, please refer to Importing a Custom Brush.

Tip: Another learning exercise: create a custom shelf, add an item to it, export the item, export the shelf, create another custom shelf, import the item you exported from the first shelf into the second shelf, import all items you exported from the first shelf into the second shelf (import shelf), replace all items in the second shelf with the items you exported, from the first shelf (load shelf), and delete both shelves!

Creating a Floating Shelf

To turn any shelf into a custom palette, select the shelf and click . Mari opens a floating window for that shelf.

Tip: If you rename a palette, Mari renames it in the **Shelf** palette.

Tip: Spawn a couple of shelves and drag items between them. Try adding an item to the spawned palette and see how this affects the original shelf.

Painting

Read about elements of painting within Mari and the different tools you can use to achieve this. Familiarize yourself with features and functionality, as well as specific workflows you might be interested in. Here are some painting basics to get you started.

Painting in Mari is similar to other standard paint programs. Paint using the various tools, then bake it onto your model. Most tools work on unbaked paint, but one or two also work directly on the baked paint on the surface. Each tool has a set of keys that control how it works. By default, the option keys for the current tool are shown on-screen at the top of the canvas.

You can also use the Multi-Paint mode to paint into multiple shader streams at once. This painting method is a useful way to save time when working with texture packs and speeding up the painting process. More information can be found at Multi-Paint.

Paint Tools

To select a tool, select from your shelves or the **Tools** toolbar:

Paint tools			
Select *Available in Multi-Paint*	Transform Selected Objects	Paint *Available in Multi-Paint*	Roller *Available in Multi-Paint*
Blur *Available in Multi-Paint*	Paint Buffer Eraser *Available in Multi-Paint*	Vector Paint	Paint Through *Available in Multi-Paint*
Gradient	Clone Stamp *Available in Multi-Paint*	Warp *Available in Multi-Paint*	Slerp *Available in Multi-Paint*

Paint tools			
Pinup *Available in Multi-Paint*	Towbrush *Available in Multi-Paint*	Marquee Select *Available in Multi-Paint*	Transform Paint Buffer *Available in Multi-Paint*
Zoom Paint Buffer	Vector Inspector	Eye Dropper	

Painting a Constant Color

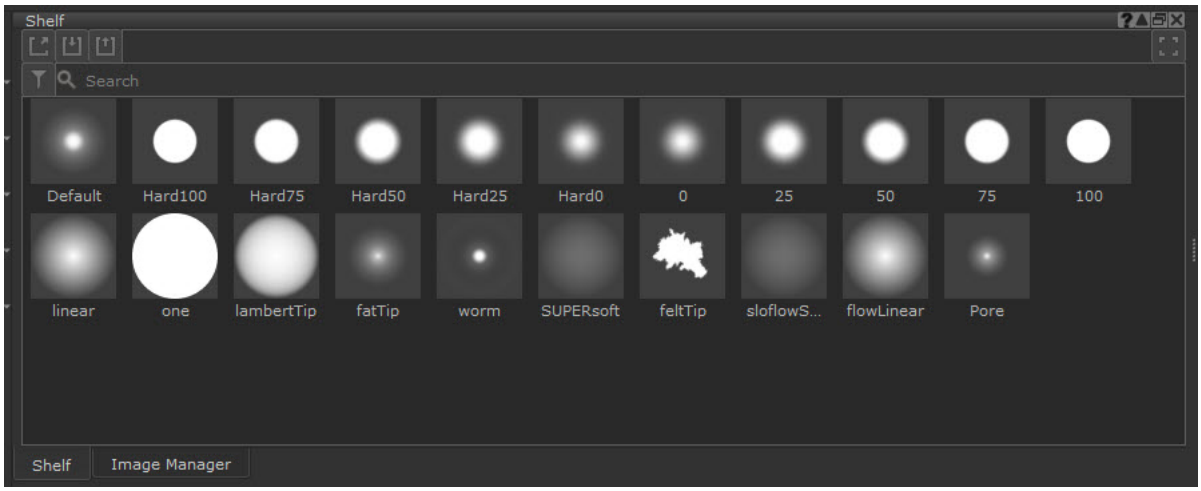
1. Click to select the **Paint** tool.
2. Click and drag to paint on the model.

Video: You can paint a straight line by clicking one end point, moving the cursor, pressing **Shift** and clicking the second end point (watch this quick demo) or pressing **Shift** and holding down the mouse button while moving horizontally or vertically (watch this quick demo).
These videos show the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Customizing Your Brush

1. On the **Tools** toolbar, select a painting tool such as **Paint**, **Blur**, **Vector Paint**, **Paint Through**, **Gradient**, and **Clone Stamp**.
2. Open the **Shelf** palette:
   - from the **View** menu, select **Palettes** > **Shelf**, or
   - right-click in the toolbar area and select **Shelf** from the dropdown menu.

   The **Shelf** palette displays.



The **Shelf** palette contains seven shelves:

- **Menu** - items that you can select from the **F9** pie selection control menu.
- **Personal** - selected items you use regularly.
- **Basic Brushes** - a set of predefined basic brushes.
- **Hard Surface Brushes** - a set of predefined hard surface brushes.
- **Organic Brushes** - a set of predefined organic brushes.
- **Brad's New Brushes** - a set of predefined brushes.
- **Project** - items just for the current project.

3. Click the shelf where your brush is stored.
4. Click your brush to select it.
5. Open the **Tool Properties** palette and change the Brush Properties.

> **Tip:** There are many options for customizing your brush, including setting values for **Paint**, **Pressure**, **Radius**, **Rotation**, **Bitmaps** to use, **Geometry** and **Noise**. You can test the brush in the scratch pad at the bottom of the **Tool Properties** palette.

6. To save your customized brush, on the **Tool Properties** toolbar of your selected painting tool, drag the brush icon to the **Shelf** palette's shelf you selected previously.

   The modified brush displays in the selected shelf.

"Painting Through" an Image

1. Open the **ImageManager** palette.
2. To load an image, click , navigate to and select the image file, and click **Open**.

   A thumbnail of the image displays, along with information about the selected image.

3. In the **Tools** toolbar, click (the **PaintThrough** tool).
4. Drag and drop the image from the **ImageManager** to the canvas.
5. Adjust the image size and position:

To...	Do this...
Resize	Grab and drag its edges or corners, or press **Ctrl**/**Cmd**+**Shift** then click and drag.
Move	Grab the "handle" in the center of the image (or press **Shift** and click anywhere on the image), and drag.
Rotate	Click and drag outside the image, or press **Ctrl**/**Cmd** and drag inside the image. Press **Shift** when dragging outside the image to rotate in increments.
Crop	Double-click the image in the **ImageManager**, drag the area you want to crop, and click .
Change the opacity	In the **ToolProperties** palette, select **Texture** > **Preview**, and change the **PreviewAlpha** (enter a number or drag the slider). Pre-multiply alpha if your image has transparency.
Reset the image	In the **ToolProperties** palette, select **Texture** > **Transform** > **Reset**.

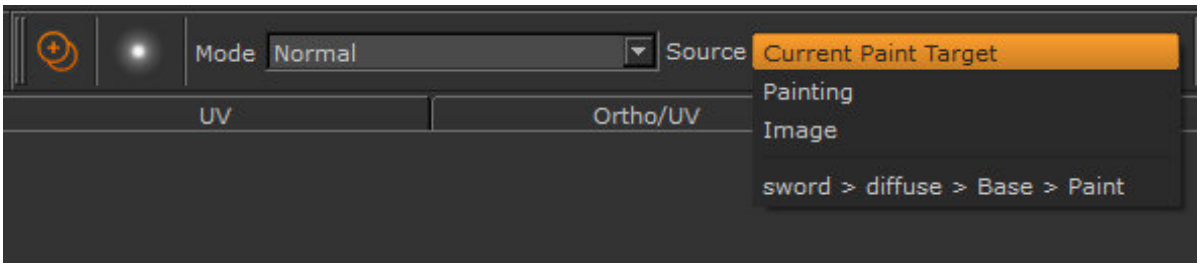
6. Paint!

Tip: You can quickly switch between the **Paint** and **Paint Through** tools by pressing **P** and **U**. To hide the image, hold the **?** key, to paint the whole image onto the model in one step, press the **'** (apostrophe) key. Toggle repeat image to paint past the edge of the floating image and have the paint continue, by pressing the **;** (semicolon) key.

Note: When using this tool in Multi-Paint, only one stream will be visible as the Paint Through tool preview. This is the Primary Stream, which is indicated by the white eye icon ( ).

Clone Stamping

1. Click to select the clone stamping tool.
2. Use the **Source** menu on the toolbar to select where to take the clone source from. You can select the current paint target, an image, or any of the paint layers in the project.



3. If you're using an image as your clone source, hold down the **Ctrl**/**Cmd** key over the image and click to select the clone source point.

> **Tip:** When cloning from a paint layer, you can clone directly from the surface. In this mode, the tool clones the paint from the paint layer surface straight up into the paint buffer directly above that point. This lets you copy the model's surface into the paint buffer so you can edit it and then re-bake.
>
> To use this mode, hold down **Shift**+**Ctrl** when you click to set the origin point. For Mac, this shortcut is **Cmd**+**Ctrl**.

4. Paint to clone your selection on the model.

Moving and Warping Paint

To move paint on the model before baking:

1. In the **Tools** toolbar, click to select the **Transform Paint Buffer** tool. See Toolbars. See the *Toolbars* section in the *Mari Reference Guide*.
2. Left-click anywhere on the paint buffer and drag to move the painting around on the model.
3. **Ctrl**/**Cmd**+left-click and drag to rotate the painting, or left-click and drag outside the paint buffer.
4. **Ctrl**/**Cmd**+**Shift**+left-click and drag to resize the painting, or grab the corners of the paint buffer and drag.

Tip: The paint buffer is visible on-screen as a white box, but it may be larger than the view window and not visible. It becomes obvious if you move or resize the painting. To reset the paint buffer to its default values click the **Reset** button in the **Painting** palette under **Paint Buffer** > **Transform**. You can also click the Reset Paint Buffer Transform ■ button on the **Paint Buffer** toolbar.

To warp paint before baking:

Using this tool...	You can...
Warp	**Shift**+click and drag to create a warp grid. Click and drag the points around to warp the paint. To increase or decrease the grid resolution, press the up or down arrow keys.
Slerp	Use the **SlerpMode** menu on the toolbar to set the mode (from **Pull**, **Grow**, **Shrink**, or **Rotate**). Click and drag to apply your effect. Erase distortion by selecting the **Eraser** mode.
Pinup	**Shift**+click to set "pins". Then click and drag to move the pins. You can use pins to protect parts of the paint that you don't want affected by the distortion.

Baking Paint onto the Model

To bake paint onto your model:

1. Make sure all patches you want to bake are selected.
2. Do one of the following:

Type **shortcut key**	*or* click on **status bar** icon
B	

Tip: Whenever you change your view of the model, it bakes automatically. You can change this setting in the **Painting Palette** under **Projection Settings** > **Projection** if required. See the *Painting Palette* section in the *Mari Reference Guide*.

Blurring Baked Paint

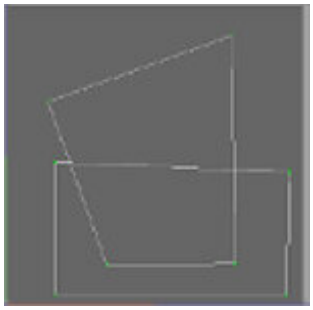
1. Click to select the blur tool.
2. Left-click and drag to blur paint baked on the surface.

Tip: As with the paint tools, you can edit the blur brush tip. When you have finished blurring the paint, you need to bake.

Modeling Requirements

Mari has certain modeling requirements that need to be met in order to use geometry, whether the format of the geometry is an Object, Ptex, Alembic, or FBX file format.

Please bear in mind that Mari allows you to import and paint models with **Overlapping UVs**, like the one shown below. However, if you paint on regions in the 3D views (orthographic or perspective), which also overlap in 2D UV view, then you may encounter painting artifacts, as Mari is forced to choose which of the conflicting overlapping paint strokes to bake down. To avoid this, you should only apply paint to one of the overlapped regions at a time in the 3D views.



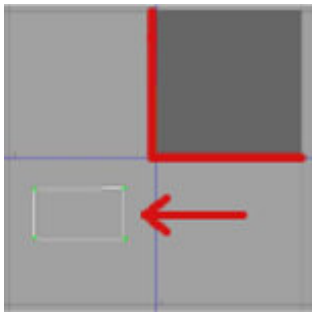
For example, if you paint a face where the left and right halves have been mirrored with overlapping UVs, then you should aim to only paint on one half of the face to avoid paint clashes.

Note: Paint clashing due to overlapping UVs cannot occur in the UV view, so you can always paint your model in that view if you find that your model is particularly tricky to paint in 3D views.

Models to paint in Mari *should not* have:

- **UVs that go over 10 on the U axis** - these are ignored.

- **Stacked UVs** - these cannot be individually selected in UV view. If you do have stacked UVs and want to select an individual UV, you need to select it in either Ortho or Perspective view first, and then switch back to UV view.
- **Negative UVs** - painting on these is not possible.



Although faces with degenerate UVs (UVs are squashed but their faces remain intact) can be loaded into Mari, they can cause issues in some cases. They do not occupy any space in UV, so it's impossible to properly paint on such faces. There is also the risk that some shaders may show undesirable lighting effects on faces with degenerate UVs.

For Ptex modeling, please see the *Ptex Modeling Requirements* section in the *Mari User Guide* for more information.

For Alembic and FBX modeling, please see the *Alembic and FBX Modeling Requirements* section in the *Mari User Guide* for more information.

Controlling the Way Mari Applies Paint to Your Model

When you paint, Mari saves your painting in a **buffer**, and then **bakes** it onto the surface of your model. You can learn About the Paint Buffer and transform paint in the buffer before baking, Masking What You Can Paint On, and apply Paint Blending Modes to control how paint in the buffer blends with what's on the surface when it bakes.

About the Paint Buffer

As you paint, Mari saves the painting since your last bake in the paint buffer. Then, when you bake, Mari saves the paint from the buffer, adds it to the surface underneath, and clears the buffer, ready for more paint.

While the paint is still in the buffer, you can make changes to it. If you don't like what you've done, you can erase sections using the **Paint Buffer Eraser** tool or just blank the entire paint buffer.

Think of the buffer as floating above the canvas. When you bake, Mari adds the paint to the current layer - but until then you can edit the buffer, move the model around underneath it, or switch to another layer and bake the paint there instead. After baking, you can set your preferences to keep the transformations for new paint you add to the buffer, or automatically reset the buffer to the Mari defaults.

Boundaries

Mari shows the boundaries of the current paint buffer as a white rectangle. By default, the paint buffer is sized slightly larger than the window (that is, it covers everything you can see), so the boundaries are not visible. As you resize it, the boundaries and current details appear.

Paint buffer resized and rotated - note that depth and resolution details display underneath.

Depth and Resolution

The paint buffer has a specific color depth and resolution, which controls the color and resolution of the paint as you apply it (rather than that of the patches that you're painting onto). For instance:

- If the paint buffer has a color depth of 8-bit (256 colors), then that's the maximum number of colors that you can use when painting. Even if you're painting onto a patch with a color depth of 16 or 32 bits (65,536 or 16.7 million colors), the paint you're applying only has 256 colors.
- If the paint buffer has a resolution of 512x512, the total buffer as it appears on-screen has that resolution. Anything you paint in the buffer is at 512x512. Any patch with a higher resolution still only receives the paint at 512x512.

On the other hand, it's also true the other way around. If you're painting onto a 512 x 512 patch with 8-bit color, using a buffer at 1024x1024 and 16-bit color, Mari still saves the patch at 512x512 8-bit. The advantage to using a paint buffer with higher quality than the resulting patches is that you can “overpaint” and use the higher quality texture and color depth to ensure good results. You can use this to help smooth out transitions between colors (preventing stepping), or to ensure that your textures are sharp (that is, at the maximum resolution for the patch).

About Masking

Within the buffer, you can control the paintable area. Mari lets you mask areas on the model. Areas covered by the mask can't be painted on. It's the electronic equivalent of putting masking tape over something you don't want to paint when you're repainting a wall.

Mari includes the following mask types:

- **ambient occlusion mask**
- **depth mask**
- **backface mask**
- **fractal noise mask**
- a configurable **edge mask**, which masks the areas on the model that are oblique to the current view, and
- a **channel mask**, where you create a channel for masking, paint in it, and use it to control where you can paint on another channel.

Tip: You can also find a variety of ready-to-use procedural masks in the Smart Masks shelf. See Smart Mask Node to read about how to use these masks, and see Ready-to-use Shelf Content to learn more about the available masks.

Mari also has several workflows that enable you to create your own masks from colors and selections. See more at Workflows for creating masks.

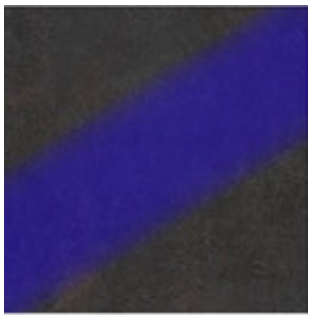
About Blend Modes

When you bake the paint buffer, Mari (by default) adds the contents of the buffer to the surface of the layer below. If you prefer, you can set the buffer to use any one of a number of other blend modes. These correspond to the layer blending modes used in 2D graphics programs such as Photoshop® or GIMP.

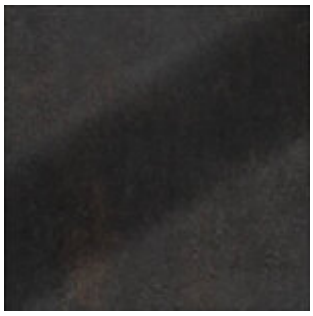
The default is for Mari to add the contents of the buffer to the surface. Other blend modes available include varieties of:

- painting the numeric difference between the buffer and the existing surface
- using the contents of the buffer as input for a dodge or burn on the surface
- using the contents of the buffer as input for a hard-light application on the surface.

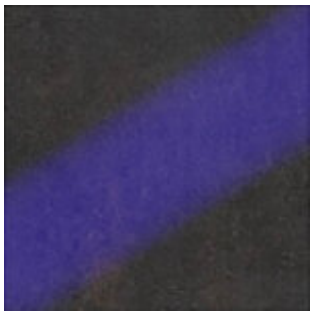
For example, here's the same paint splash in different blend modes:



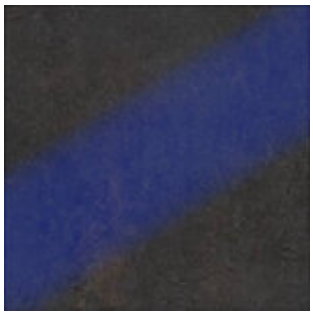
Default (over) mode.



Burn Midrange mode.



Screen mode.

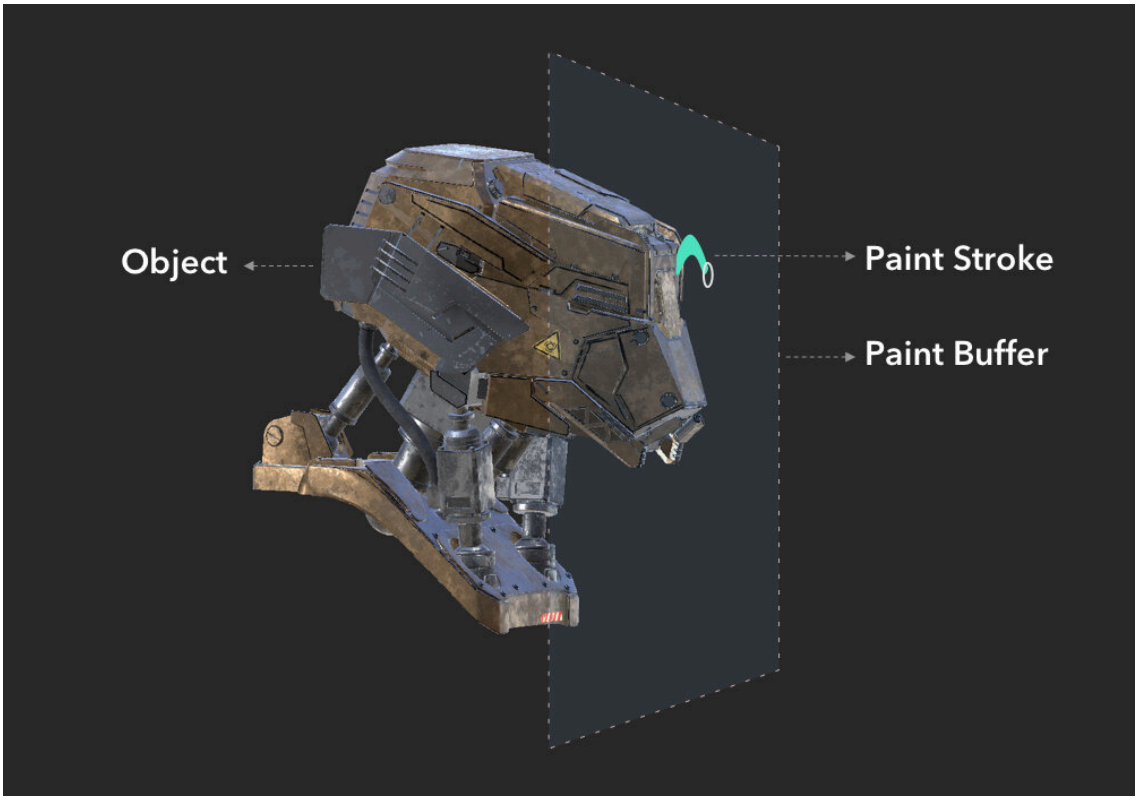


Difference mode.

Note: See Paint Blending Modes for details of all the blending modes available.

Paint Buffer

Unlike most 3D painting applications, Mari uses a projection based painting system, which is achieved by using a paint buffer. You can imagine Mari's paint buffer as a pane of glass located between the artist's view and the canvas.



The Paint Buffer is controlled by one of three Bake Behaviors, which dictate whether:

- Mari bakes and clears automatically when you change the view (**AutoBakeAndClear**),
- the paint buffer clears automatically once you bake (**ClearOnly**), or
- it needs to be baked and cleared manually (**Manual**).

Refer to Configuring the Paint Buffer for more information.

Before projecting paint onto objects, you paint into the paint buffer and then bake, which then applies the paint onto the surface of your object.

While the paint is still in the paint buffer, you can make changes to it. If you don't like what you've done, you can erase sections using the **Paint Buffer Eraser** tool (press **E** or see Erasing Paint) or just clear the entire paint buffer (see the *Paint Buffer Toolbar* section in the *Mari Reference Guide*). To adjust where the paint is projected onto the object, you can move the object around underneath the paint buffer or move the paint buffer, see Moving or Transforming Paint Buffer. Other tools can also be used to affect the paint in the Paint Buffer, refer to:

- Clone Stamping
- Masking What You Can Paint On
- Mirror Projection
- Paint Buffer Symmetry
- Distorting the Paint with Pins

- Pulling, Smudging, Growing, and Shrinking Paint
- Applying a Color Grade as You Clone
- Towing Paint Around
- Warping Paint Using a Grid
- Zoom Paint Buffer tool in the **Tools Toolbar**, see the *Tool Toolbar* section in the *Mari Reference Guide*

Video: Watch this video to learn about Mari's Paint Buffer.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

The images below show you a basic example of how the paint buffer is used in Mari. The **Bake Behavior** is set to **ClearOnly** meaning that baking paint is done manually and the paint buffer is cleared automatically.

Step 1

Paint strokes applied. The paint is saved in the paint buffer.

The paint buffer is moved around (press **M** key) and the yellow paint moves along with it as it is not baked down yet.

Step 2

The paint buffer is repositioned so that the yellow paint stroke is projected on the object.

Then the yellow paint is baked down onto the object (press **B** key). The paint is added to the current layer.

The paint buffer is moved around and clear of paint.

Step 3

An additional blue paint stroke is applied. The blue paint is saved in the paint buffer.

The paint buffer is moved around (press **M** key) and the blue paint moves along with it as it is not baked down yet, while the yellow paint does not move as it has been baked down onto the object as in step 2.

Step 4

The paint buffer is repositioned so that the blue paint stroke is projected on the object.

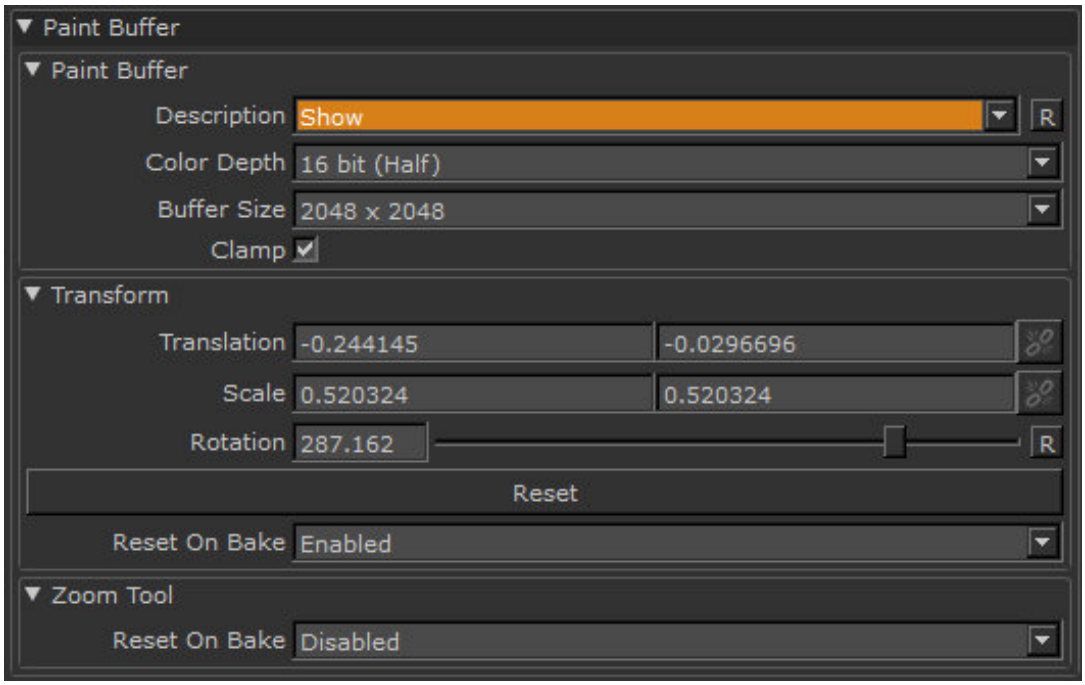
Then the blue paint is baked down onto the object (press **B** key). The paint is added to the current layer.

The paint buffer is moved around and clear of paint.

Configuring the Paint Buffer

1. Start Mari and open your project.
2. Open the **Painting** palette, scroll down, and expand the **Paint Buffer** section.

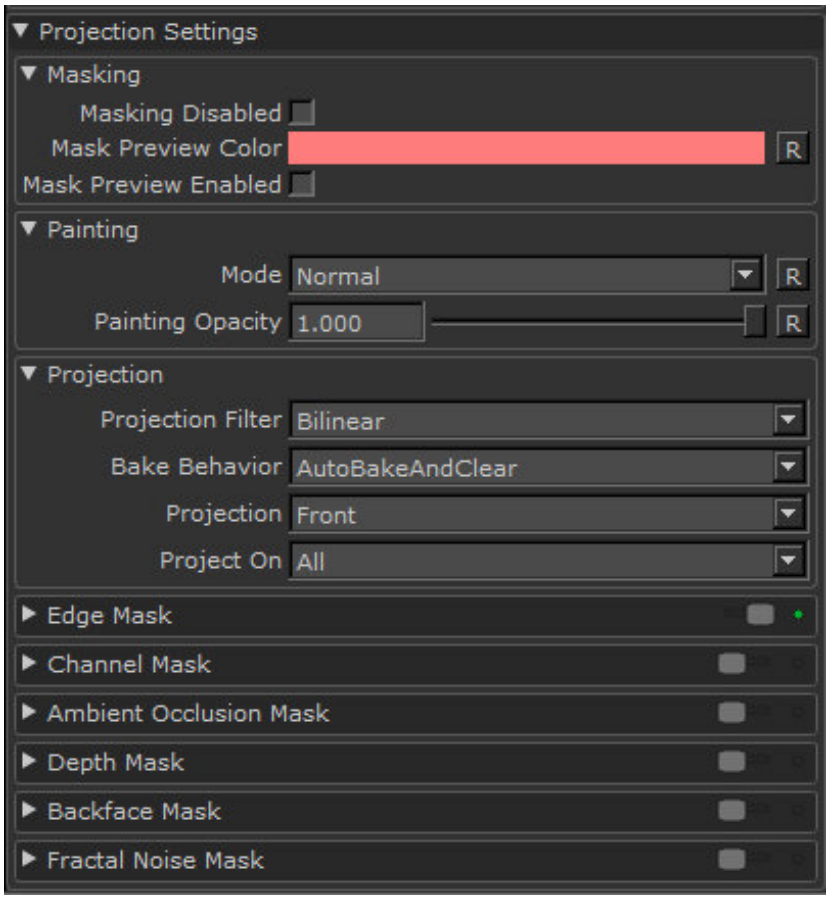
   Mari displays basic properties of the paint buffer.



3. Set paint buffer properties:
   - **Color Depth** - controls how many colors the paint buffer can hold. Uncheck this if you want to clone from or paint through a source that has values higher than this range.
   - **Buffer Size** - controls the resolution of the buffer itself.
   - **Clamp** - if checked, Mari restricts the range of the paint buffer to values between 0 and 1.
   - **Transform (Translation, Scale, and Rotation)** - used to Moving or Transforming Paint Buffer in the buffer. But it's easier to use the **Transform Paint Buffer** tool.
   - **Reset on bake** - controls whether Mari resets the buffer to its defaults after baking. If enabled, any transformations you've made to the paint buffer revert to the defaults. That is, after baking you get a new paint buffer (once again slightly larger than the visible screen). If you want to preserve your transformations after baking, change this to disabled.

> **Tip:** You should always try to set the color depth (and, if possible, buffer size) to at least as high as the highest values for the patches on your model. Mari supports a maximum **Buffer Size** of 16384 x 16384. However, this setting is not available for all graphics cards. The availability of this setting is determined by the maximum texture size supported by your graphics card.
>
> Please note that increasing the **Buffer Size**, even when your graphics card supports large textures, can slow Mari's performance. If you are using patch resolutions higher than 4K, we recommend that you zoom in to the surface when painting, to keep the resolution sharp.

4. Open the **Painting** palette. This controls how Mari projects paint onto the model when you bake the paint, and mask settings.



5. Try changing the value for **Painting Opacity**.

   This controls how opaque your paint is in the buffer. At 1.0, you see the paint exactly as you painted it. At 0.5, the paint is half transparent, and at 0 it's completely transparent.

6. To set properties for baking the buffer, scroll down and expand the **Projection** section.

   The **Projection Filter** controls how Mari calculates the colors to apply to the surface when you bake the buffer down:

   - **Bilinear** - gives a smooth projection down onto the surface.
   - **Nearest** - gives a hard-edged projection down onto the surface.

7. The **Bake Behavior** setting controls whether Mari bakes automatically when you change the view, and whether the buffer clears automatically once you bake. The options are:

   - **AutoBakeAndClear** - Mari automatically bakes and clears the buffer when you move the model.
   - **Manual** - You have to bake manually; and when you do, the paint stays in the paint buffer until you manually clear it.
   - **ClearOnly** - You have to bake manually; and when you do, the paint buffer clears automatically.

Tip: With **Manual** or **ClearOnly**, you can reposition the model underneath the buffer. This is useful, for example, if you realize that you need to get a better angle on a particular surface on the model.

8. The **Projection** setting lets you set whether Mari projects only onto the **Front** of the model (as you're seeing it) or whether paint goes straight **Through** the model (appearing on the back as well as the front).

	The paint stroke, painted onto the left side of the face.
	The result, projecting on the **Front**, rotated so you can see that it's only appearing in the front of the original view.
	When projecting **Through**, the paint stroke appears on the back as well as the front.

9. In the **Project On** settings, select whether to project onto only the selected areas, or all patches/objects in the project. For example, if you set **Project On** to **Selected Only**, you can only paint on the currently selected areas (whether it's an area, patch, or object, depending on your selection mode).

   This is useful when, for example, painting a face so you don't have to worry about accidentally overpainting onto the eyes.

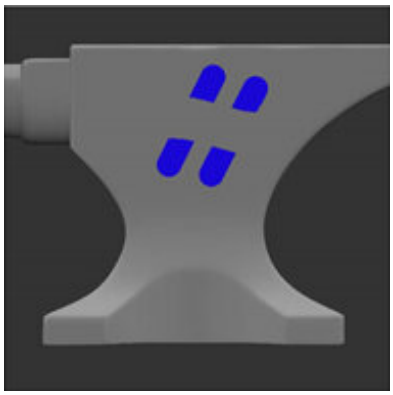
Experiment: Paint a couple of example strokes onto your model, so you can see the effects as you work. Try:

- Setting Bake Behavior to **Manual** or **Clear Only**, and then moving the model around underneath the buffer.
- The paint buffer "floats" above the model until you bake it on. For example, you could move it around to get a better angle on a particular surface that you're painting on, or to paint on a different section of the model.
- Using the **Zoom Paint Buffer** tool to zoom in on your model with the paint buffer locked and without baking the paint down.
- Changing channels (through the **Channels** menu or palette, or using the **Page Up** and **Page Down** keys). The paint buffer stays in front of the canvas, on the new channel.
- Setting **Projection** to **Through**. Paint some strokes on your model and bake them. Now rotate the model to see how the paint has "penetrated" it.
- Set **Project On** to **Selected**. Using the **Selection** tool, select a patch on your model. Paint a stroke that goes off the edge of the selected patch, and bake it. See how the paint that goes off the selection patch disappears when you bake.

Erasing Paint

1. To erase a section of paint, click on the **Paint Buffer Eraser** tool.
2. Click and drag to erase paint from the paint buffer.

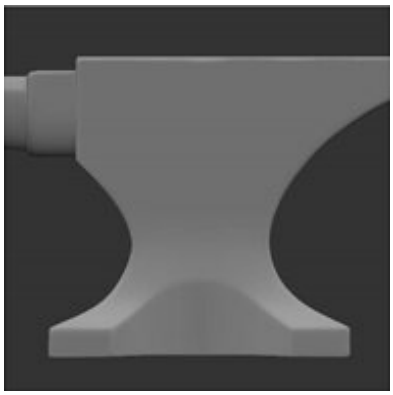
   This does not affect the underlying paint baked onto the model, only the paint in the buffer.



Tip: You can edit the brush details for the eraser (brush shape, size, and so on) the same as you would for any other brush.

3. To clear the entire paint buffer, click the **Clear the Paint Buffer** button.

   Mari clears the contents of the paint buffer.



Note: You can also clear the paint buffer by pressing **Ctrl**/**Cmd**+**Shift**+**C**.

Video: Watch Painting and Erasing in Mari for more details.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Masking What You Can Paint On

1. In the **Painting** palette, scroll down to the **Projection Settings** section.

   Masking options are under the **Masking**, **Channel Mask**, **Ambient Occlusion Mask**, **Backface Mask**, **Depth Mask**, and **Edge Mask** sections.

2. If you want to change how masking appears on-screen, change the **Mask Preview** settings.
   - You can temporarily set the **Masking Disabled** feature by selecting the check box, or pressing **Ctrl**/**Cmd**+**M**. The allows you to see the geometry without any of the current masking channels. This overrides the **Mask Preview Enabled** feature, even if it is already selected.
   - You can set the **Mask Preview Color** - click the swatch to select a color for the mask from a color picker. The color includes an alpha value, so you can have the mask displayed semi-transparent.
   - To display the masking, click **Mask Preview Enabled** or press **,** (comma).

     When this is on, Mari shows all the masking you have turned on, including channel, edge, depth, and so on.

> **Tip:** Layer masking, projection masks, and masks on projectors are all hardwired to use the red color component. If you want to use another color component for your mask value, the **Shuffle** adjustment layer allows you to change the RGBA components so that any can be replaced by another.
>
> Additionally, the **Luminosity** adjustment layer allows you to convert an RGB input to a grayscale output.

When the masking is visible, the icon displays in the status bar. Clicking this icon, as well as the button in the palette, hides the masking.

Other icons show which masking is in use. They are:

for edge masking

for channel masking

for ambient occlusion masking

for depth masking

for backface masking

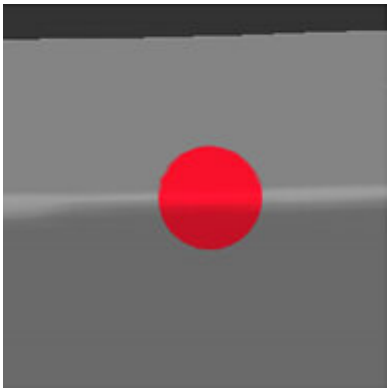
for masking disabled

> **Note:** For more information about masks, see the Painting Palette.

Video: Watch this video to learn more about on Projection Masking.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Edge Masks

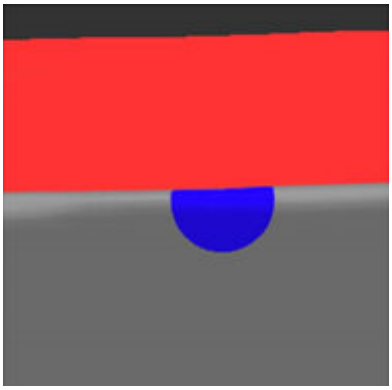
Mari projects paint directly onto the surface of your model based on your **view**. In other words, it applies paint to what you can see, from the angle you see it. If you paint something you can see, but you're not facing it directly, there might be **falloff**, where what you've painted has "smeared". For example, the images below show a spot painted onto a surface of an anvil. From the angle it was painted, it looks like a circle. But if you rotate it, you can see that it "stretched" over the edge.

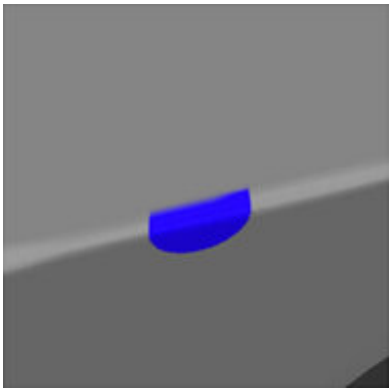


Circle painted front on.

Rotated, you can see the falloff.

To prevent this from happening, you can configure an **edge mask**, which masks the falloff area so you can't paint on it (or so that you can, but the falloff is minimized). The following illustration shows the edge mask tinged in red, and the result of the painting, again baked and rotated.



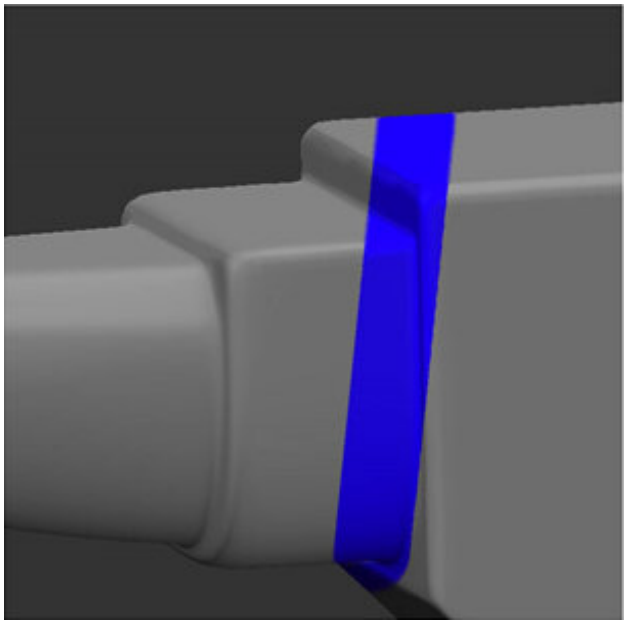


Circle painted front on, with edge mask.

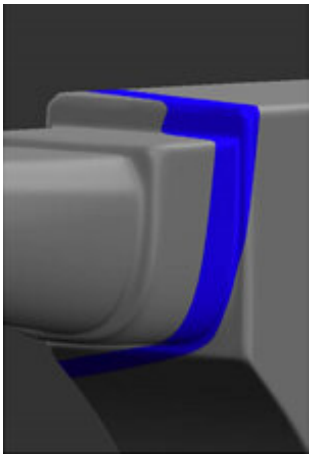
Rotated, you can see the mask prevented falloff.

Generally, you wouldn't want such a sharp edge to the paintable area, so Mari lets you specify falloff start and end points to make it fuzzy (unlike the illustration above, where the start and end points are the same). The falloff **start** is where the masking begins, a proportion of where the projection deviates from directly facing the view. From there, the projection fades out as the mask blends to the falloff **end**, at which point the projection ends completely (is completely masked) till it hits 90 degrees from facing. You can control how fast the masking falls off.

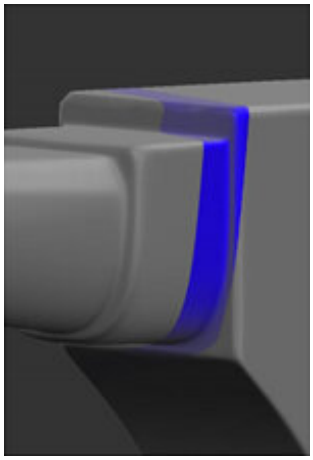
To see how this works, the following illustrations show the result of painting a stripe on the model, like this:



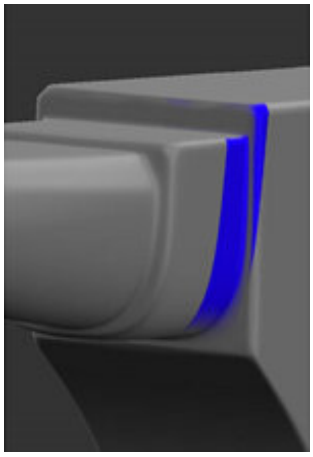
If we bake it, and then rotate to see how much “smeared” into the falloff area, you can see the results of using masks at falloff end/start points of 1.00/0.00 and 0.75/0.25:



No mask.



Mask Falloff:

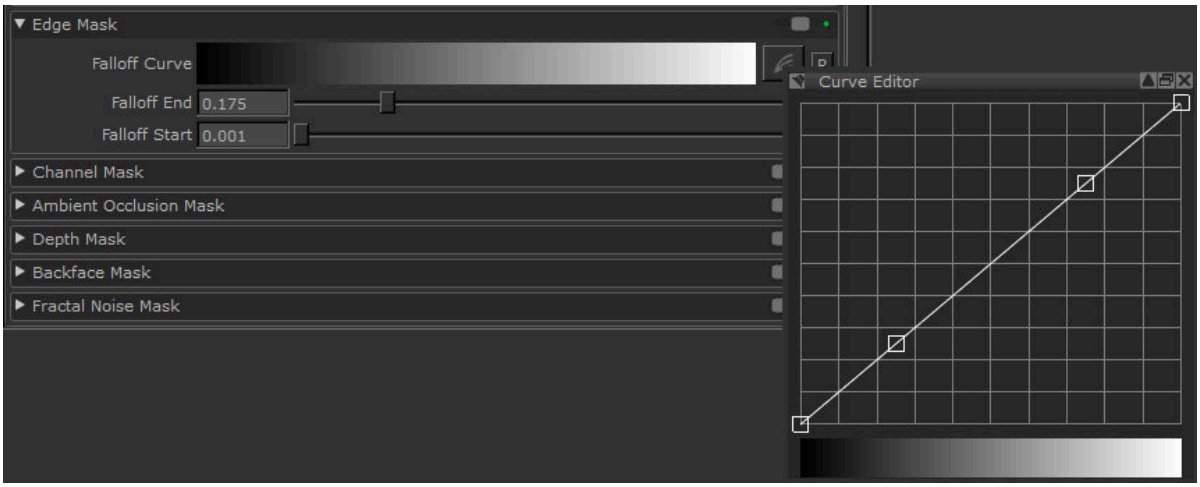


Mask Falloff:

End 1.00, Start 0.00. End 0.75, Start 0.25.

Tip: If you want to, you can set the end to a higher value than the start. This reverses the masking - the facing surfaces are masked and the oblique surfaces are paintable.

To use edge masks, set the options under **Edge Mask**:



- **Falloff Curve** - use the curve control to set how the masking falls off between the Start and End points. The start point is at the left of the curve, the end is at the right, and the degree of masking is on the vertical axis.
- **Falloff End** - where the projection completely fades and the painting is completely masked.
- **Falloff Start** - is where the masking begins, a proportion of where the projection deviates from directly facing the view, at which point the projection starts gradually fading as the masks blends to the falloff end.

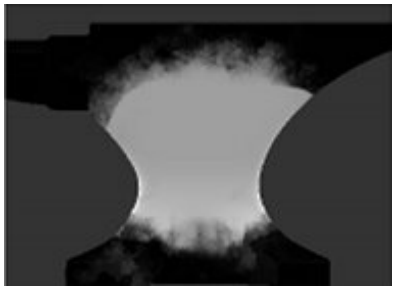
Enable the masking by turning on the toggle at the top of the section.

When enabled, the icon displays in the status bar.

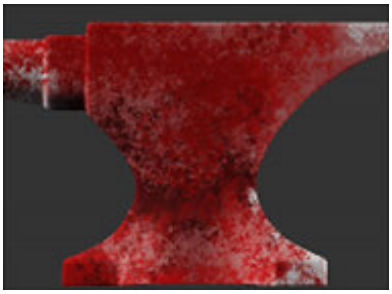
Video: You can use shortcut keys to enable/disable the edge mask (**G**) and hide/show the masking (**,**). Watch this quick demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Channel Masks

You can use a **channel mask** to manually define your own painting mask. You do this by painting into a channel and then using it to control where paint applies to the surface.



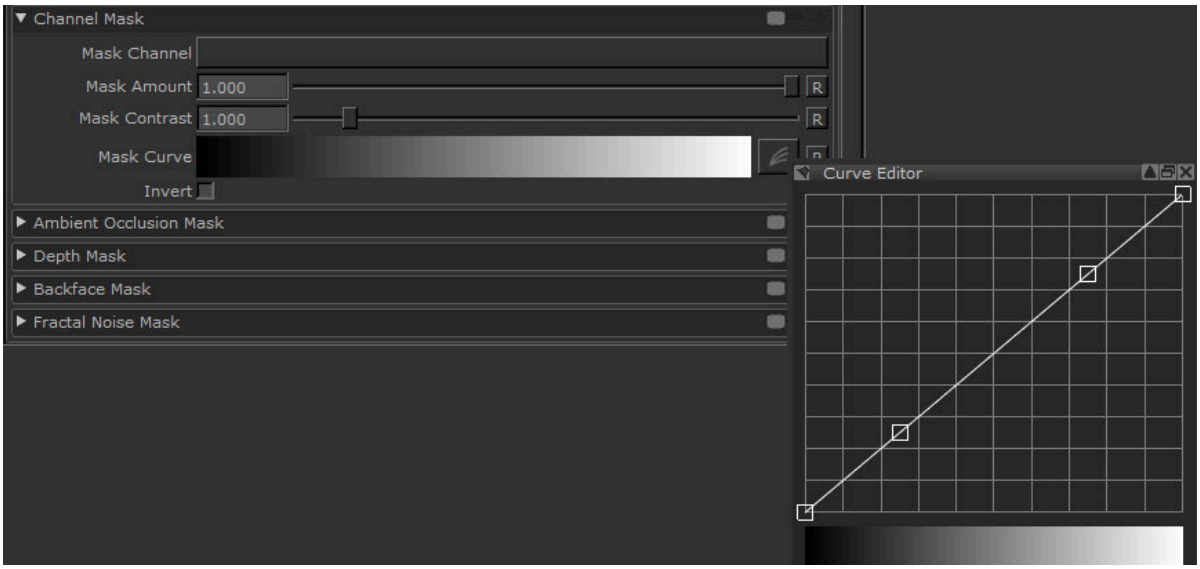
Channel mask.



Paint buffer without channel mask.

Paint buffer with channel mask – see the red masked area where the white is on the mask.

Mari applies the channel mask based on color values on the surface of the channel. This is set up through a curve control:



By default:

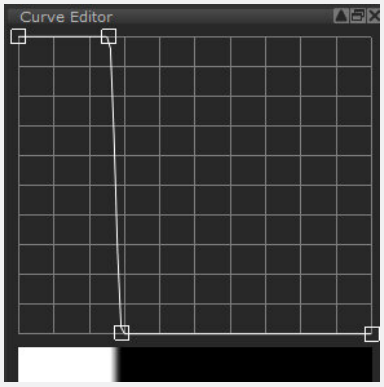
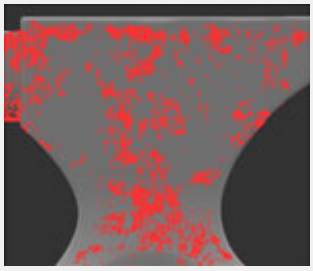
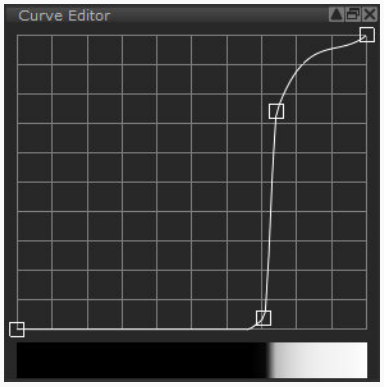
- White pixels (value of 1,1,1) are completely masked (totally unpaintable).
- Black pixels (value of 0,0,0) are completely unmasked (totally paintable).
- Other pixels are less masked as their color values approach 0, more masked as they approach 1. For example, 50% gray (0.5,0.5,0.5) creates a mask with 50% opacity.

Mask channel including colored mask data.

The resulting paint mask.

By altering the curve control, you can set how Mari masks different color values. For example, if you use a displacement map channel as your mask source, you can adjust the mask value to mask out everything except the darkest areas. This gives you a mask that covers everything except the “cracks” in your mask texture.

For example, here's the result of different settings using this displacement map as a channel mask:

Setting	Result	Notes
		This paints into the dark areas – the cracks in the texture.
		This only unmasks the lightest (that is, the highest) points in the map.

You can also set the contrast for the mask. Mari applies this contrast to the mask channel when producing the final mask.

Tip: Mari expects masks to be black on white. That is, colors are less masked as they approach 0, more masked as they approach 1. However, if you prefer to paint your masks white on black, you can invert the mask input.

To use channel masks, set the options under **Channel Mask**:

- **Mask Channel** - lets you select a channel to use as a mask (if you are using a channel mask, you need to create and paint a mask in a channel first).
- **Mask Amount** - controls the strength of the masked texture coming through, how much unmasked texture to apply. For example, at 0.5, Mari applies the unmasked paint with alpha of 0.5.
- **Mask Contrast** - controls the contrast used on the mask texture. This controls how sharply defined the mask is – at 1.0, this is just the original mask texture, lower values are fuzzier and higher values are sharper.
- **Mask Curve** - controls how the input pixel values relate to the degree of masking. The horizontal axis is the darkness of the input pixel (black to the left, white to the right), the vertical axis is the degree of visibility. By default, this is a direct line, where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.
- **Invert** - applies the mask in reverse. Mari expects masks to be black on white, where the white areas are masked and black is unmasked. However, if you prefer to paint white on black, you can invert the mask input.

When enabled, the icon displays in the status bar.

Tip: If you want to turn channel masking off, just select the blank entry from the **Mask Channel** list.

Ambient Occlusion Masks

The ambient occlusion mask lets you mask out the parts on your model that are covered by the ambient occlusion. Ambient occlusion is a way of mimicking the soft shadows produced by natural ambient light. The ambient occlusion mask lets you mask out the areas that are shadowed according to this calculation.

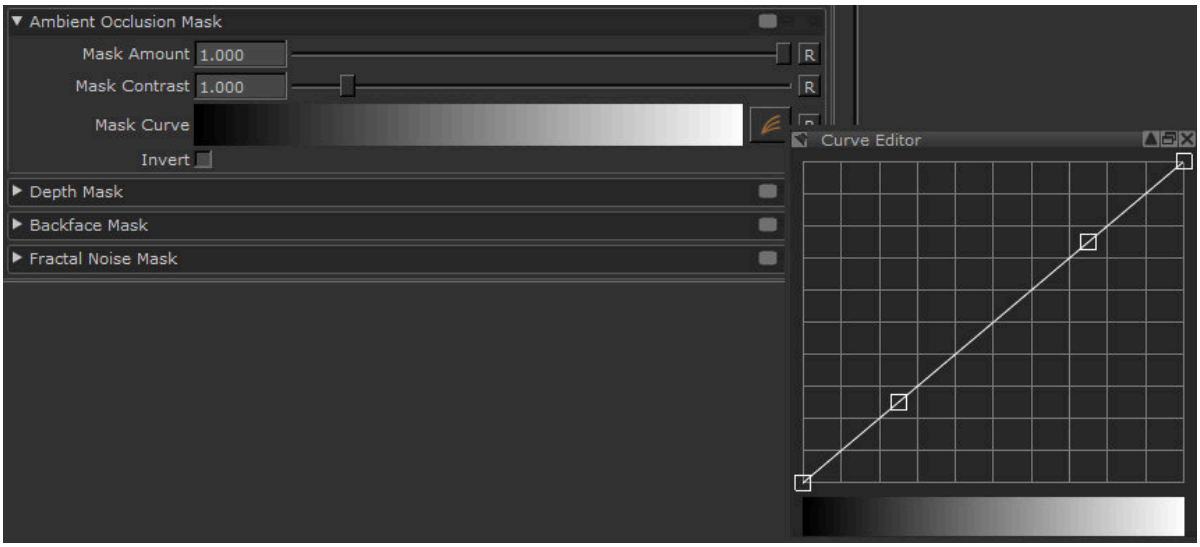
Shader showing the ambient occlusion on this part of the model.

Ambient occlusion mask - the darker the area on the original, the more it's masked.

If required, you can invert the mask, so as to mask out all the areas not covered by the ambient occlusion.

Before you use an ambient occlusion mask, you must calculate the ambient occlusion for the object. See Calculating Ambient Occlusion for more details.

To use an ambient occlusion mask, set the options under **Ambient Occlusion Mask**:

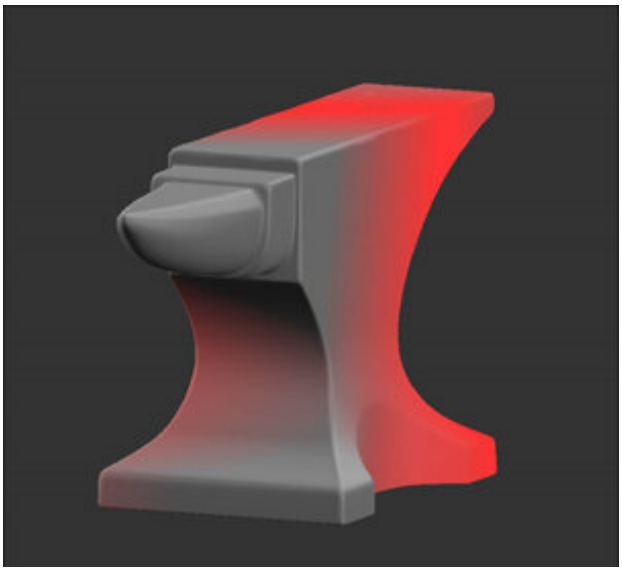


- **Mask Amount** - is the amount of masking to apply. This is from 0 to 1; higher values mask the areas that are covered by ambient occlusion, while lower values mask everything except the areas that are covered by the occlusion.
- **Mask Contrast** - controls how sharply defined the mask is. At 1.0, the mask is precisely the same as the ambient occlusion. Higher values mask the occluded areas more, lower values mask them less.
- **Mask Curve** - controls how the ambient occlusion pixel values relate to the degree of masking. The horizontal axis is the darkness of the occluded pixels (black to the left, white to the right), the vertical axis is the degree of visibility. By default, this is a direct line, where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.
- **Invert** - if this is selected, the mask data is black on white rather than white on black.

When enabled, the icon displays in the status bar.

Depth Masks

The depth mask masks out areas on the model depending on how deep they are in the scene (that is, how far from the render camera). This lets you set start and end values and quickly mask out the region between them.



Using depth mask to mask out the far part of the object.

To turn the depth masking on, set the options under **Depth Mask**:

- **Start** - is how far through the scene the masking begins. This is a proportion of the entire scene in the project, including all objects, lights, and so on.
- **End** - is how far through the scene the painting is completely masked.
- **Falloff** - sets how quickly the masking falls off between the start and end points. The interaction between the three settings can be subtle - you may need to play around with different settings to get the effect you want.

When enabled, the icon displays in the status bar.

Tip: For details on creating new channels, see Channels.

Experiment: Experiment with the masking options. Create a channel to use as a mask, then use it to control where you can paint on the model. Try setting the falloff options for the edge mask to minimum and maximum values. Try setting the falloff options so that the forward facing areas are masked and the oblique areas are paintable.

Backface Masks

Turning this on simply masks out the backfaces on your model, so you can't paint on them. There are no other configuration options.

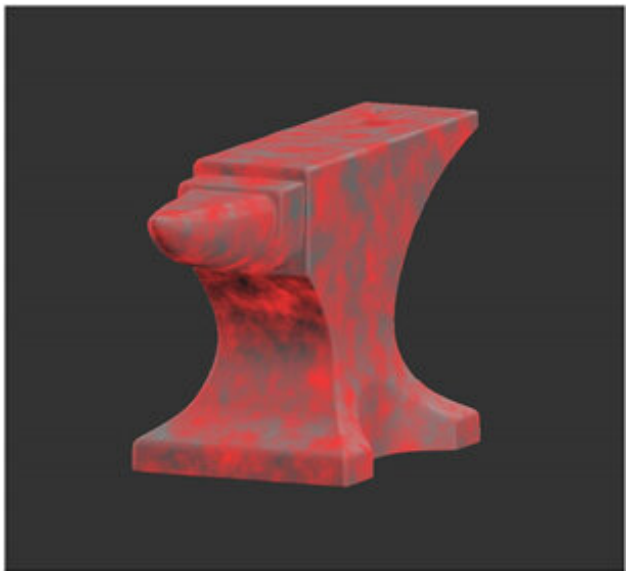
Backface mask off.	Backface mask on, masking the inside of the lens.

To turn the depth mask on, next to **Backface Mask**, click to turn the toggle on.

When enabled, the icon displays in the status bar.

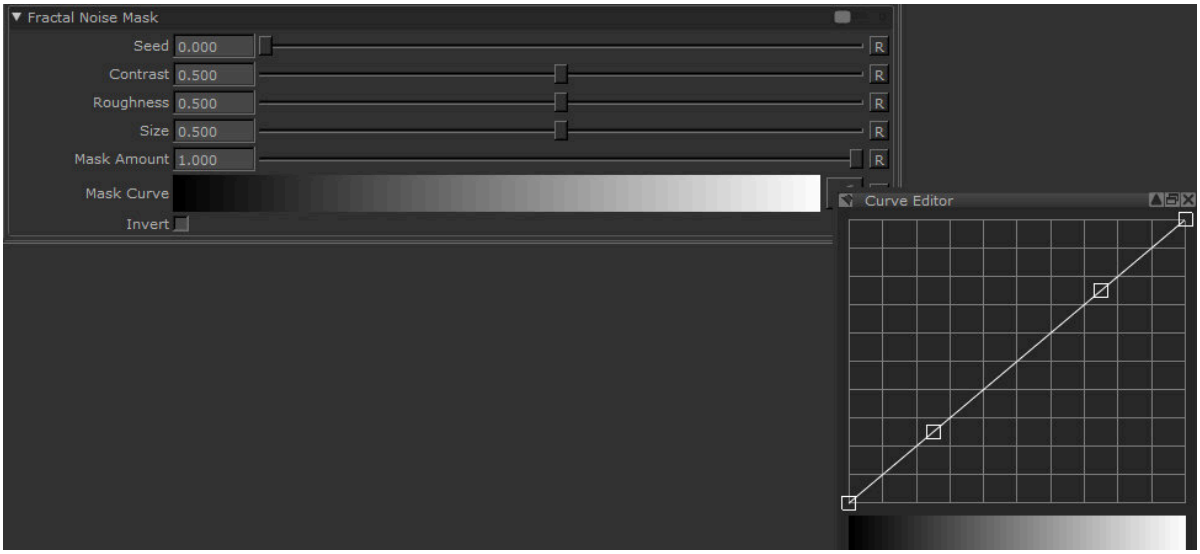
Fractal Noise Masks

The fractal noise mask is a user-controllable masking tool that uses generated noise to mask areas on the surface of an object while painting.



Using fractal noise mask to mask where paint is applied to the surface of the object.

To turn fractal noise masking on, set the options under **Fractal Noise Mask**:



- **Seed** - changes the pattern of the fractal noise mask by moving the pattern in 3D space.
- **Contrast** - controls the level of contrast applied to the fractal noise features to derive the mask.
- **Roughness** - determines the roughness of the fractal noise features.
- **Size** - Determines the size of the fractal noise features.
- **Mask Amount** - the amount that the mask affects the paint buffer. 1.0 means that the mask absolutely controls where you can paint; the effects of the mask decrease as the value gets lower.
- **Mask Curve** - controls how the fractal noise relates to the degree of masking, based on the above fields. This allows you to fine tune the contrast and pattern appearance of the fractal noise mask.
- **Invert** - if this is selected, the mask data is black on white rather than white on black.

When enabled, the icon displays in the status bar.

Adding a mask to a node

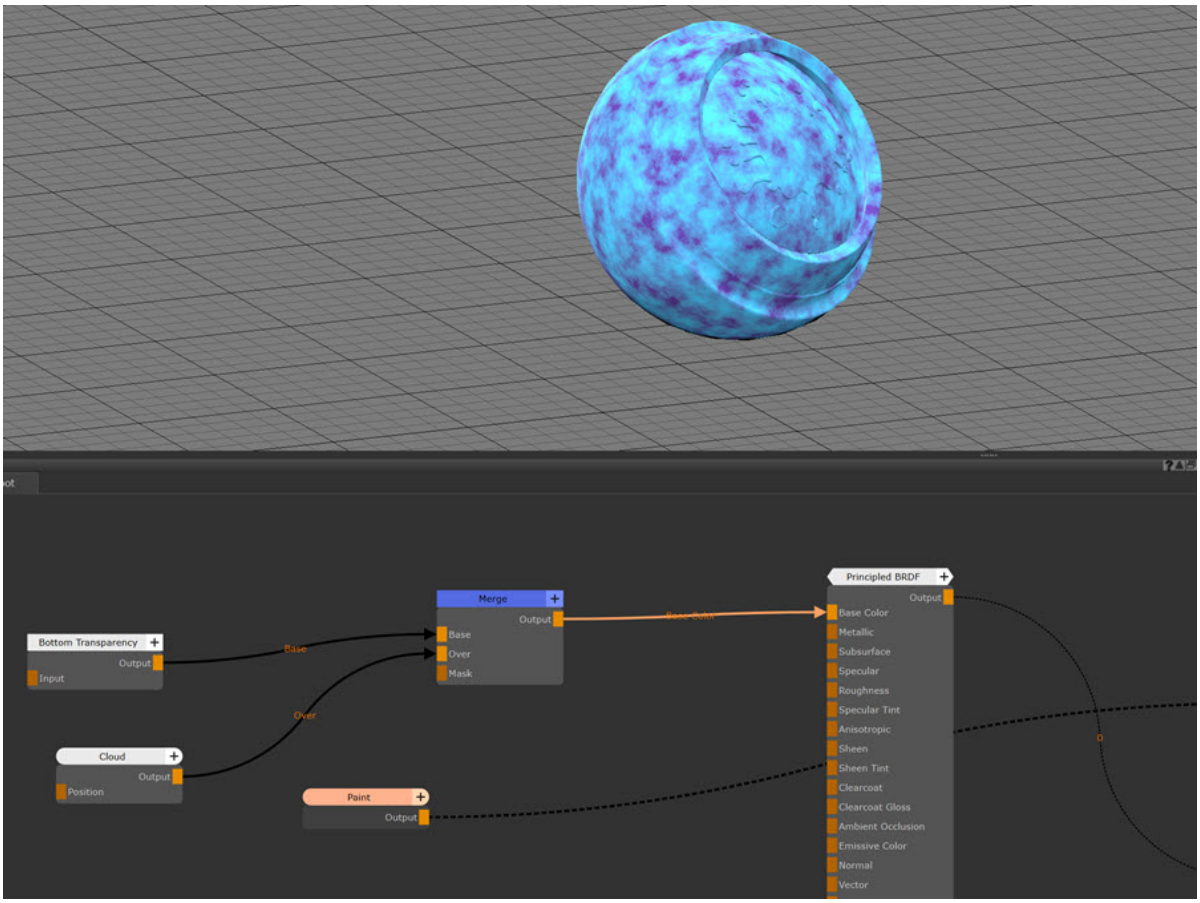
You can add a specialized mask port to any node in your graph.

To do this, go to your Node Graph and **Right-Click > Edit > Add Mask to Node**.

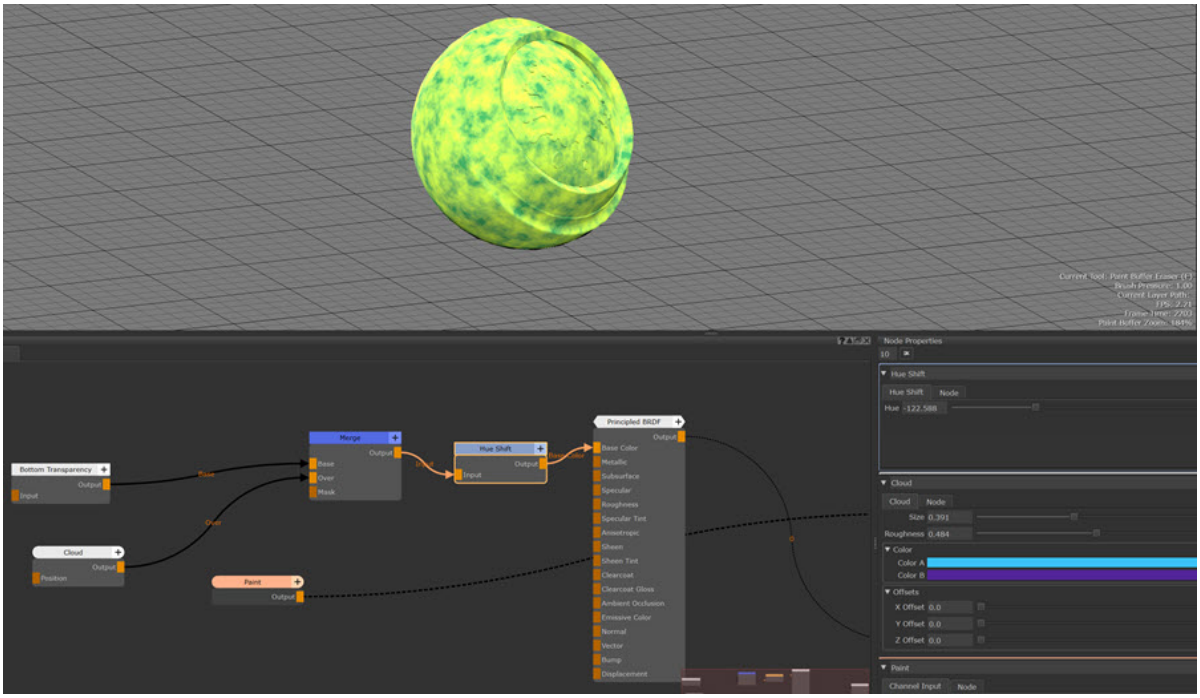
Mari automatically wraps your node selection in a group node with a dedicated Mask input. It also adds a **Mask Intensity** parameter to the node.

Mask to Node example

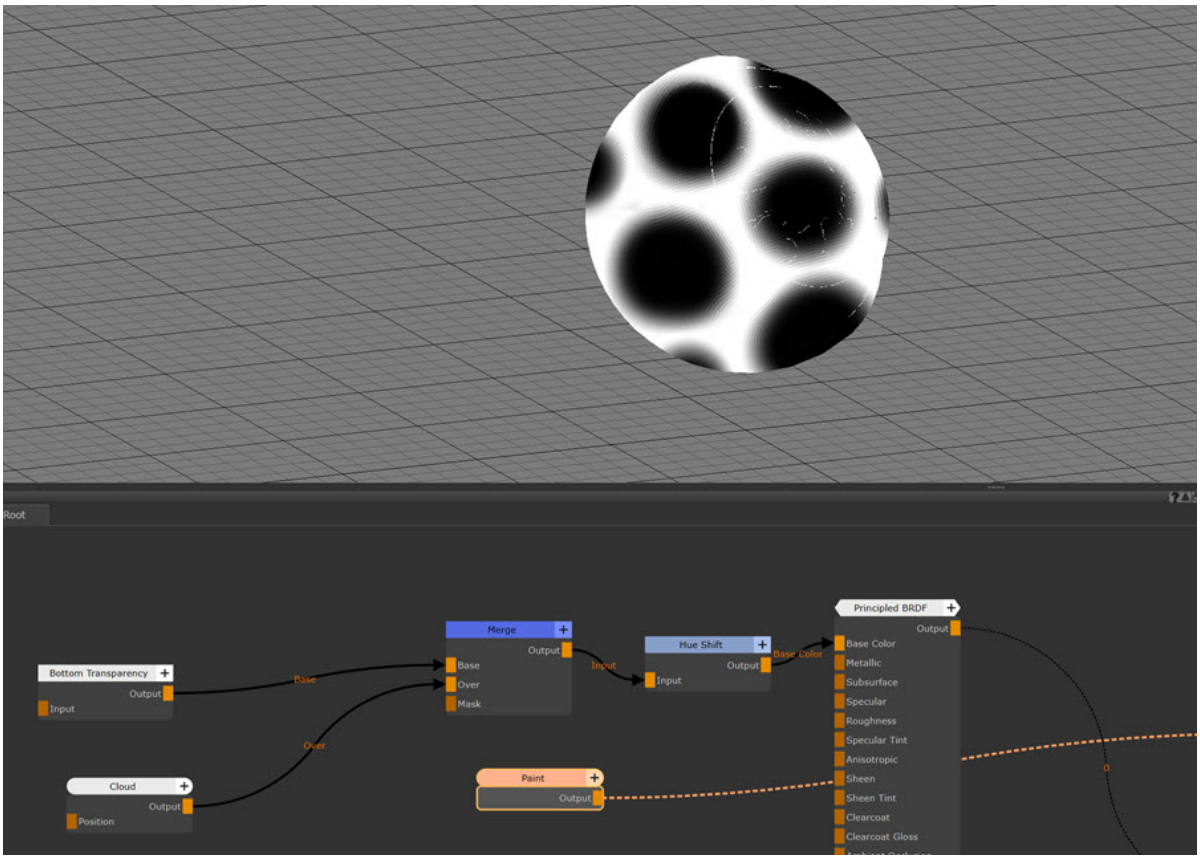
In this example we have a model with a cloud texture on it. Let's say we want to have two different color hues that create a polka dot pattern across the model.



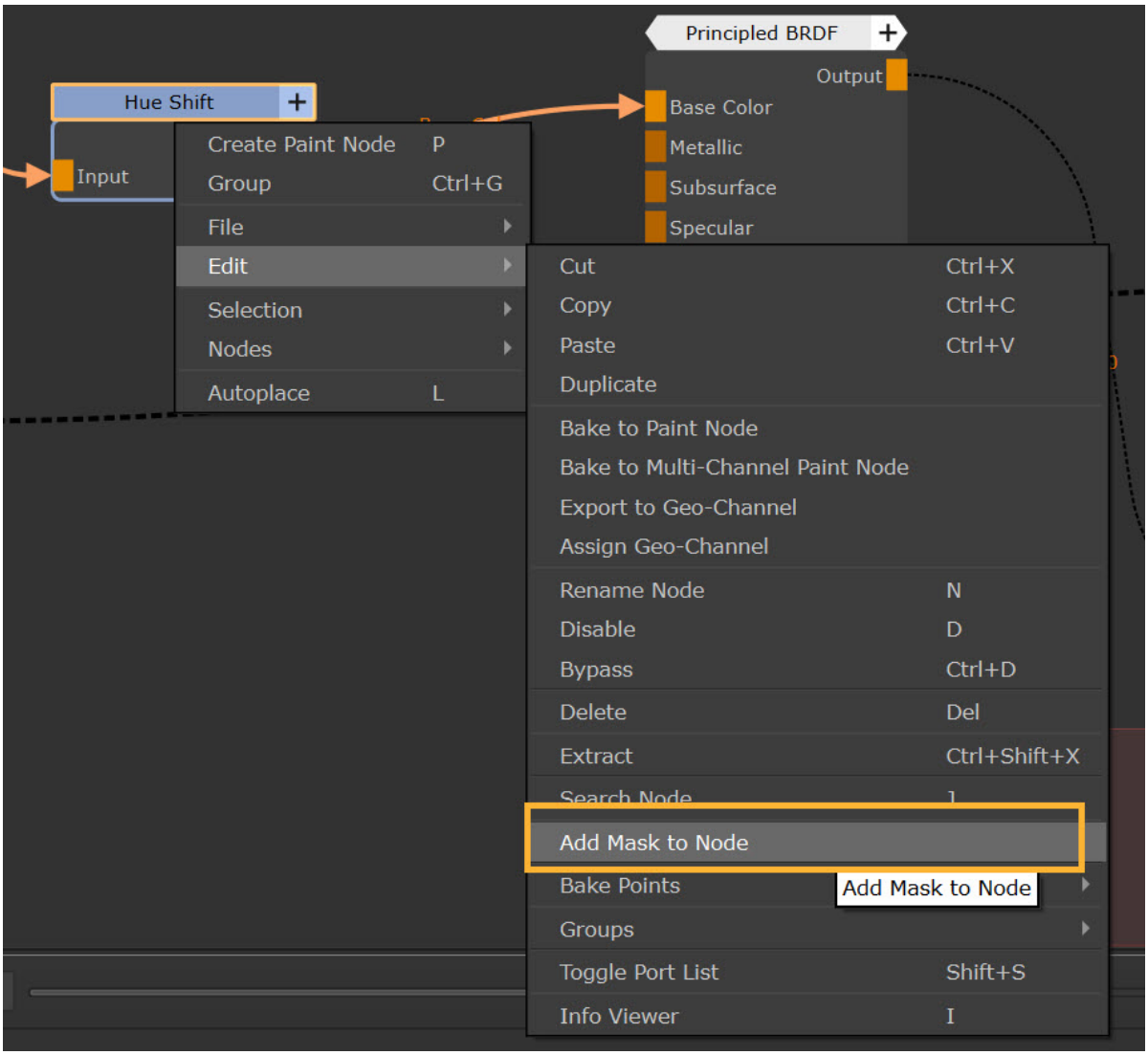
To choose the secondary color scheme, let's add a **Hue Shift** node and choose our selected color.

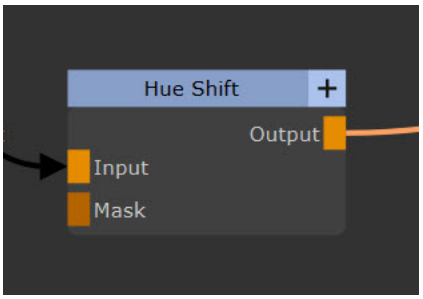


Next we use a **Paint** node to define the mask we want to apply to the model.

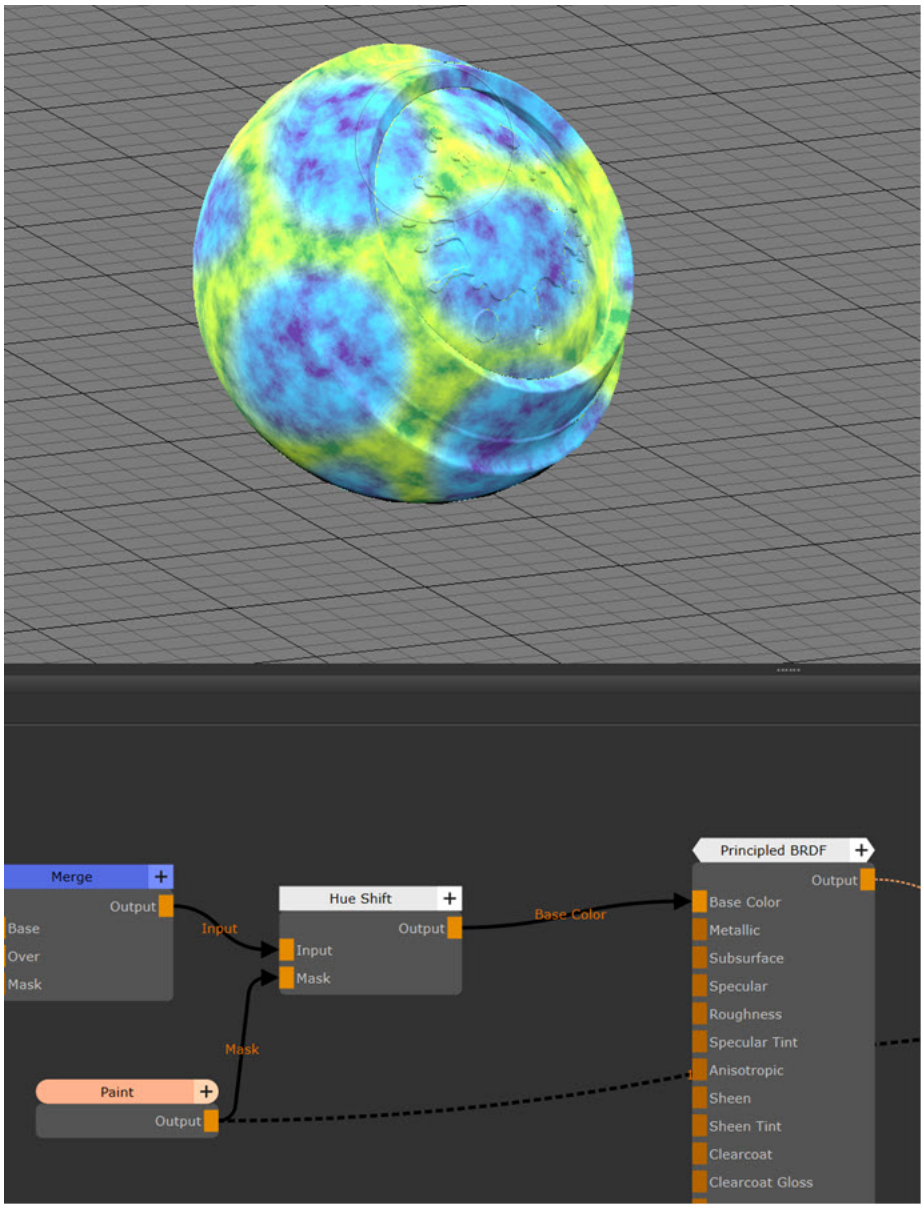


To apply the mask to the Hue Shift node, right click on the node and select **Edit > Add Mask to Node**.





Now we can just plug in the paint node into the Mask input and we get our final look:



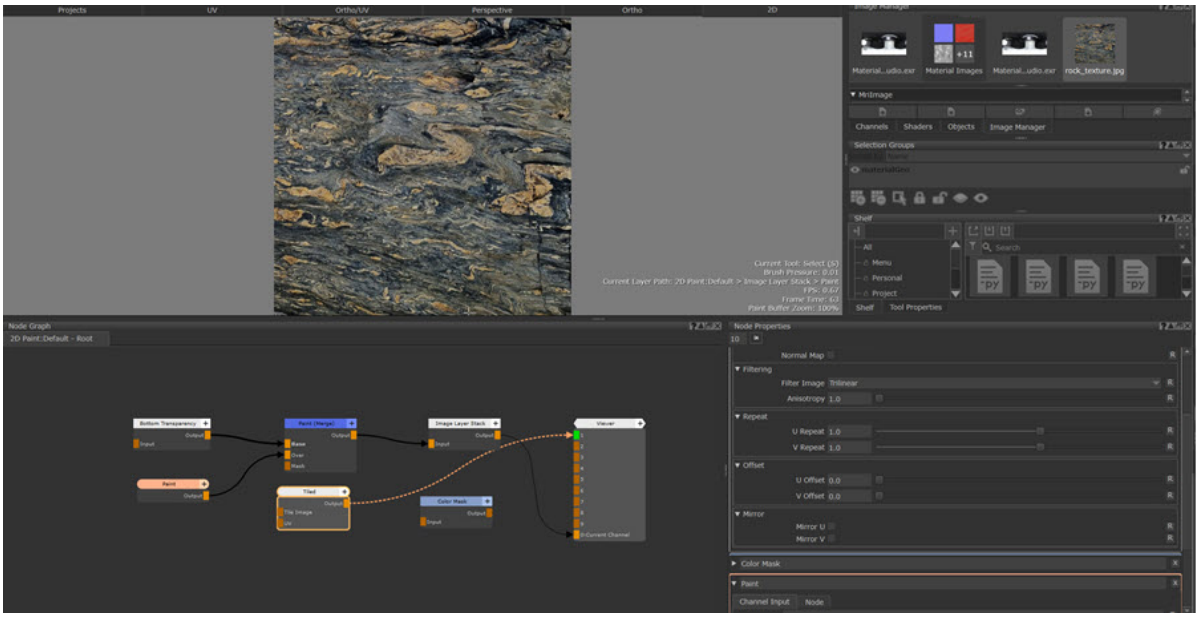
Creating masks from colors

Mari's color mask workflow enables you to create masks from a specific color (or colors) in your image. There are two nodes that provide pro

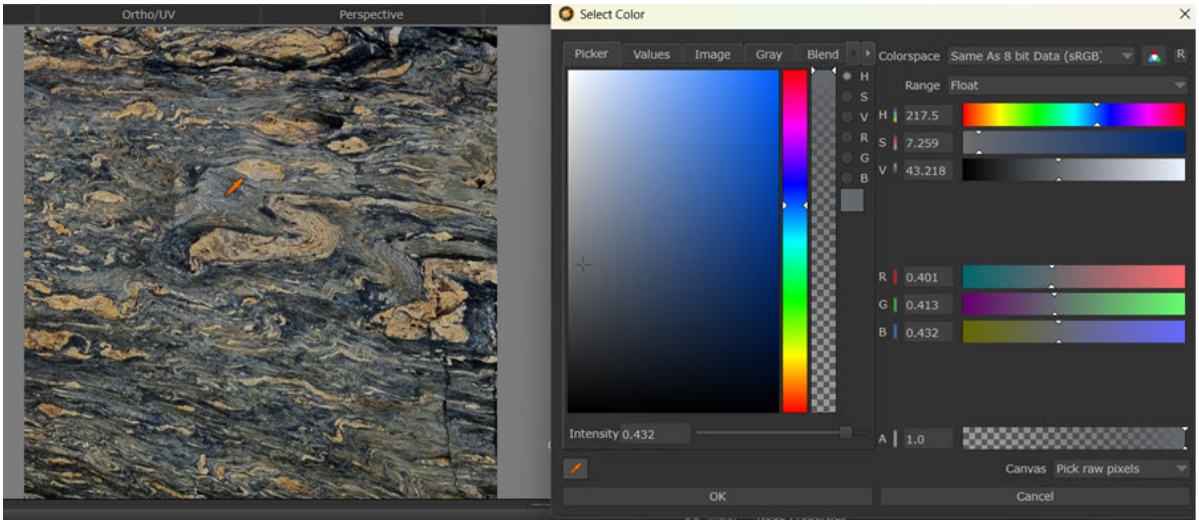
These nodes can be used on your baked ID maps, painted color blocks and layered textures.

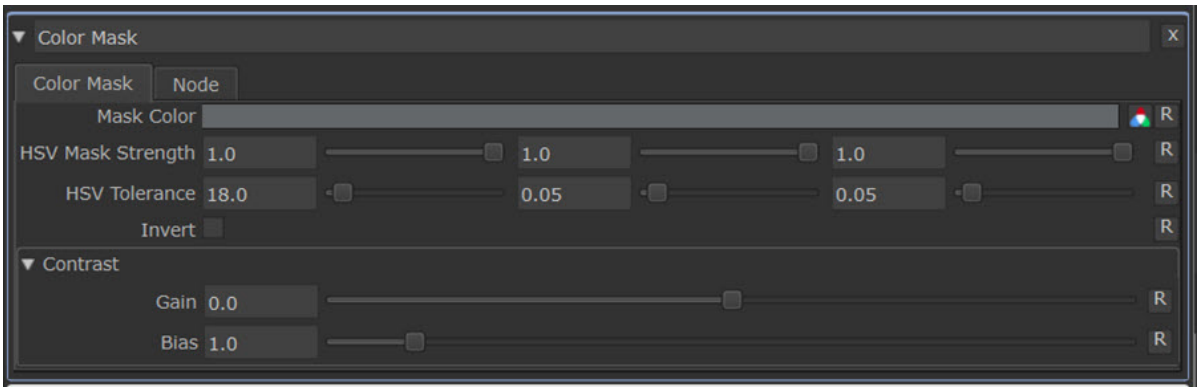
Creating a mask from a color

In this example we want to create a mask from this image by isolating a specific color, that we can then use to define the pattern of our mask. To do this, we can first create a Color Mask node.

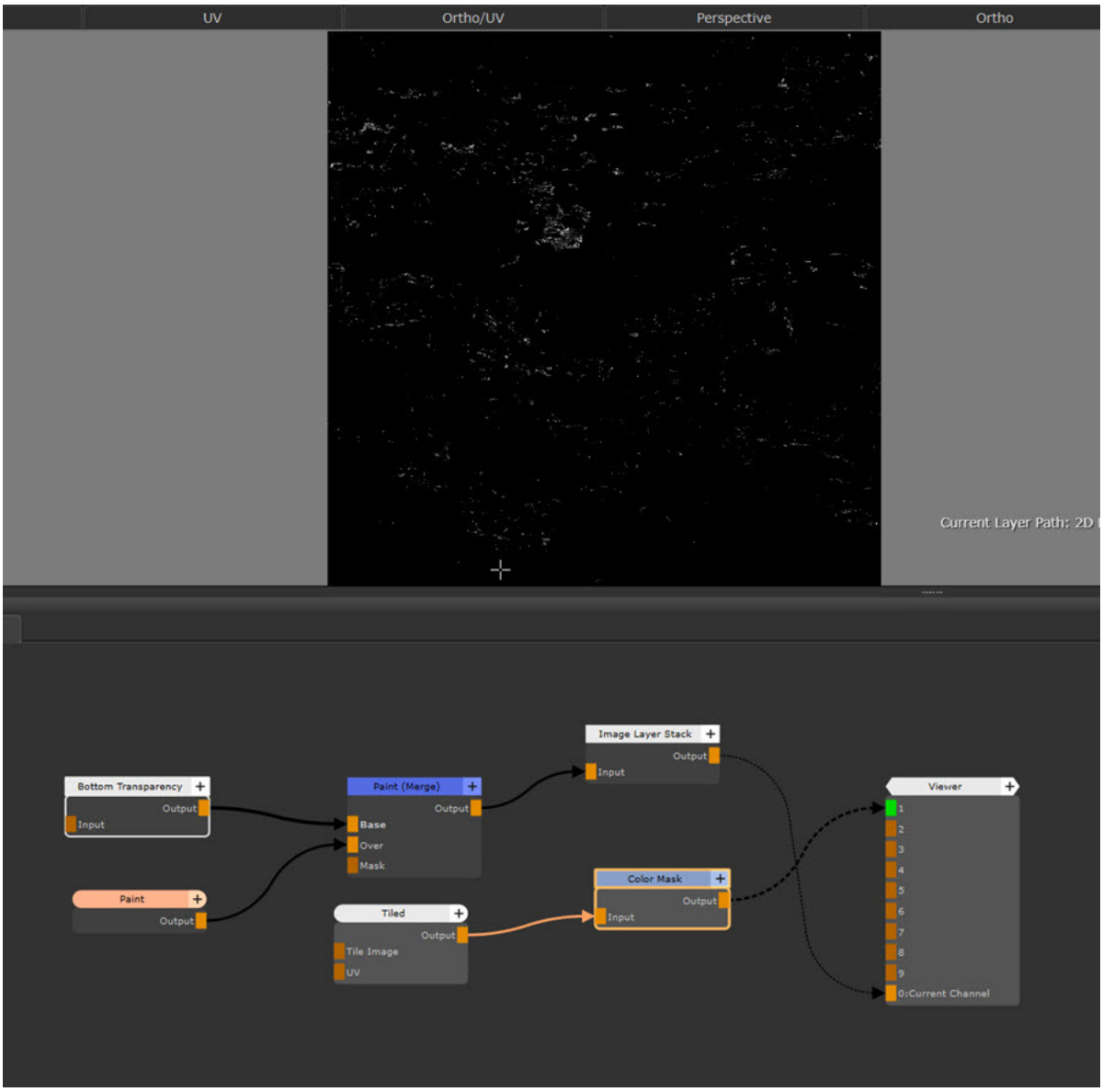


Within the Color Mask node properties, we can define the Mask Color. We are going to use the color picker option here to select a specific color in the image. In this example we're using a mid-tone gray color because we want our mask to give a scattered effect.

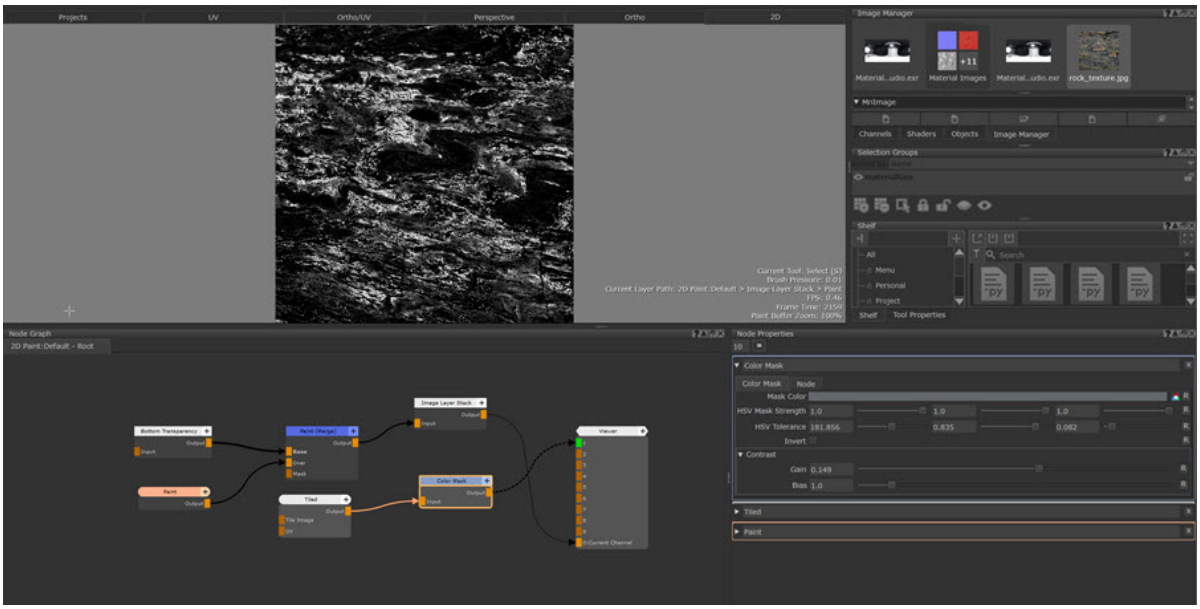




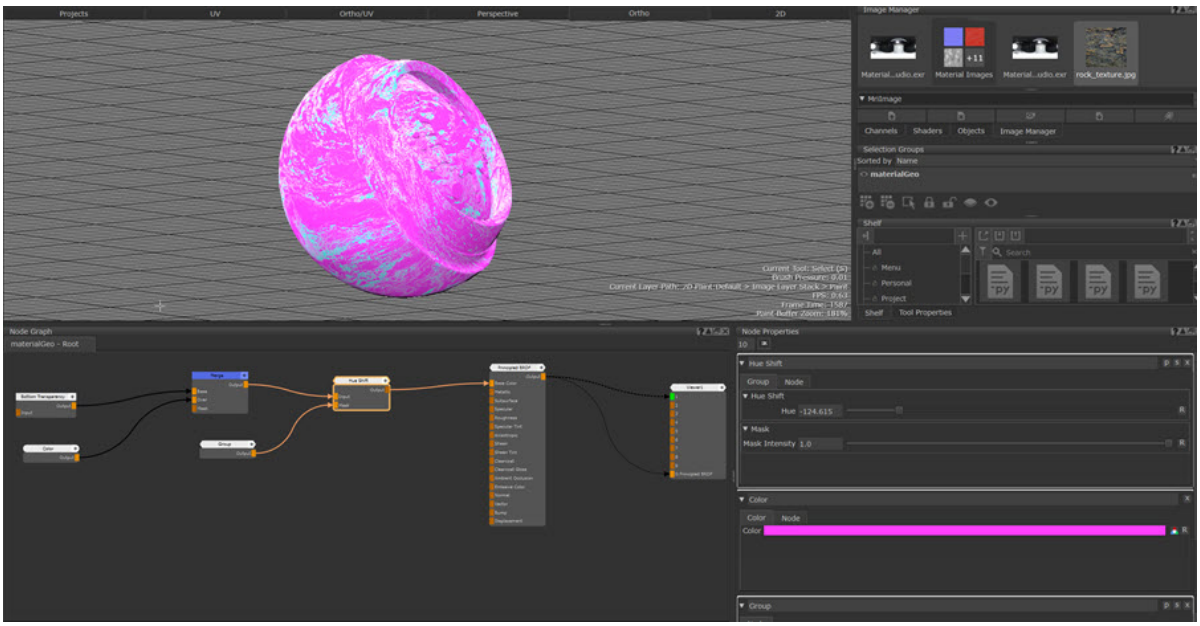
Now that the mask color has been updated, we can plug the Color Mask node into the image, so we can see the result of the mask.



From here, we can tweak the Color Mask properties.



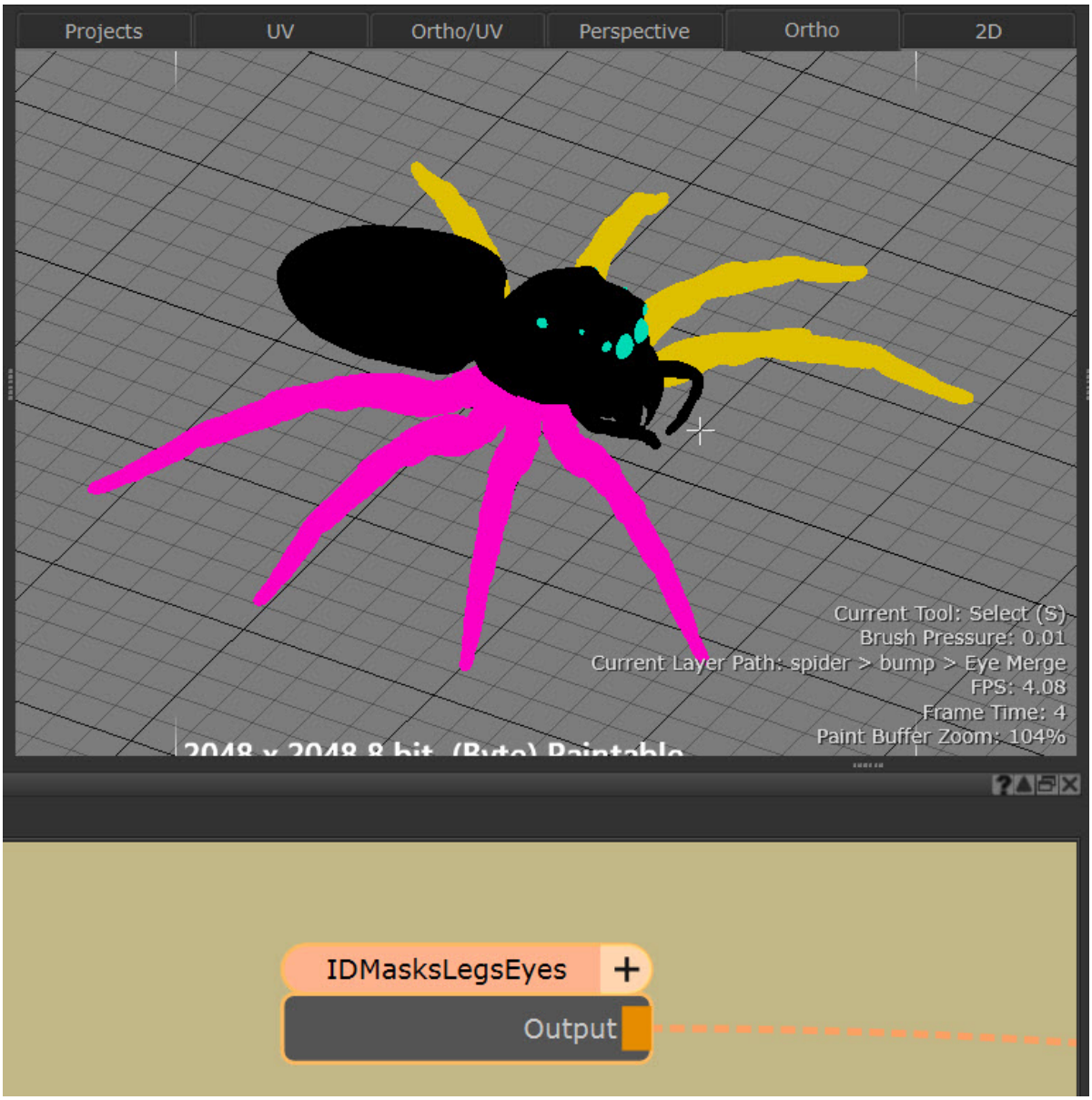
Next we are going to group up this node set and use it as a mask for our model. Because we have grouped it, it could also be exported for reuse in other projects.



Creating a mask color ID masks using the Multi Color Mask node

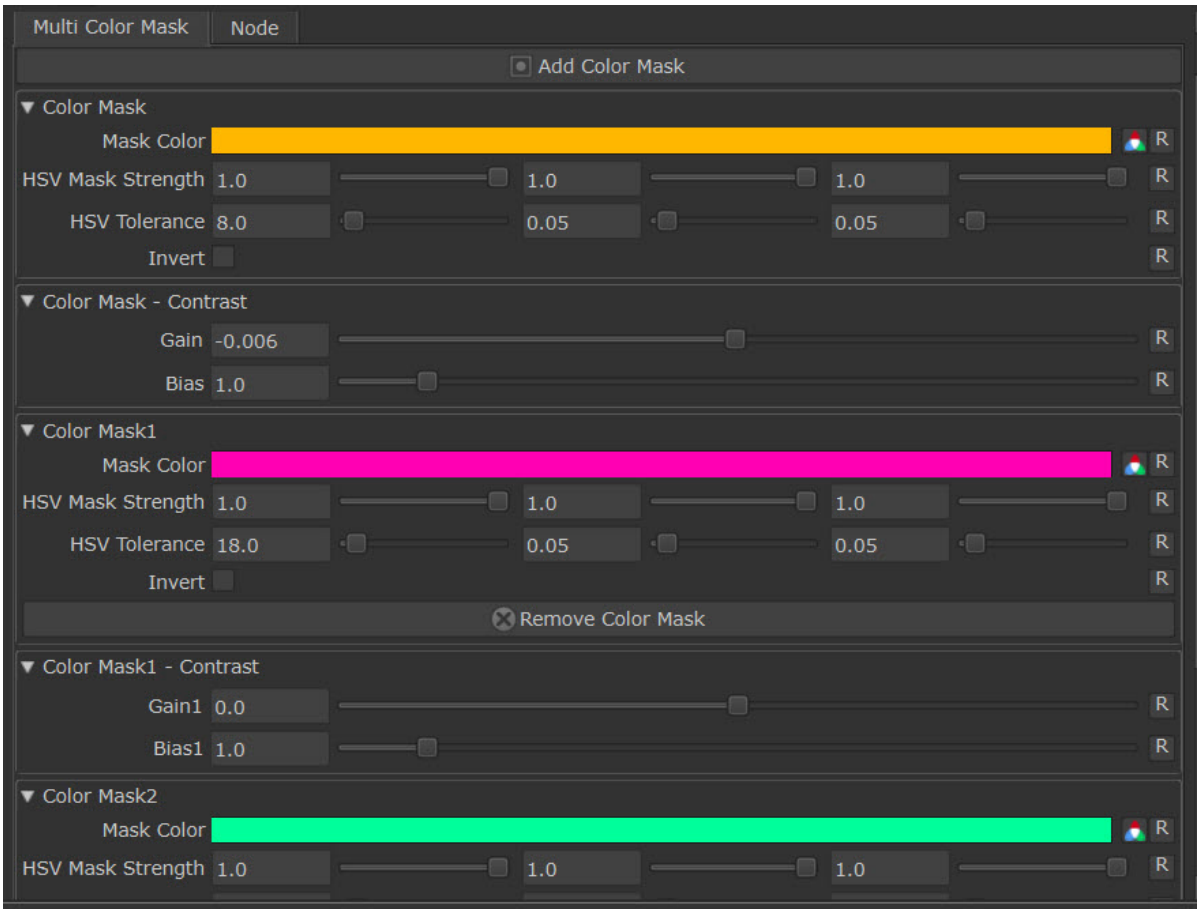
You can use the Multi Color Mask node to create bespoke masks from color ID masks.

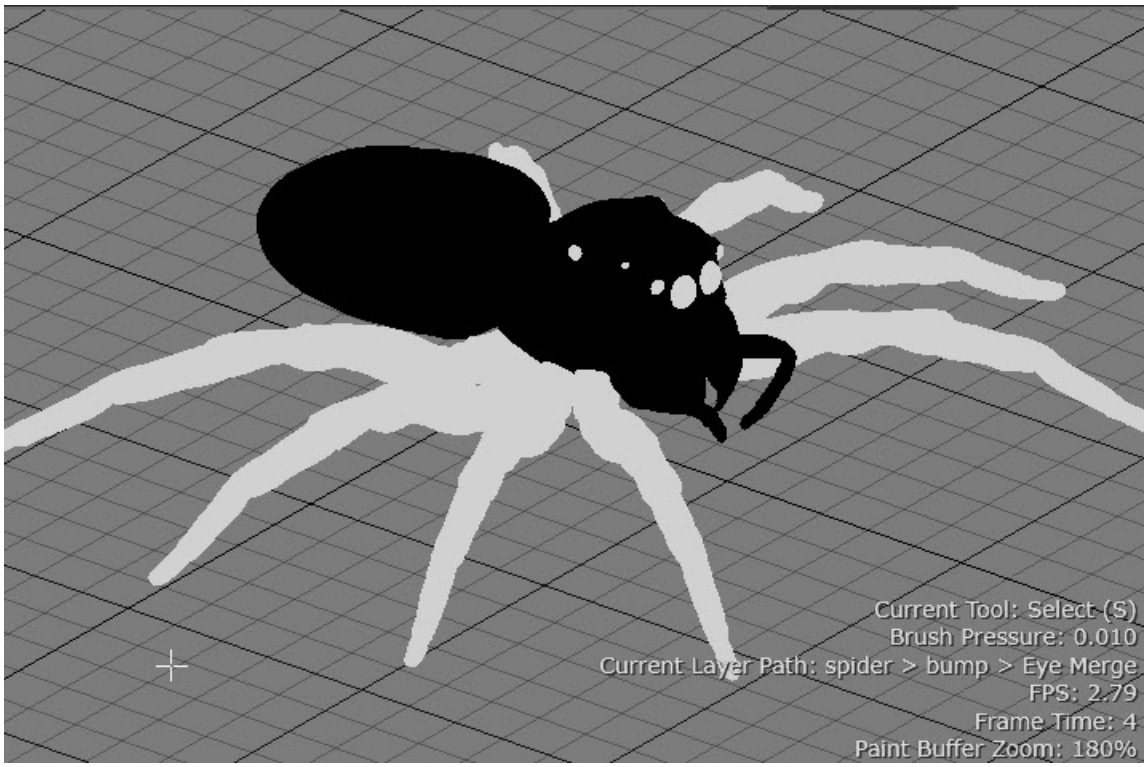
For example, if there was an effect we wanted on three different selection groups, we can make ID masks out of those groups so they are assigned a color (for more information on how to do this, go to Creating ID Masks from Selection Groups).



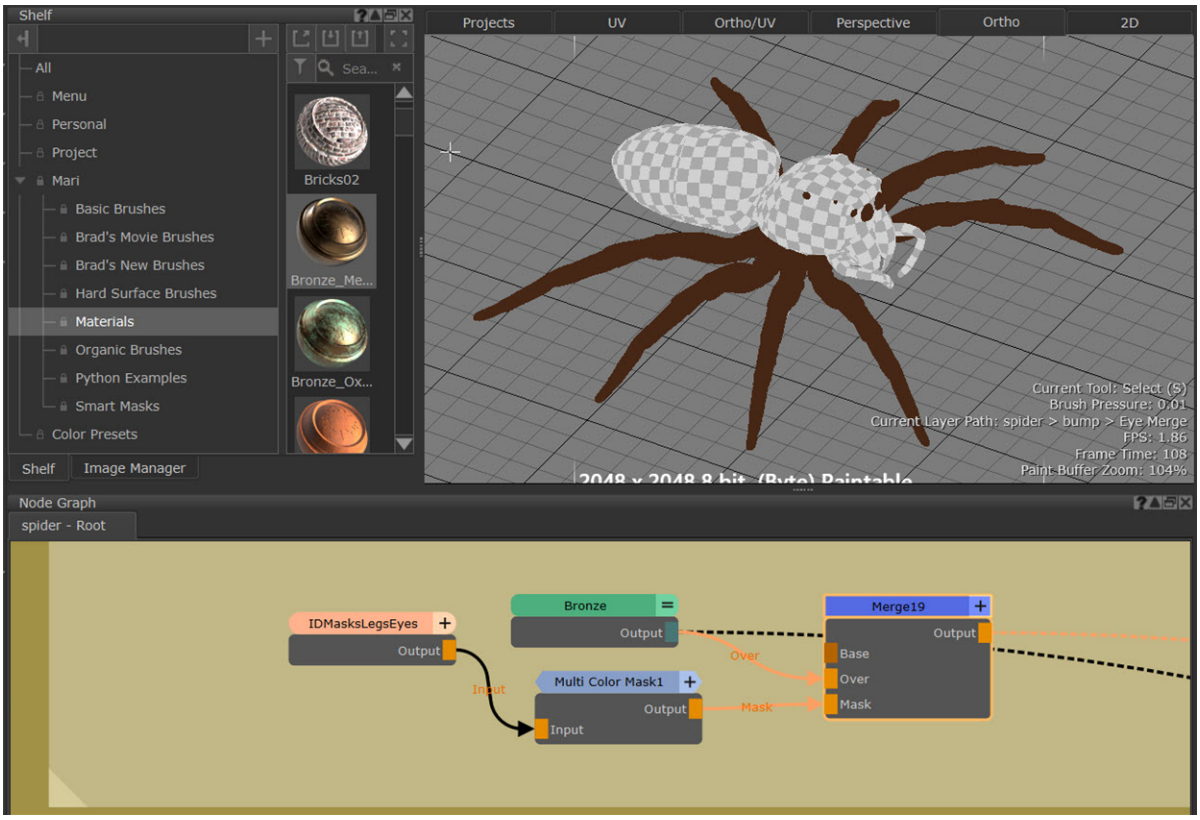
Next we can create a **Multi Color Mask** node, and add three color masks to it. Double-click on the **Mask Color** property to open up the Select Color palette, and use the color picker to assign each of the ID Mask colors to a color mask.

Note: In the Select Color palette, ensure you have the Canvas property to **Pick Raw Pixels**, so that the mask color is set by the raw color data.


Multi Color Mask
Node
Add Color Mask
Color Mask
Mask Color
HSV Mask Strength 1.0 1.0 1.0
HSV Tolerance 8.0 0.05 0.05
Invert
Color Mask - Contrast
Gain -0.006
Bias 1.0
Color Mask1
Mask Color
HSV Mask Strength 1.0 1.0 1.0
HSV Tolerance 18.0 0.05 0.05
Invert
Remove Color Mask
Color Mask1 - Contrast
Gain1 0.0
Bias1 1.0
Color Mask2
Mask Color
HSV Mask Strength 1.0 1.0 1.0


Current Tool: Select (S)
Brush Pressure: 0.010
Current Layer Path: spider > bump > Eye Merge
FPS: 2.79
Frame Time: 4
Paint Buffer Zoom: 180%

You can now use this Multi Color Mask node in the rest of your project, and it will treat the three colors as a single mask.



For more information go to Color Mask node and Multi Color Mask node pages.

Creating a mask from a selection

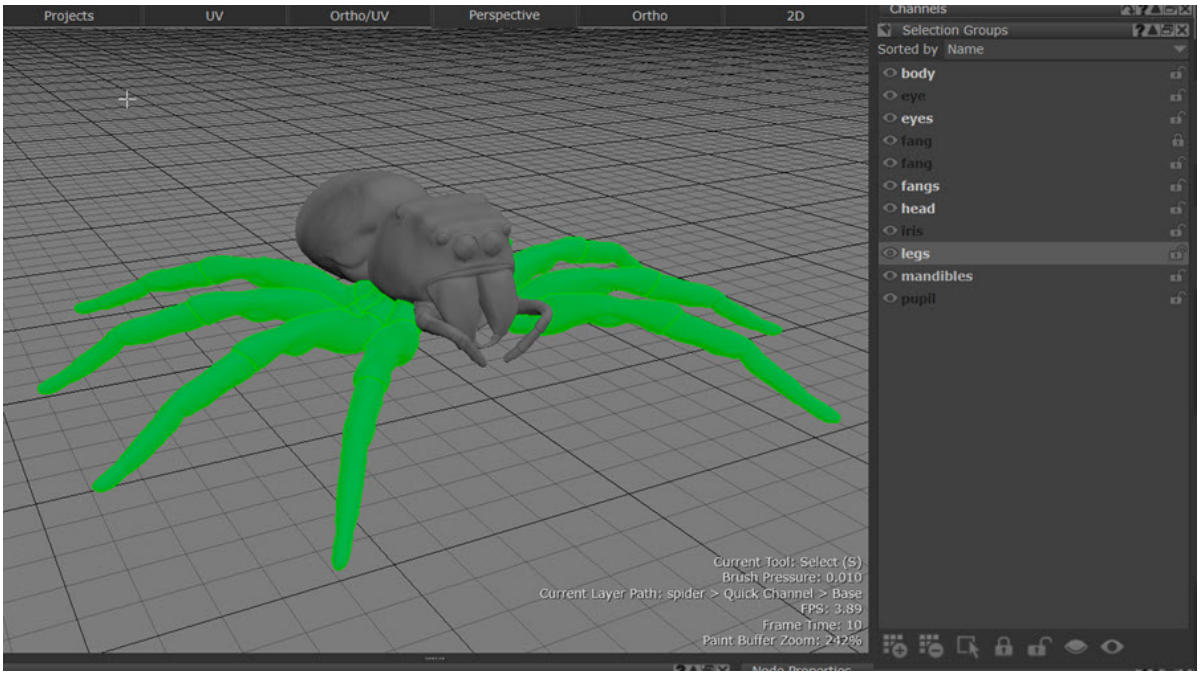
Creating a mask from an active selection or existing selection group gives you fine-tuned control to paint on and apply specific effects to certain sections of your model.

To create a mask from a selection:

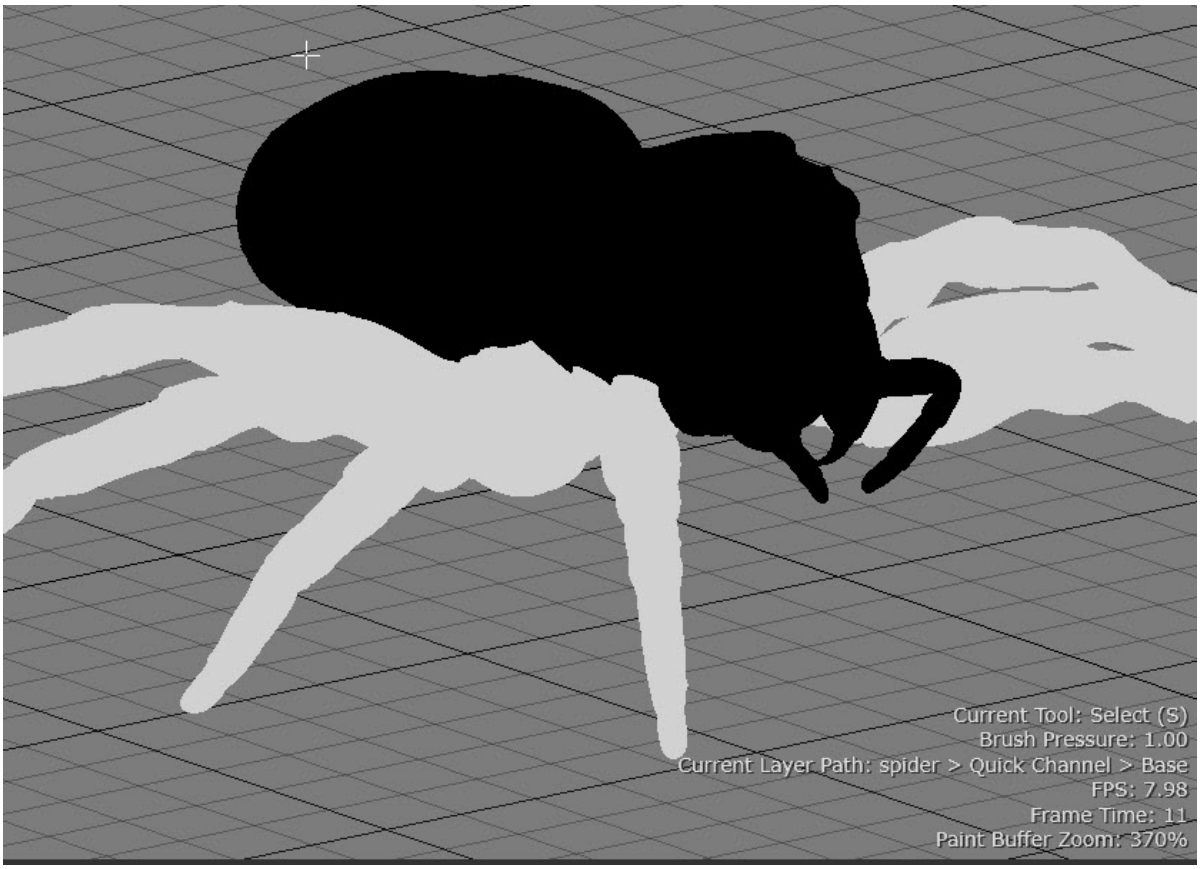
1. If you want to use an existing selection group, right-click the selection group you want to make a mask from. Otherwise, select the desired areas you want to mask using the Select tool in the canvas, and then right-click in the canvas.
2. In the context menu, select **Mask > Create mask from Selection**.
3. Define the **Size** and **Depth** of the mask.
4. You will need to enter either a Paint node or a Geo-Channel node option to be able to create the mask

- If you want to create a Paint node for the mask, enter the name of the node in the text input.
- Define whether the mask should be assigned a Geo-Channel (you can either create a new Geo-Channel or select an existing one). If you choose an existing Geo-Channel, the Geo-Channel image set will be overwritten with the new mask.

5. Click **Ok**.



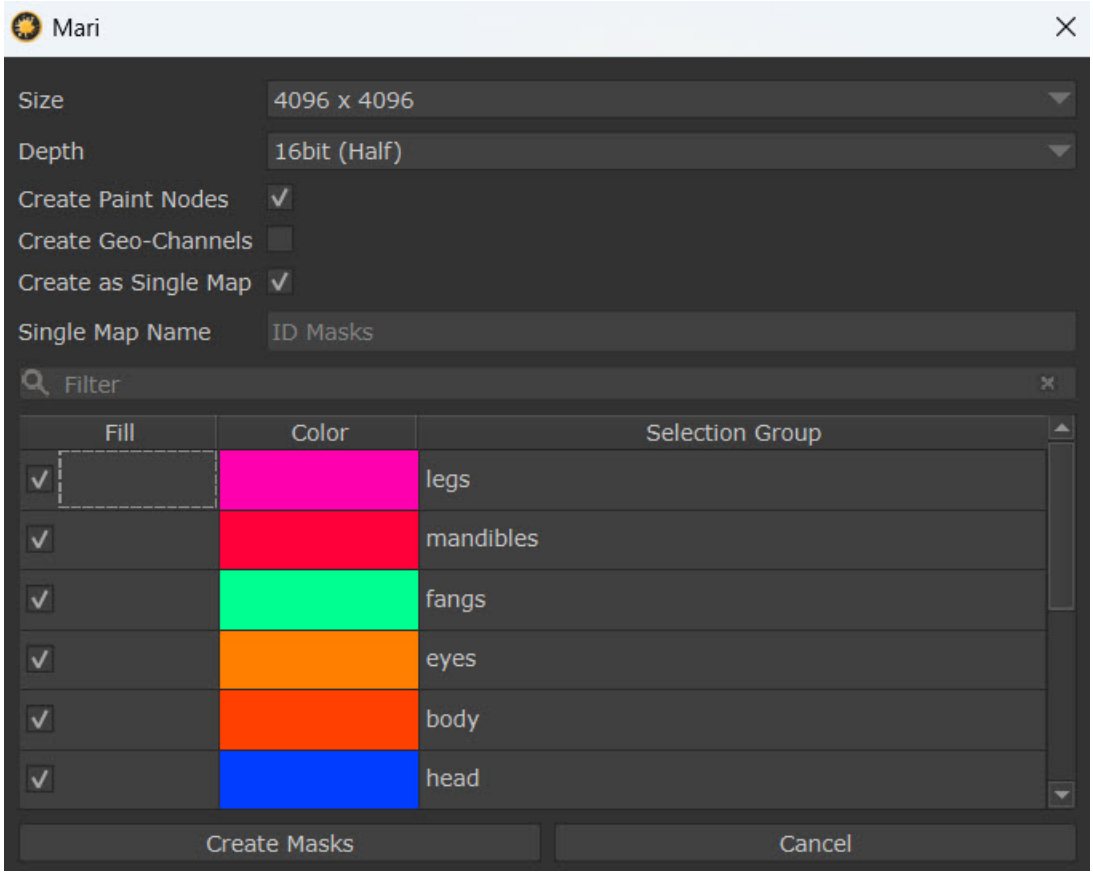
Mask from the selection group:



Now you have a mask, you can plug it into a specific effect you want to create for that selection group, or tweak the Geo-Channel and it will apply to the selection group to achieve a consistent look.

Creating ID Masks from Selection Groups

You can automatically generate ID masks from existing selection groups. This doesn't require manual polygon painting or baking, and is useful for Mari's Color Mask workflow.

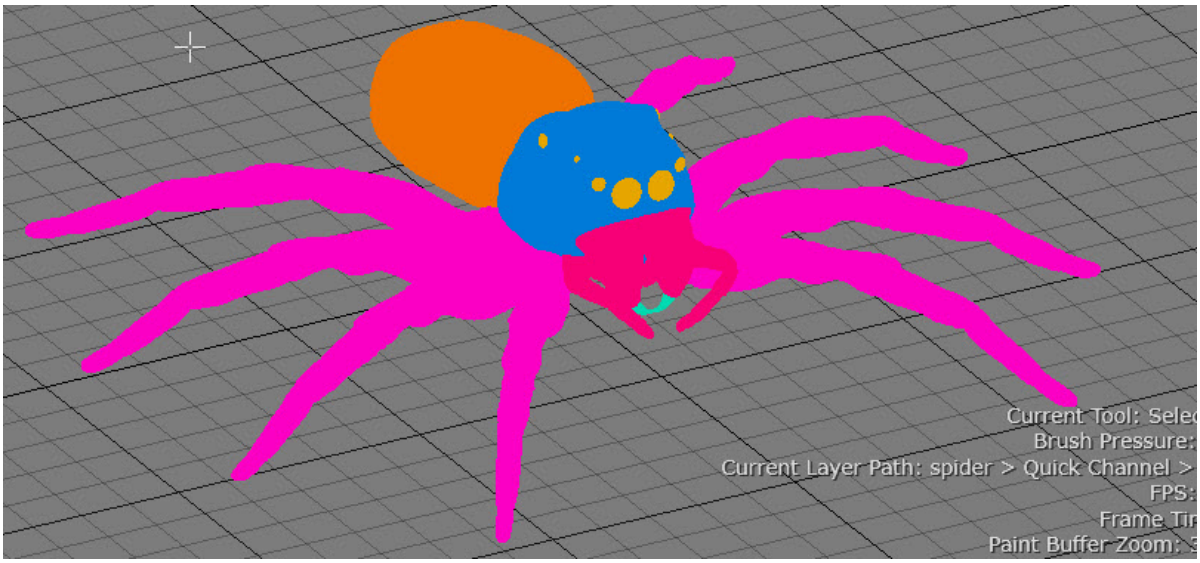


To create ID Masks for your selection group(s):

1. Right-click on the Canvas and go to **Mask > Create ID Masks from Selection Groups**.
2. Define the size and depth of the masks.
3. Define whether you want to create an associated paint node and Geo-Channel with the color masks.
4. Select whether to **Create as Single Map**. Choosing this option will mean that every selection group you choose will be grouped in one map, although each selection group will still maintain its unique color ID. You can also name the map, and that name will appear on the paint node.

   If you do not check this option, an individual color map will be created for each selection. the corresponding nodes will be named after the selection groups.
5. All of the selection groups have an associated color. You can edit this color by clicking on it, which will open up the color picker dialog. The color ID is stored with the selection group. Select the selection groups you want to bake out as color maps. You can use the search bar to find a specific selection group from a long list.
6. Click **Ok** and the paint nodes will appear in the node graph.

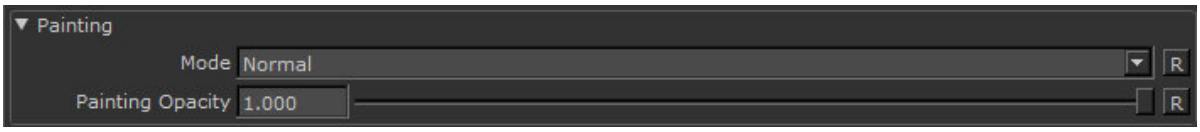
If a polygon belongs to more than one selection group, it'll be colored with the last one in the list to get rendered of that selection.



Setting the Paint Blending Mode

1. In the **Painting** palette, scroll to the **Projection Settings** > **Painting** section.

   Mari displays the details of the current paint blending mode:

   You can also set this in the **Tool Properties** toolbar.

2. Paint a stroke or two on your model (so you can see the effects of changing the settings).
3. Select a paint blending mode from the **Painting Mode** dropdown.

   The paint blending modes available are similar to many other paint packages. If you aren't familiar with the various modes, see Paint Blending Modes for descriptions and examples.

   As you change modes, the paint in the buffer updates to show the effect.

> **Tip:** Play around with the blend modes. As you change them, see how the paint in your buffer changes. Experiment with the results you can get with different blend modes, how they work with different types of brush strokes, paint alpha values, and so on.

Paint Blending Modes

Mari comes with several paint blending modes. The chart below uses the following example surface and painting to illustrate the effect of applying the different modes:

Unpainted surface.	Paint stroke in default blend mode.

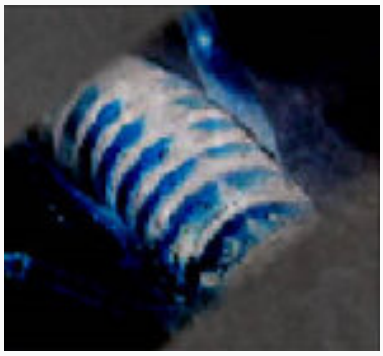
There are a number of shortcuts that can help you switch between Previous Blend, Last Blend, or Next Blend modes (among others), as well as resetting the paint blend mode. For a list of available shortcuts, see Appendix A: Shortcuts.

In the application, these effects are grouped by functional area, for example, all the light modes (hard light, soft light, and so on) are grouped together. This list is alphabetical to make it easier for you to find a description of a particular effect.

Blend mode	Description	Example
Add	Adds the values of the colors in the paint buffer to the values on the layer's surface.	
Add Normal Maps	Blends two RGB-encoded vector layers with the **Add** input by converting the raw values from the 0.0 to +1.0 colorspace-encoded range to the -1.0 to +1.0 vector range, applying the blend factor to the **Over** input, adding the values, then normalizing the values before converting it back to the 0.0 to +1.0 colorspace-encoded range.	
Burn Highlights	Burns the layer based on the highlights in the paint buffer.	

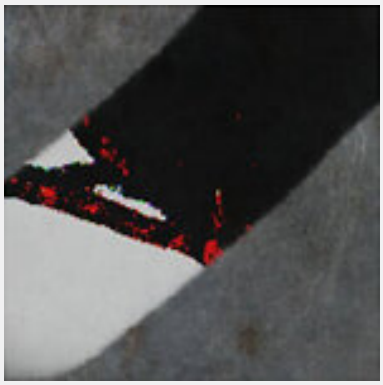
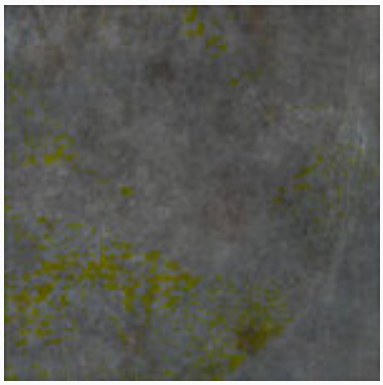
Blend mode	Description	Example
Burn Midrange	Burns the layer based on the middle range (that is, not the lightest or darkest pixels) of the paint buffer.	
Burn Shadows	Burns the layer based on the contents of the paint buffer, with darker pixels being burned harder.	
Clear	Subtracts the opacity of the paint stroke from the current surface. For example, an 80% opaque stroke leaves a surface with 20% opacity.	

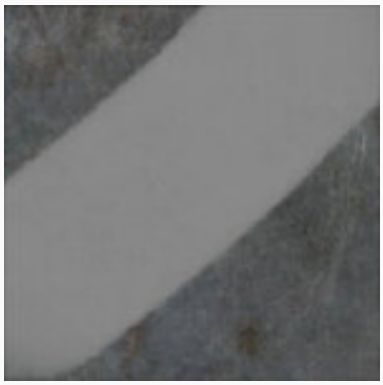
Blend mode	Description	Example
Color	Takes the luminance of the layer, and the color and saturation of the paint buffer.	
Color Burn	Burns the layer, using the contents of the paint buffer as input for the burn operation.	
Color Dodge	Dodges the layer surface, using the contents of the paint buffer to control the degree of dodging.	

Blend mode	Description	Example
Contrast	The destination color moves away from the painted color by contrasting around the painted color. How much it contrasts depends on the alpha.	
Darken	Gives each pixel the darker of two possible values: the current layer surface or the contents of the paint buffer (whichever is darker).	
Decontrast	The destination color moves towards the painted color by decontrasting around the painted color. How much is decontrasts depends on the alpha.	

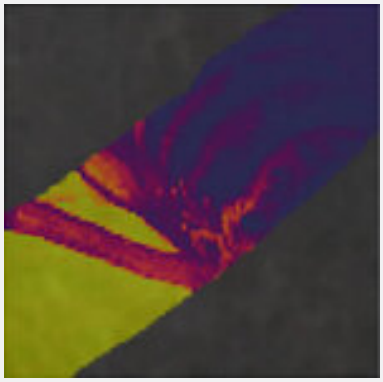
Blend mode	Description	Example
Difference	Inverts the colors. Mari subtracts the values in the paint buffer from those on the layer surface.	
Dodge Highlights	Dodges the layer based on the highlights in the paint buffer.	
Dodge Midrange	Dodges the layer based on the middle range (that is, not the lightest or darkest pixels) of the paint buffer.	

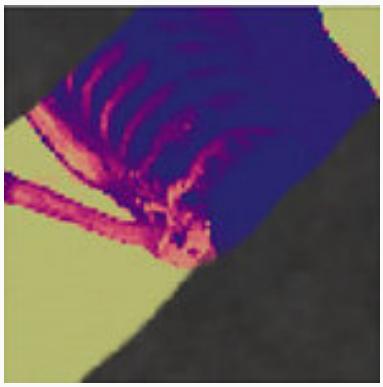
Blend mode	Description	Example
Dodge Shadows	Dodges the layer based on the contents of the paint buffer, with darker pixels being dodged harder.	
Exclusion	Inverts the colors, but with a lowered contrast.	
Hard Light	Mimics a harsh spotlight. Makes light areas lighter (using a screen effect), and dark areas darker (using a multiply effect).	

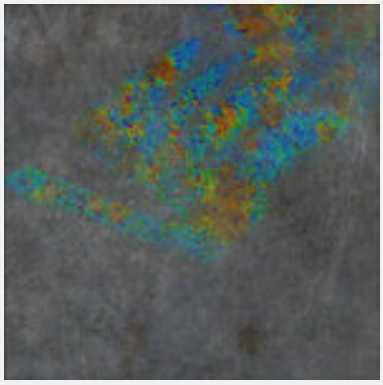
Blend mode	Description	Example
Hard Mix	First performs a vivid blend, then thresholds the colors to their extreme values. The result is that the pixels can be one of eight colors - black, white, red, green, blue, cyan, magenta, or yellow.	
Hue	Takes the luminance and saturation from the layer pixels, with the hue from the paint buffer.	
Inverse Difference	Subtracts the colors in the paint buffer from the layer surface, and then inverts the result.	

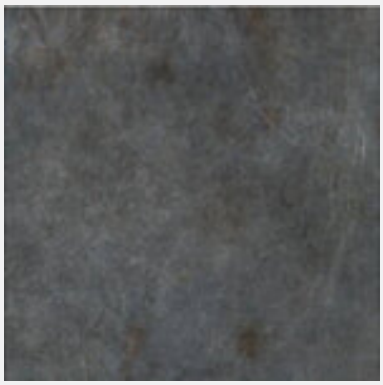
Blend mode	Description	Example
Invert	Uses the paint buffer as the input to invert the pixels on the base layer.	
Lighten	Gives each pixel the lighter of two possible values: the current layer surface or the contents of the paint buffer (whichever is lighter).	
Luminance	Takes the hue and saturation of the layer color, and the luminance of the paint buffer. (This is the opposite of the Color blend mode.)	

Blend mode	Description	Example
Mix Normal Maps	Blends two RGB-encoded vector layers with the **Mix** input by converting the raw values from the 0.0 to +1.0 colorspace-encoded range to the -1.0 to +1.0 vector range, applying the blend factor to the **Over** input, adding the values, then normalizing the values before converting the value back to the 0.0 to +1.0 colorspace-encoded range.	
Mix Reoriented Normal Maps	Blends two RGB-encoded vector layers in a way that preserves the strength and details of both the base and the overlay inputs.	
Multiply	Darkens the color. Mari multiplies the layer color by the color in the paint buffer.	

Blend mode	Description	Example
Normal	Replace values on the layer's surface with those from the paint buffer. (Mari's default mode.)	
Nudge Flow Vectors	Changes the paint buffer to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. The **Nudge Flow Vectors** mode adds the vectors together to get the blend result, and is best for use with flow maps.	
Nudge Normal Vectors	Changes the paint buffer to tangent vector space. Allows the z value to face outward in 3D space, so that vectors face out from the object's surface anywhere on the model. The **Nudge Normal Vectors** mode adds the vectors together to get the blend result, and is best for use with normal maps.	

Blend mode	Description	Example
Overlay	Uses the patterns and colors from the paint buffer, but keeps the highlights and shadows from the layer. Adjustment stacks always use an overlay blend mode.	
Paint Flow Vectors	Changes the paint buffer to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. Unlike **Nudge Flow Vectors**, the **Paint Flow Vectors** mode blends between the vectors instead of adding them. This mode is best for use with flow maps.	
Paint Normal Vectors	Changes the paint buffer to tangent vector space. Allows the z value to face outward in 3D space, so that vectors can face out from the object's surface anywhere on the object. Unlike **Nudge Normal Vectors**, the **Paint Normal Vectors** mode blends between the vectors instead of adding them. This mode is best for use with normal maps.	

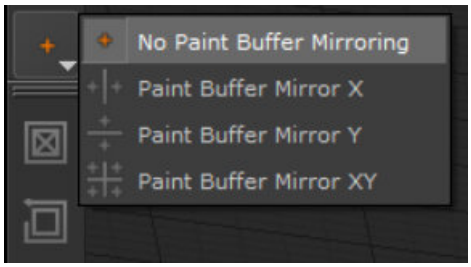
Blend mode	Description	Example
Pin Light	For light areas in the paint buffer, replaces pixels on the layer's surface that are darker than the corresponding pixels in the paint buffer. For dark areas in the paint buffer, replaces pixels on the layer's surface that are lighter than the corresponding pixels in the paint buffer. (Otherwise, leaves the layer surface unchanged.)	
Saturation	Takes the luminance and hue of the layer color, with the saturation from the paint buffer.	
Scale Vectors	This is most useful with the basic **Paint** tool, while preferably using grayscale paint. **Scale Vectors** uses the intensity of color to control the length of the normal in 3D space.	

Blend mode	Description	Example
Screen	Lightens the image. Mari divides the color on the surface of the layer by the value of the paint. (This is the opposite of Multiply.)	
Soft Light	In dark areas on the layer, burns the paint buffer onto the layer surface. In lighter areas, dodges it on.	
Sponge Desaturate	Uses the paint buffer as the input to desaturate the pixels on the base layer.	

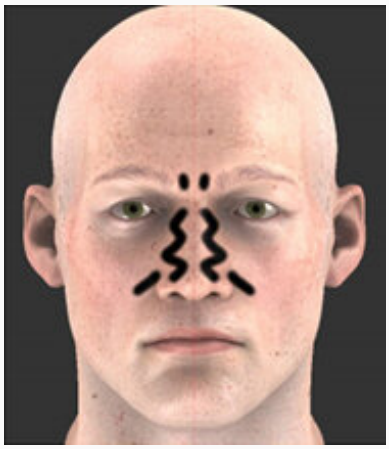
Blend mode	Description	Example
Vivid Light	In dark areas on the layer, adds the paint buffer with more contrast. In light areas, adds it with less contrast.	

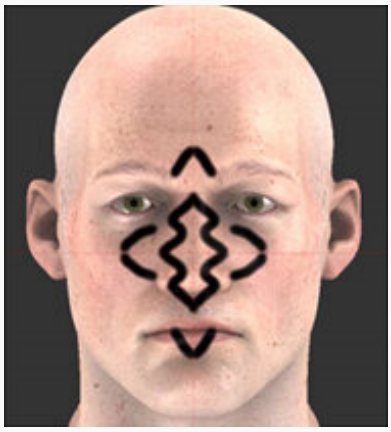
Paint Buffer Symmetry

You can mirror paint strokes in the paint buffer. On the **Project Controls** toolbar, you can access the **Mirror Painting** modes. They control whether you want your paint strokes to be mirrored on the paint buffer axis and, if so, how the paint strokes should be mirrored. If you can't see the **Project Controls** toolbar, right-click on the toolbar area and choose **Project Controls** from the dropdown menu.



The four **Mirror Painting** modes are included below, with example paint strokes illustrating the symmetry for each option:

Symmetry Option	Description	Example
	Mari's default paint buffer setting is to have paint buffer symmetry disabled.	
	Mirroring left and right divides the paint buffer vertically. Paint strokes created on either side of the buffer mirror the same strokes on the other side.	
	Mirroring top and bottom divides the paint buffer horizontally. Paint strokes created on either side of the buffer mirror the same strokes on the other side.	

Symmetry Option	Description	Example
	Mirroring four ways divides the paint buffer into quadrants. Paint strokes created in any of the four quadrants mirror in the other three.	

Painting in the paint buffer with symmetry enabled only applies to the actual paint strokes. If mirroring is used while in **Paint Through** mode or while using the **Clone Stamp** tool, only the paint strokes are mirrored; not the texture being painted through or cloned.

Paint buffer symmetry also mirrors only the paint buffer; it does not project paint onto the opposite side of the geometry.

Video: Have a quick look at the Painting and Symmetry in Mari video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Mirror Projection

You can mirror the contents of the paint buffer, including any paint strokes and painted-through textures it contains, based on a world space location. Mirror Projection allows symmetrical and asymmetrical painting as you can orientate the mirror plane about the X, Y, and Z axes. You can also use masking options to paint on only one side of the mirror plane.

Note: Mirror projection re-projects the same paint buffer from a symmetrical angle. This means for tools like Clone Stamp, Blur, and TowBrush to work with Mirror Projection, the paint on the model needs to already be symmetrical, or you may find artifacts appear on the mirrored side.

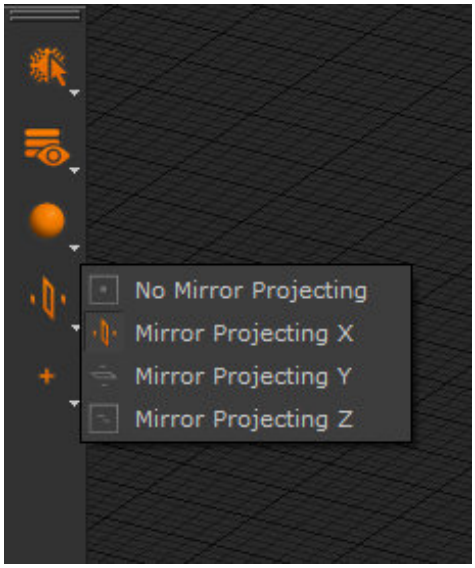
Accessing Mirror Projection Controls

You can access the Mirror Projection controls in two places: the **Project Controls** toolbar and the **Painting** palette.

Video: Watch Mirror Plane's Painting Style and Color.

Project Controls Toolbar

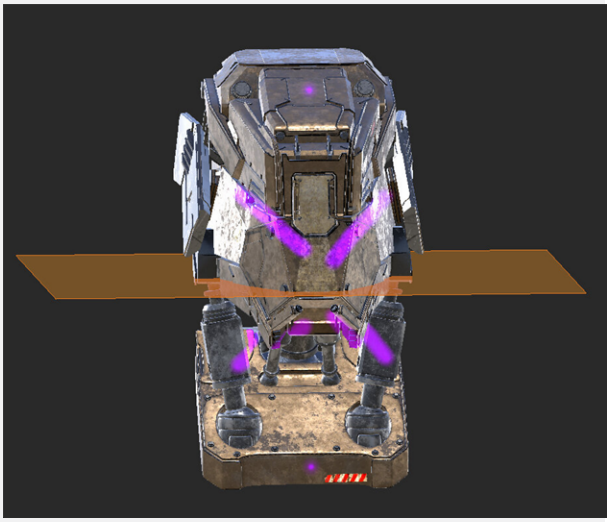
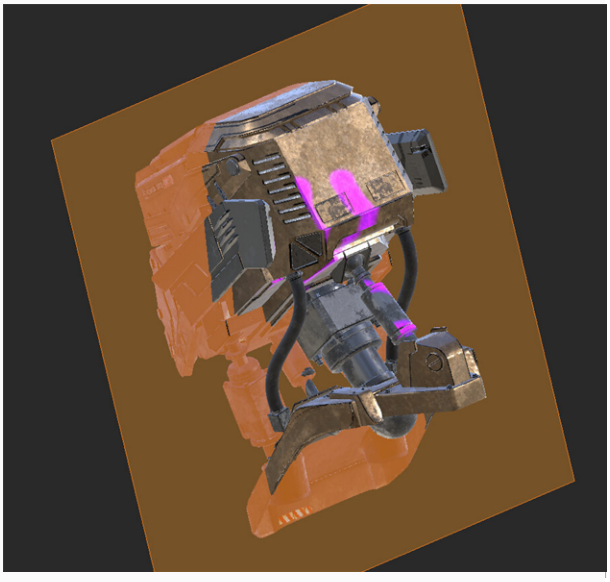
On the **Project Controls** toolbar, you can select the **Mirror Projecting** modes. These modes set the position of the mirror plane perpendicular to the selected axis. You can also disable Mirror Projection by setting it to **No Mirror Projecting**.



Note: If you can't see the **Project Controls** toolbar, right-click on the toolbar area and choose **Project Controls** from the context menu.

The **Mirror Projecting** modes are described below, with example paint strokes illustrating the symmetry or asymmetry for each option:

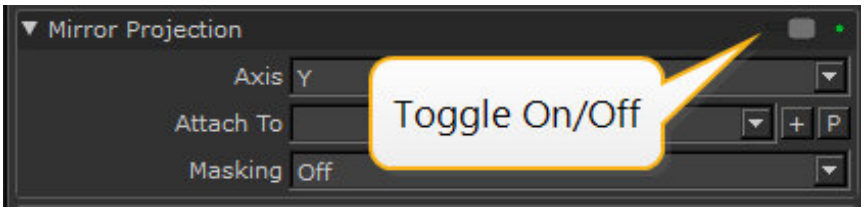
	By default, mirror projection is disabled.	
	With **Mirror Projecting X** selected, the mirror line appears vertically. Paint strokes created on either the left or right side of the mirror plane project the same strokes on the other side.	

	With **Mirror Projecting Y** selected, the mirror line appears horizontally. Paint strokes created either on top or under the mirror plane project the same strokes on the other side.	
	With **Mirror Projecting Z** selected, the mirror line appears vertically intersecting the profile of the asset. Paint strokes created either at the front or back of the mirror plane are projected on the opposite side of the geometry.	Front 

		Back

Painting palette

You can also access the Mirror Projection controls in the **Painting** palette in the **Mirror Projection** section.



In the **Mirror Projection** section, you can select one of the **Mirror Projecting** modes using the **Axis** dropdown menu. You can also select an object or create a locator to which the mirror plane is attached, using the **Attach To** dropdown menu and controls. This allows you to move and rotate the mirror plane for different mirroring effects, see Transforming the Mirror Plane. The **Masking** dropdown menu let you select the masking modes to paint on only one side of the mirror plane, see Using Masking Modes.

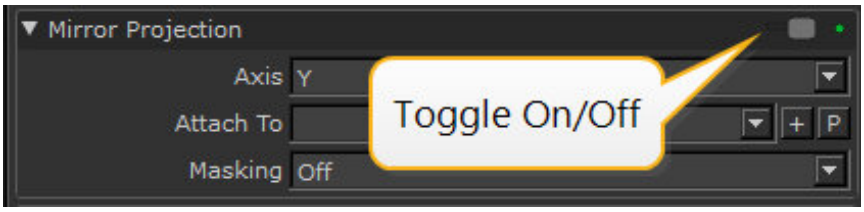
Transforming the Mirror Plane

You can transform the mirror plane using locators to align it with objects that are not orientated in line with the origin. The mirror plane attaches to an object or locator, which can then be moved or rotated through 3D space.

Video: Watch Transforming the Mirror Plane.

Attaching the Mirror Plane to a Locator

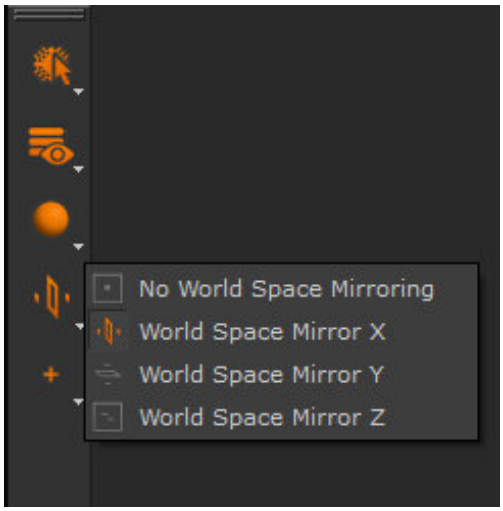
1. Navigate to the **Painting** palette > **Mirror Projection** section and toggle on **Mirror Projection**.



2. Select an **Axis** to choose between vertical (**X**), horizontal (**Y**), or vertically front facing (**Z**) painting.

   OR

From the **Project Controls** toolbar, select one of the **Mirror Projecting** controls.



3. Select a locator in the **Attach To** dropdown menu.

   OR

   Click the **+** button to create a new locator.

> **Note:** If nothing is selected in the **Attach To** dropdown, the mirror plane cannot be transformed.

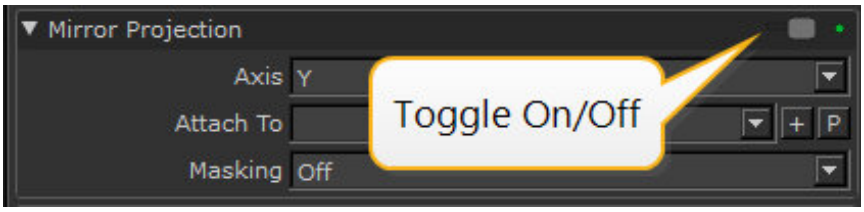
4. Ensure the newly-created locator in the **Attach To** dropdown menu is selected.
5. In the **Painting** palette > **Mirror Projection** section, click the **P** button.

   The transform handles appear on the canvas. This allows you to move and rotate the mirror plane. See Moving, Viewing, and Locking Objects for details on how to use transform handles.
6. Using the transform handles, drag an arrow or drag a specific axis ring to move and rotate the mirror plane.

> **Note:** Once you have placed the mirror plane, in the **Objects** palette, make sure to select the required object before painting.

Attaching the Mirror Plane to an Object

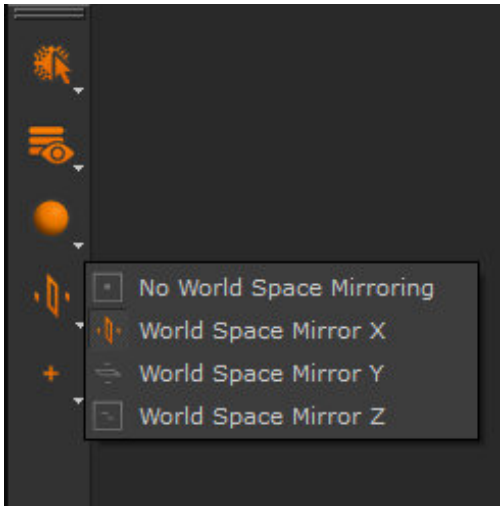
1. Navigate to the **Painting** palette > **Mirror Projection** section and toggle on **Mirror Projection**.



2. Select an **Axis** to choose between vertical (**X**), horizontal (**Y**), or vertically front facing (**Z**) painting.

   OR

   From the **Project Controls** toolbar, select one of the **Mirror Projecting** controls.



3. Select an object in the **Attach To** dropdown menu.

> **Note:** If nothing is selected in the **Attach To** dropdown, the mirror plane cannot be transformed.

4. In the **Painting** palette > **Mirror Projection** section, click the **P** button.

   The transform handles appear on the canvas. This allows you to move and rotate the object along with the mirror plane. See Moving, Viewing, and Locking Objects for details on how to use transform handles.

5. Using the transform handles, drag an arrow or drag a specific axis ring to move and rotate the object along with the mirror plane.

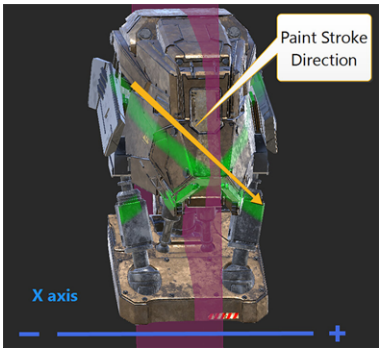
Using Masking Modes

> **Video:** Watch Masking Modes.

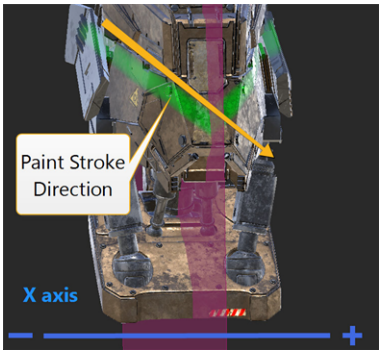
The Mirror Projection masking modes let you paint on only one side of the mirror plane so that your paint strokes can't cross over the mirror plane line, giving your paint a reflected edge result. You can choose between the following modes:

- **Off** - This disables the Mirror Projection masking.
- **Dynamic** - This masks the side (positive or negative) of the mirror plane that is on the opposite side to the current canvas view.
- **Positive** - This masks the positive side of the current mirror axis, X, Y, or Z, allowing you to paint on the negative side only.
- **Negative** - This masks the negative side of the current mirror axis, X, Y, or Z, allowing you to paint on the positive side only.

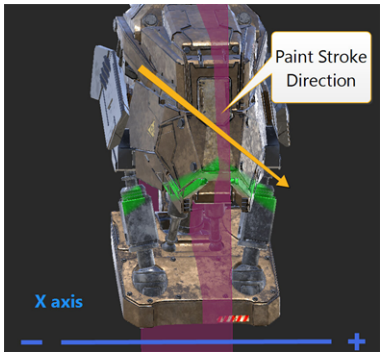
For instance, in the example below, the **Mirror Projecting** mode is set to **Mirror Projecting Y**, and a paint stroke is drawn diagonally from top to bottom.



Masking set to **Off**



Masking set to **Positive**



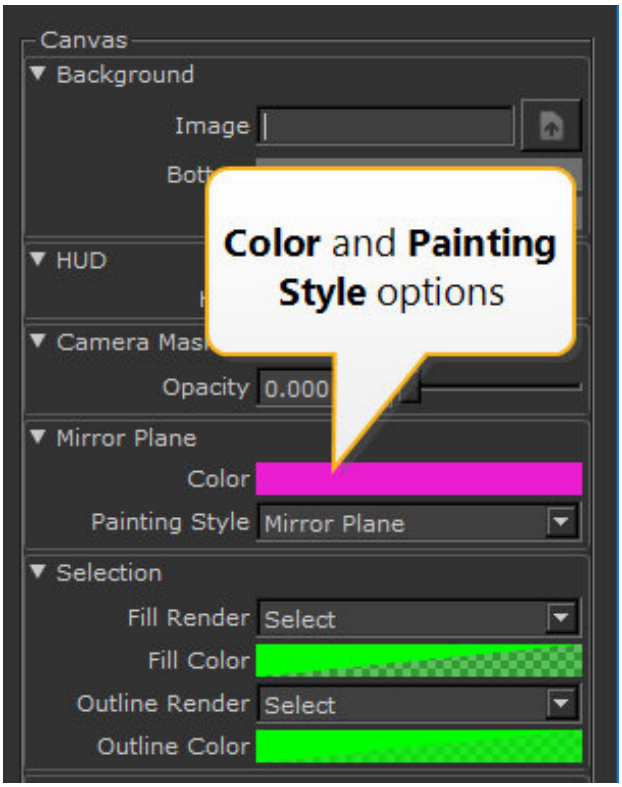
Masking set to **Negative**

Changing the Mirror Plane's Painting Style and Color

To change the color and the painting style of the mirror plane:

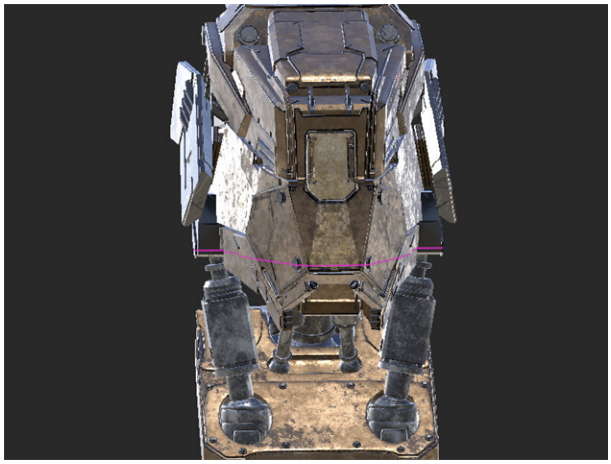
1. Right-click on the canvas and select **Display Properties**.

   This opens the **Display Properties** dialog.
2. In the **Mirror Plane** section, select the **Color**, and for the **Painting Style** select either **Mirror Plane** or **Mirror Line** in the dropdown menu.



3. Click **OK** to apply changes.

Painting Style set to **Mirror Plane**



Painting Style set to **Mirror Line**

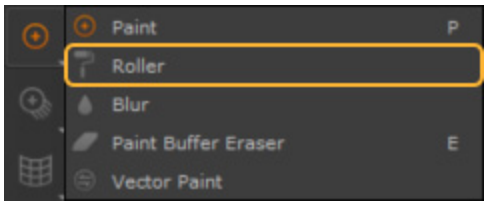
Note: The **Painting Style** is used while painting whether it is set to **Mirror Plane** or **Mirror Line**, however for better visibility the **Mirror Plane** option is always used when transforming the mirror plane. See Transforming the Mirror Plane.

Paint Continuous, Repeating Textures with the Roller Brush

The Roller Brush tool adds a new painting mode that paints a tileable image following the direction of your brush stroke. For example, you can create seams or decals on a character's clothing without having to manually line up each tileable texture, one by one, to mimic a brush stroke.

To paint with the Roller Brush:

In the left-hand toolbar, click and hold (the **Paint** tool). In the drop-down menu that appears, click **Roller**.



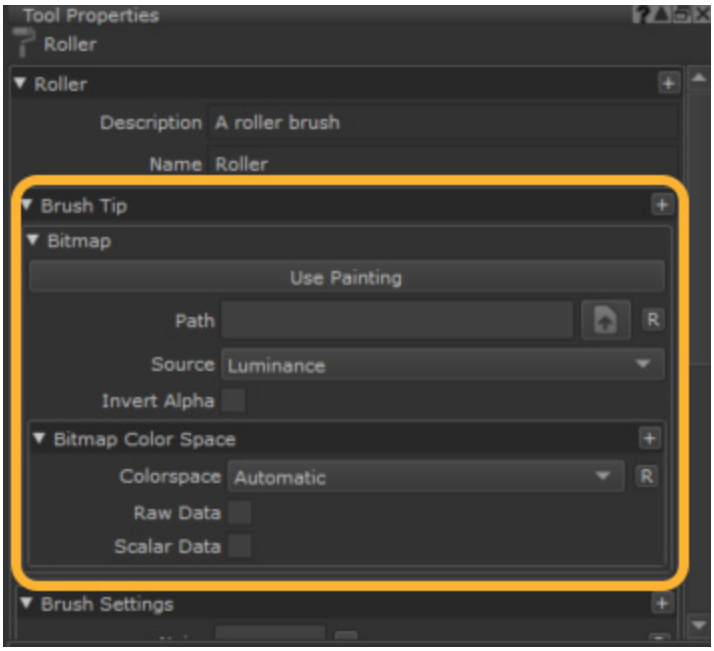
The Roller Brush tool is now available to use.

Adding Images to the Roller Brush

The Roller Brush tool can be used in conjunction with tileable images, allowing you to create seamless details with the stroke of a brush.

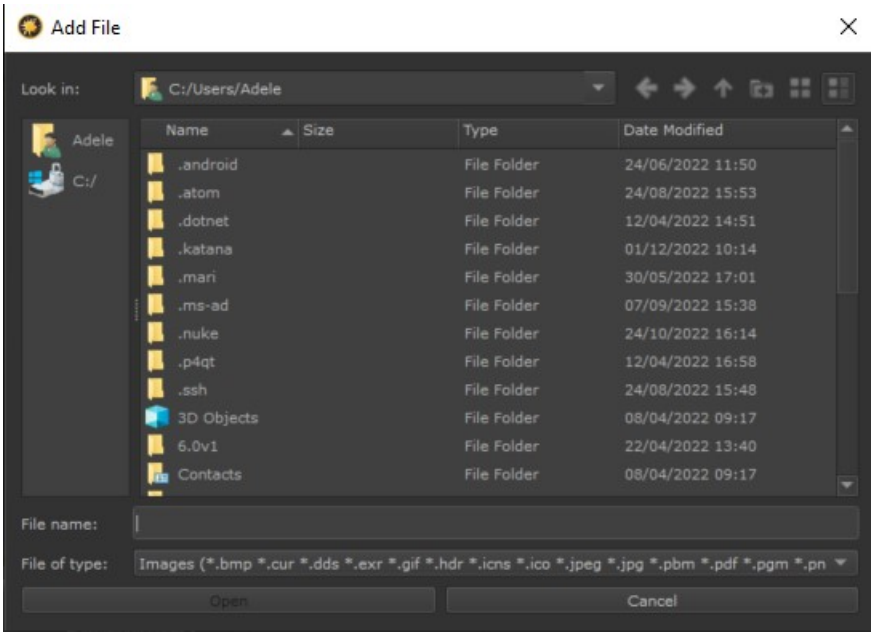
To apply an image to the Roller Brush:

1. Activate the Roller Brush tool.
2. In the Tool Properties window found in the Brush Pallet, expand the **Brush Tip** menu.



3. Click the button next to **Path** field.

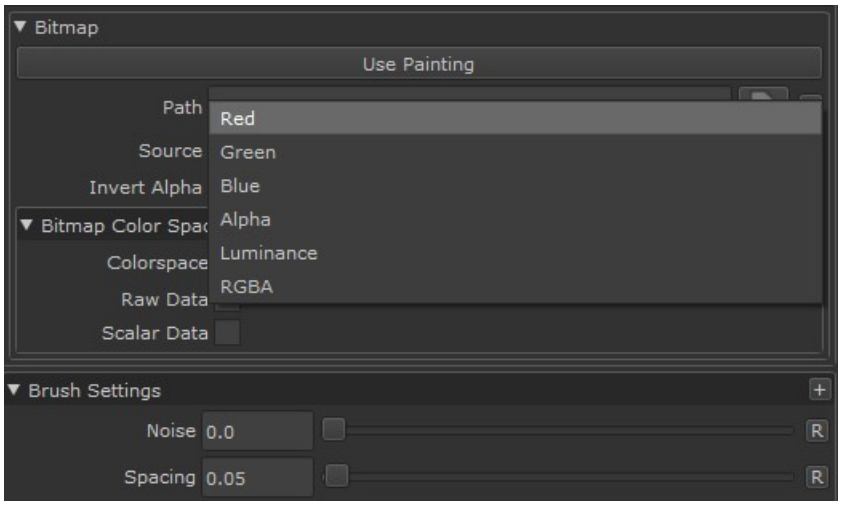
   An explorer window opens.



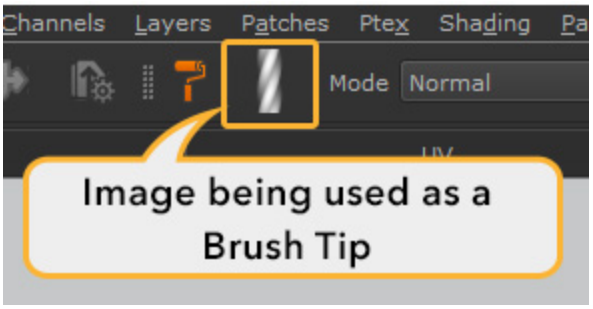
4. In the explorer window, select the image you want to use to drive the Roller Brush and click **Open**.
5. Depending on how you are planning to use the image assigned to the Roller Brush, you may need to adjust the **Source** parameter.
   - If using an RGB image, you can choose the image's **Red**, **Green** or **Blue** channels, or **RGBA** to use all three channels plus an Alpha channel.

- If your image has an Alpha channel you would like to use, choose **Alpha**.
- If using a black and white image that you would like to use as an Alpha, choose **Luminance**.
- When **Invert Alpha** is enabled, any alpha in the brush **Source** is inverted.

> **Note:** If you are using the Roller Brush tool for Multi-Paint, set up your images or colors in the Multi-Channel Paint node and activate the Roller Brush tool. To roll all images simultaneously, navigate to the **Tool Properties** and set **Brush tip > Source** from **Alpha** to **RGBA**. For more information go to Using the Roller Tool to Multi-Paint the Patches Example Project.

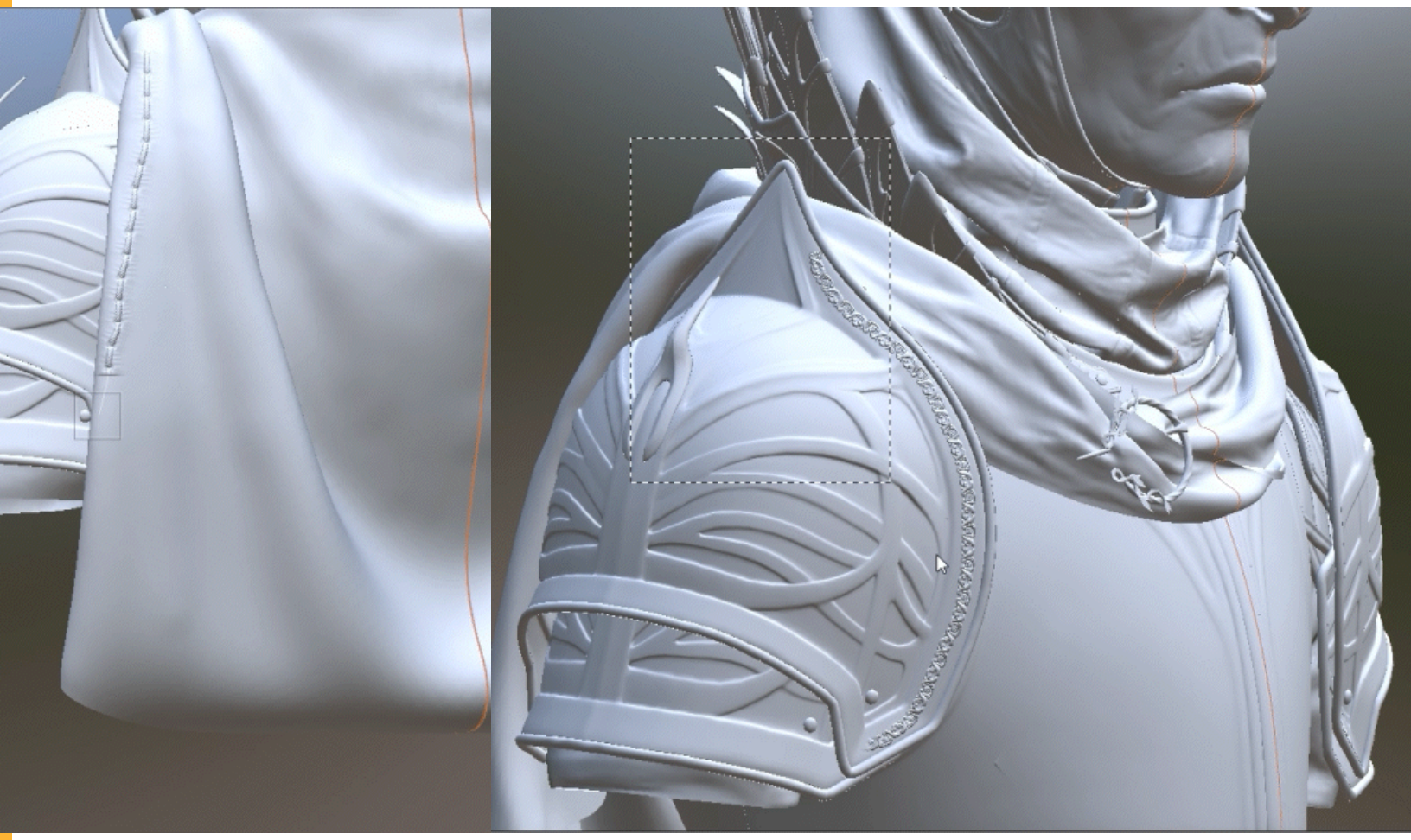


6. Once a brush tip is loaded and the source is set, the brush is visible in the top navigation bar, next to the roller brush icon.



Once an image is assigned to the Roller Brush tool, the Roller Brush can be used in the same way as Mari's default brush. Strokes made by the Roller brush can be manipulated on the paint buffer in the same ways a regular brush stroke would be, and can be baked onto your asset or cleared from the Paint Buffer using the same default commands. For more information on the Paint Buffer, see Paint Buffer.

Tip: You can also drag and drop an image from the Shelf or Image Manager onto the brush tip **Path** field, rather than manually browsing for the file.



Roller Brush tool being used with Mirror painting to create even stitches on both sides of the asset.

Curve drawn with the Roller Brush tool being manipulated before baking on the paint buffer with the Warp tool.

Note: When using a brush tip driven by an image, the brush tip will inherit whatever your current foreground color is. To ensure that alpha images and RGBA images are displayed properly, set your foreground color to white.

Tip: You can use the **Trim** option to remove any excess transparent borders on the image that has been applied as the Roller Brush tiling image. This will trim top/bottom in Horizontal tiling mode and left/right in Vertical tiling mode.

Roller Brush Settings

There are a number of settings within the Roller Brush **Tool Properties** that allow you to customize the behavior of the Roller Brush tool. Mari's Brush Engine dynamics can also be utilized with the Roller Brush, allowing you to create stylized strokes while you paint. For more information on Mari's dynamic brush engine, see Changing Brush Color Dynamically.

The following parameters are specific to the Roller Brush:

Repeat Direction

Repeat Direction configures Mari to tile your image brush tip **Horizontal** or **Vertical**, as well as **Vertical Mirrored** and **Horizontal Mirrored**.

Vertical tiles your image vertically along your brush stroke.

Horizontal tiles your image horizontally along your brush stroke.

Vertical Mirrored tiles your image vertically along your brush stroke, while also mirroring every second tip vertically.

Horizontal Mirrored tiles your image horizontally along your brush stroke while also mirroring every second tip horizontally.

Transforming and Warping Paint

Mari transforms and distorts paint within the buffer with these four tools:

- **Transform Paint Buffer**
- **Warp**
- **Slerp**
- **Pinup**.

Tip: See Controlling the Way Mari Applies Paint to Your Model for details of how the paint buffer works.

Video: To learn more about how to transform paint once it reaches the paint buffer, watch the Transforming Paint in the Paint Buffer video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Moving or Transforming Paint Buffer

You can use the **Transform Paint Buffer** tool to resize and stretch the paint buffer as a whole. This lets you:

- resize the buffer (either proportionally or stretched),
- rotate it,
- or just move it around.

Video: Watch a quick demo to learn how to transform the paint buffer.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

As you transform the paint buffer, its contents - that is, the paint as it appears on your model - changes too. For example, you can paint a stroke, then stretch it, rotate it, and position it exactly on your model, before finally baking it onto the layer.

To transform the paint buffer:

1. Start Mari and open your project.
2. Paint a couple of example strokes onto your model, so you can see the effects as you work.

3. In the **Tools** toolbar, click to select the **Transform Paint Buffer** tool or press **M**.
4. To transform the paint, you can do the following:

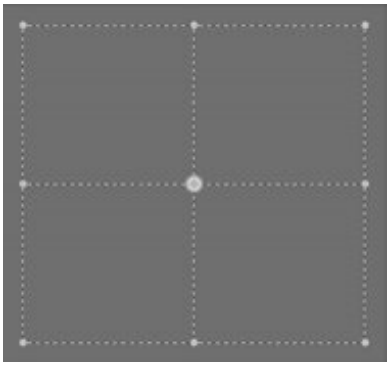
To...	You need to...	Example:
Move the paint buffer around	Click on the buffer and drag it.	
Resize the paint buffer, preserving the scale	Hold down **Ctrl**/**Cmd**+**Shift**, click anywhere on the canvas and drag. Dragging to the left makes the buffer smaller, to the right makes it bigger.	
Stretch the buffer (resize without preserving scale)	Click on a side or corner of the buffer and drag it.	
Rotate the buffer	Either: • click outside the buffer and drag, or • hold down **Ctrl**/**Cmd**, click inside the buffer, and drag.	

Tip: You can also make these transformations by manually editing the values under **Transform** in the **Paint Buffer** section of the **Painting** palette. For example, editing the **Translation** values moves the buffer around on screen, changing the **Scale** resizes the buffer, and the **Rotation** control sets the rotation to specific degrees.

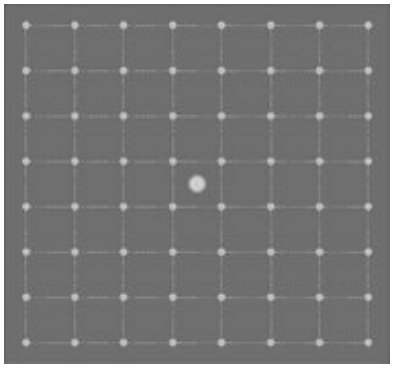
Experiment: Experiment with the various transformations you can make to the paint buffer. Try stretching, rotating, and moving the buffer, then baking the paint onto the surface of your model. Use both the **Transform Paint** tool and the controls under **Paint Buffer** in the **Painting** palette.

Warping Paint Using a Grid

The **Warp** tool lets you warp a selected area within your paint buffer. You create a **warp box** - a rectangle with between 4 and 64 **control points**.

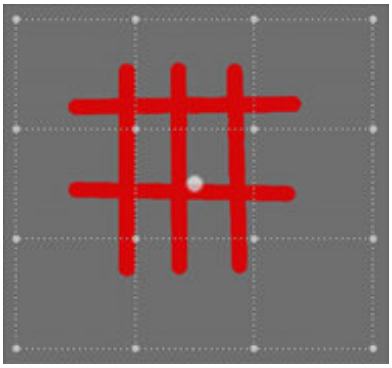


Warp box with four control points.

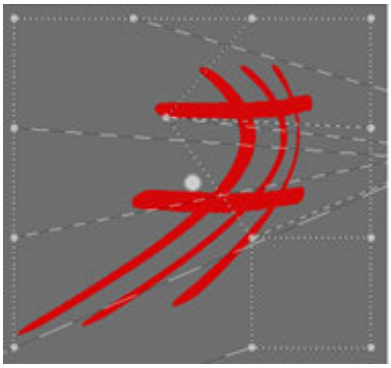


Warp box with 64 control points.

You can then warp the paint buffer under the box by moving the control points around.



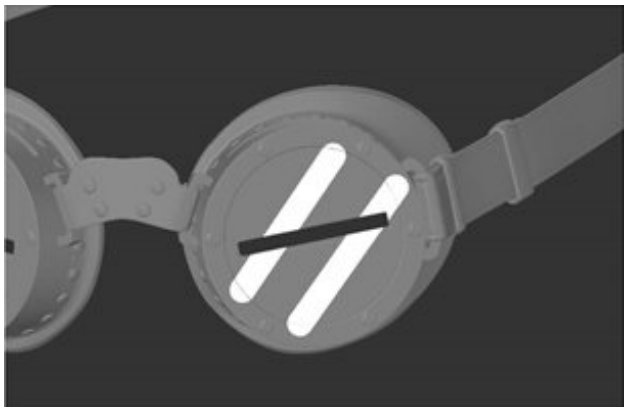
Initial transform box
(16 control points).



After some warping.

To use the **Warp** tool:

1. If necessary, paint another couple of strokes on the model to experiment with.

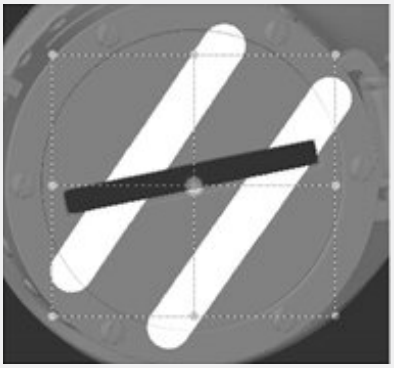


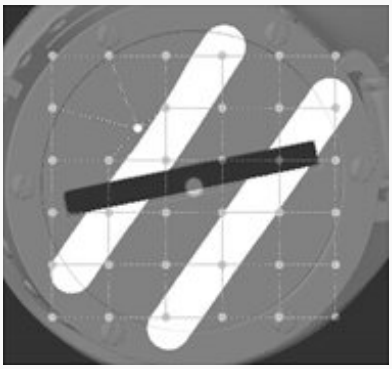
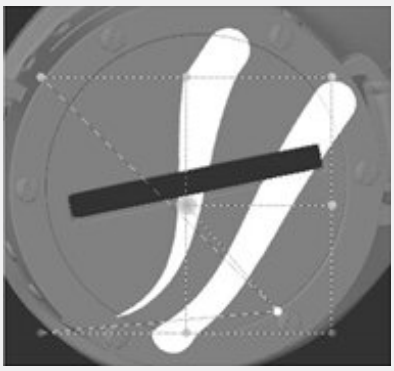
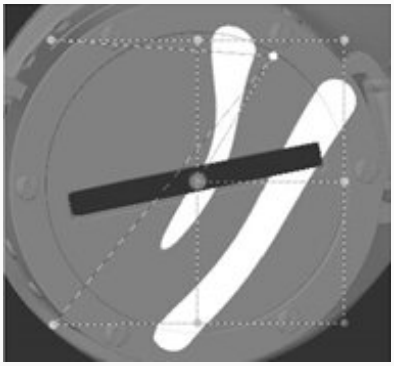
2. Click to select the **Warp** tool .
3. Click and drag to create a rectangle to warp.

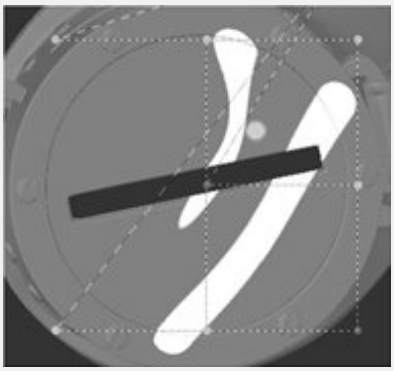
   The warp rectangle displays, with nine control points, and a central rotation pivot (the large point in the center).

> **Tip:** After you draw a warp box, to replace it with another warp box: hold down **Shift**, click and drag.

4. To warp your selection, you can do the following.

To...	You need to...	Example:
Increase or decrease the number of control points (up to 64)	press the **Up** or **Down** arrow keys (↑ ↓).	

To...	You need to...	Example:
Move a single control point	click and drag it.	
Move multiple control points	click and drag out a rectangle covering the control points, then click one of the points and drag it - the other points follow.	
Rotate around the central pivot	hold down **Shift**+**R**, then drag one or more points.	

To...	You need to...	Example:
Move the central pivot point	click and drag it. Be careful to click on the pivot point (the big circle), rather than the central control point.	
Remove the current warp	press ' (apostrophe).	

5. To complete the warp, do one of the following:
   - draw a new warp box (remember to hold down **Shift** when dragging to create the new box),
   - change to another tool, or
   - bake the paint.

Tip: You cannot undo individual stages of a warp. You can only undo the warp as a whole.

Experiment: Have a play with the warp tool. Try:
- moving one point,
- clicking and dragging to select two points, then moving them together,
- using the **Shift**+ **R** to rotate points.

Pulling, Smudging, Growing, and Shrinking Paint

The **Slerp** tool works like the Liquify tool in Photoshop®. It lets you "pull" the contents of the paint buffer around.

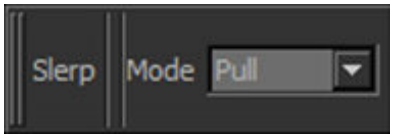
Initial contents of paint buffer.	Body pushed in to give a narrow waist.	Slerped using a big brush tip in an S motion.

Slerp uses a standard Mari brush tip, so you can fine-tune how to push things around. As well as pulling, Slerp lets you grow, shrink, and rotate the paint. You can also use it to selectively erase distortions you've already applied.

To use the **Slerp** tool:

1. The effects of the **Slerp** tool can be quite subtle, so we recommend you use a sample picture to see them clearly. Use the **Paint Through** tool to drag a picture over the model, then stamp it down with the **'** (apostrophe) key.

2. Click to select the **Slerp** tool .
3. If it's not visible, show the **Tool Properties** toolbar. You can select the tool mode from the **Mode** field:



4. Select from the **Slerp mode selector** to use Slerp as follows:

To...	You need to...	Example:
pull paint around	select the (default) **Pull** mode, click and drag.	
make the paint in a specific area larger	select the **Grow** mode, click and drag.	

To...	You need to...	Example:
make the paint in a specific area smaller	select the **Shrink** mode, click and drag.	
rotate your paint around	select the **Rotate** mode, click and drag.	
undo the distortion in a specific area of the painting	select the **Eraser** mode, click and drag.	n/a
undo all the distortions you've applied	press **'** (apostrophe).	n/a
apply your changes to the paint buffer (that is, make them so you can't erase the distortions)	press **;** (semicolon).	n/a

5. To complete the distortion, do one of the following:
   - press **;** (semicolon),
   - change to another tool, or
   - bake the paint.

Tip: As with the **Warp** tool, you cannot undo individual stages of a Slerp. You can only undo the Slerp as a whole. Until you have completed the Slerp, using Undo actually undoes the last completed action.

Experiment: Try using **Slerp**'s various modes. Each mode has radius and opacity options that you can change. Try adjusting these to see how dramatically that mode can change the texture.

Distorting the Paint with Pins

The **Pinup** tool works like the **Warp** tool, but lets you set your own control points (pins) wherever you want. Think of it as turning the paint buffer into a rubber sheet, which you can pull around with the pins. You can use pins either to pull the paint around, or to hold it in place. So if you get a distortion that you're happy with in part of the buffer, you can set some pins around it and other distortions won't affect it.

You can move the pins around (either one by one or in groups), or rotate them around a central point. You can also move the point of rotation.

Initial contents of paint buffer, with pins set. Point of rotation is the big white circle.

Pins on the legs moved to the right, distorting the buffer but not around the head.

The same pins rotated to the right instead of moved, stretching the buffer up.

You can also change the “strength” of each pin. This sets how much the pin affects the texture around itself.

Tip: The **Transform** tool works on the entire paint buffer at once, while the **Warp**, **Slerp**, and **Pinup** tools let you select which area within the buffer to work on.

To use the **Pinup** tool:

1. Clear the paint buffer, and use the **Paint Through** tool to stamp another example texture onto your model.
2. Select the **Pinup** tool .
3. Use the **Pinup** tool as follows:

To...	You need to...	Example:
add or remove a pin	**Shift**-click.	
select multiple pins	click and drag around the pins.	n/a
move pins around (distorting the paint)	click on one of the selected pins and drag.	

To...	You need to...	Example:
rotate pins around the central point of rotation	click on one of the selected pins, hold down **Shift**+**R** and drag.	
move the central point of rotation	click on the point of rotation (big dot) and drag it.	n/a
change the strength of a pin	click on the pin to select it, then press the **Up** or **Down** arrow keys. As you increase the strength of a pin, the preview circle around the pin gets bigger (or smaller as you lower the strength).	n/a
remove all the current pins	press **Ctrl**/**Cmd**+**'** (apostrophe).	n/a
undo all the distortions you've applied, moving the pins back to their original positions	press **'** (apostrophe).	n/a
apply your changes to the paint buffer and remove all current pins	press **;** (semicolon).	n/a

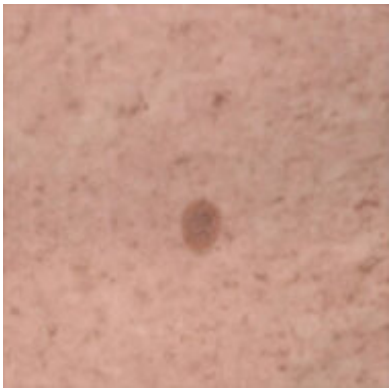
4. To complete the distortion, do one of the following:
   - press **;** (semicolon),
   - change to another tool, or
   - bake the paint.

As with the **Warp** tool, you cannot undo individual stages of a **Pinup**. You can only undo the **Pinup** as a whole. Until you have completed the **Pinup**, using **Undo** actually undoes the last completed action.
Using pins is also hard to describe but quite easy to get the hang of. Try doing the following:
• Create some pins.
• Drag one pin around.
• Select multiple pins and drag them around.
• Rotate pins.
• Move the point of rotation, and rotate pins around the new point of rotation.
• Increase and decrease the strength of individual pins. See how stronger pins affect the texture more.

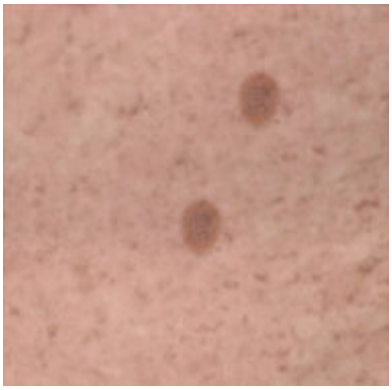
Towing Paint Around

The **Towbrush** is similar to the Healing Brush and Patch tools in Photoshop®. It lets you select areas of paint and drag them around the surface of your model. As you tow the paint around, it blends the edges of your selection with the surrounding paint. It works like the **Clone** tool, but blends in the edges of the cloned selection.

Put simply, it lets you grab paint and move it around on the surface, automatically blending in the edges as you go.



Original surface.



Mole cloned and moved, broken veins moved.

The **Towbrush** has three modes, allowing you to:

• **Clone** - clone from the selected region to another part of the surface.

• **Heal** - overwrite the selected region with another part of the surface.

• **Move Feature** - swap the contents of the selected region with another area on the surface.

There are limits to the amount of blending the **Towbrush** can do: as with the Healing Brush in Photoshop®, it's not magic. But it's very good at working with surfaces with regular textures, such as skin, rock, stone, and so on.

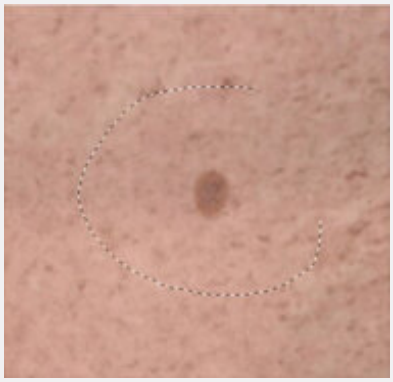
Video: To learn how to use Mari's Towbrush on paint that has already been baked, watch Using the Towbrush.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Selecting and Towing Paint

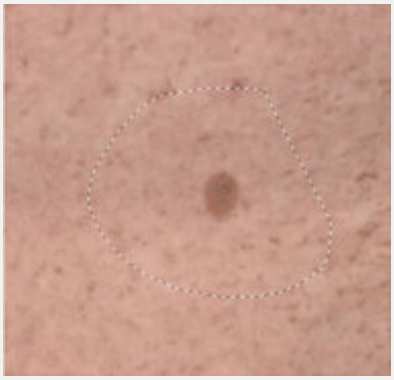
1. Start Mari and open your project.
2. If you have any unbaked paint on your model, either bake it or clear the buffer.
   The Towbrush only works on baked paint.
3. Click to select the **Towbrush** tool.
4. If it's not visible, show the **Tool Properties** toolbar. You can select the tool by clicking the icons:
   Select the mode from:

	Clone – clones from the selected region to another part of the surface. *This is the default mode.*
	Heal – overwrites the selected region with another part of the surface. That is, it clones the texture under your mouse over the selected area.
	Move Feature – swaps the contents of the selected region with another area on the surface.

   OR
   With the **Towbrush** tool selected, in the **Tool Properties** palette, select a mode from the **Mode** dropdown.
5. Click and drag on your model to select the area to tow.



You can drag any shaped area.



When you release the mouse, the selection area completes.

6. Click on the selection area and drag it around.

   As you tow the selection, the edges blend in with the surrounding texture.

7. You can adjust the settings for the **Towbrush**. You can set the:
   - **Falloff** - controls how the texture blends in from the edges of the patch. Higher values have sharper transition to the surrounding texture, lower values blend in more smoothly. Use **{** and **}** to lower and raise the falloff.
   - **Value** - how much of the color of the original patch to preserve. Higher values keep more of the original color, lower values blend the whole patch in with the background as you move it. Use **[** and **]** to lower and raise the value.

8. To fix any mistakes:
   - If you want to clear your current selection, press **'** (apostrophe).
   - If you want to undo a clone/move, just clear the paint buffer.

9. Bake your changes.

   If you want to paint multiple strokes with the **Towbrush** tool, you must bake between each move.

> **Experiment:** As the Towbrush blends its edges dynamically, it's hard to see unless you actually have a go with it. Experiment with the brush and see what results you can get on a variety of textures.

Copying and Pasting Paint

When you need to copy and paste paint in your Mari project, you must copy from a source patch, and paste onto a target patch.

You can copy and paste both single or multiple patches and in both cases, you can copy between channels. When copying multiple patches, you can copy to other objects in your project.

Copying and Pasting Single Patches

Mari's quick copy function lets you copy a single patch and then copy it back onto the same object (on any channel). You can paste the copied patch back onto multiple target patches.

1. Select your source patch from the **Patches** palette.
   This only works with single patches. You cannot copy multiple patches at once.
2. From the **Patches** menu, select **QuickCopy**.
3. Select your destination patch(es) from the **Patches** palette.
   This can be on the same layer or a different layer on the same object.
4. From the **Patches** menu, select **QuickPaste**.
   If you selected multiple patches, you may see a warning. If you choose to go ahead, the source patch is pasted across each of the target patches.
   Mari pastes a copy of the source patch onto the destination.
   Depending on how different the shapes of the two patches are, you can probably expect to see some blank areas or artifacts at the edges of the patch.

Tip: You can also find these options on the right-click dropdown menu on the canvas.

Video: Watch this quick demo on how to copy and paste between patches.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Copying and Pasting Multiple Patches

Mari's Copy Multiple Textures function lets you copy entire patches between places in your project. You can copy patches:

- between objects,
- between layers on the same object, or
- between patches on the same layer.

You can only copy textures on a patch-to-patch basis – you cannot copy one patch onto more than one other patch at once.

You can also use an offset. This is added to the source patch number to determine where to paste the texture to. For example, if you copied the texture on patch 1001 with an offset of 20, the texture would copy to patch 1021.

So, for example, if your model has a consistent numbering scheme, you can use this to paste from one side of the model to another. As you do this, Mari reverses the textures.

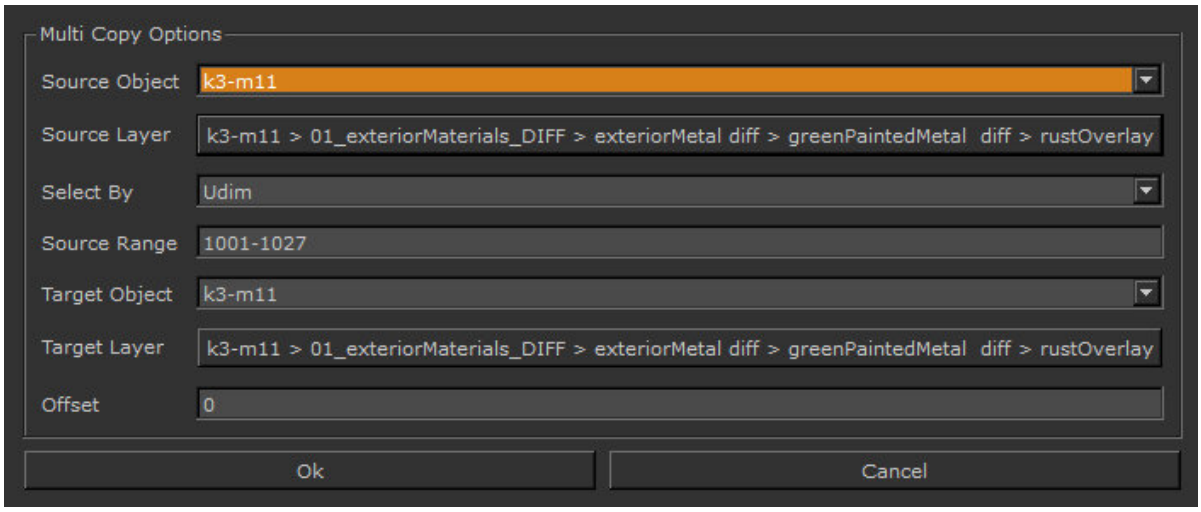
Here, we paint a couple of black stripes on the face, and then select patch 1002:

Black stripes painted.	Patch 1002 selected.

If we copy patch 1002 back onto the same channel, with an offset of 20, the paint copies to patch 1022 (the corresponding patch on the other side of the model's face). Mari reverses the texture as it pastes.

To copy and paste multiple patches:

1. From the **Patches** menu, select **CopyTextures**.
   The **CopyMultipleTextures** dialog box displays:



2. Set the details of where to take the textures from:

   - **SourceObject** and **SourceLayer** set the location of the source patches.
   - Set the **SelectBy** field to the patch numbering scheme you want to use to select the patches to copy.
   - In the **SourceRange** field, enter the numbers for the patches to copy. You can separate patch numbers with commas, or use a hyphen to indicate a range. For example, this:

     1001, 1003, 1005-1007

     selects patches 1001, 1003, 1005, 1006, and 1007.
3. Set the details of where to copy the textures to in the **Target Object** and **TargetLayer** fields. Both these fields can be the same as the source versions.
4. If required, set an **Offset**. This is added to the patch numbers when copying them. For example:

   - With an offset of 0, patch 1001 is copied to patch 1001 in the target layer.
   - With an offset of 20, patch 1001 is copied to patch 1021 in the target layer (which can be the same as the source layer).
5. Click **OK** to copy the textures.
   Mari copies the textures to the designated object, layer, and patches.

Editing Paint on Layers

When you paint in Mari, you paint into the paint buffer - which floats above the surface of the model. Only when you bake is the paint actually applied to the model's surface.

Mari's **Blur** tool lets you edit the baked surface of a layer, while the **Clone Stamp** tool includes a mode to clone paint directly up off the surface of the layer into the paint buffer.

Blurring Paint on a Layer

The **Blur** tool lets you apply a blur effect directly to the surface of your channel. You can use the **Blur** tool the same way as any other Mari brush, to apply a normal blur effect to any part of the surface.

Texture before blurring.

After blurring.

Tip: If you want to blur large areas of the surface, you may want to use the blur **filters** instead. These apply a controlled blur to entire patches. Mari includes several blur filters, including a controllable Gaussian blur. For details, see Filtering Images and Paint.

To use the **Blur** tool on paint:

1. Start Mari and open your project.
2. Select a layer with some paint on it.
3. Select the **Blur** tool .
4. Click and drag to blur your paint.

Pulling Paint into the Buffer

The **Clone Stamp** tool has a mode that clones the surface of the current layer up into the paint buffer directly above the surface. This lets you grab the baked paint and make changes.

See Clone Stamp for details on using the **Clone Stamp** tool to clone the layer surface into the paint buffer.

Clone Stamping

The clone tools let you copy paint from one image or part of a channel to the surface of your model. This is a standard feature in painting packages such as Adobe® Photoshop®, GIMP, and so on.

You start clone stamping by setting a source point. This is the point from around which Mari takes the cloned texture. As you paint, Mari duplicates the area around the origin point you've set.

Mari's **Clone Stamp** tool lets you clone from:

- the model's surface (baked paint) in the active channel, or in a different channel
- the current (unbaked) paint in the paint buffer
- an external image (selected from the **Image Manager**, optionally zoomed in or out).

When cloning from the baked paint on a channel, you can select to clone directly from the surface. In this mode, the tool clones the paint from the channel surface straight up into the paint buffer directly above that point. This is basically a way of copying the model's surface into the paint buffer so you can edit it and then re-bake the paint.

Tip: When you clone from a source on the model (any mode other than using an external image), the source point you set is based on the **view** of the model, not the model itself. Moving the view moves the source point. For instance, you cannot set the source point on the front of the model, move the view to the rear, and paint the cloned texture from the front to the rear of the model. (The source point has moved to the same spot in the view, but on the rear of the model.) Note, however, that you can use a split screen to select a source in one view and clone it to another.

In these modes, think of the source point as being a position within the window. As you zoom, pan, and orbit the model, the source point remains in the same position in the window, but not the same position on the model.

Video: To learn how to clone paint using the Clone Stamp, watch Cloning Paint in Mari. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Choosing a Source Point

When cloning from a particular channel, the **Clone Stamp** tool takes a “snapshot” of that channel as it appears on screen and uses that as the source. As a result, the resolution of the cloned texture can depend on the current
display resolution within Mari. Higher display resolutions give better cloning results, but can dramatically slow Mari's performance.

When cloning from a source point on the model's surface, bear in mind that the shape of the model determines what gets cloned. For example, if your source point is on a part that sticks out from the model, the space between that part and the rest of the model does not provide a source texture when painting.

The original model, with clone point set in the paint on the arm.	See how the cloned texture comes directly from the surface as seen in the view.

Setting a Start Point

You can set the start point for each stroke. This can be either:

- **Relative to the source point** - Your first brush stroke starts from the source point, but for later strokes the cloned texture starts from a position relative to the source point. For example, if your second brush stroke starts from 2 cm below the first, the texture cloned there is the texture 2 cm below the source point in the original image. This is the default setting.
- **From the source point** - All brush strokes take their cloned texture starting from the source point. As you finish each stroke, you can see the source area preview snap back to the original source point.

This example shows the difference between the two modes:

Source texture for cloning.	Texture cloned relative to source point.	Texture cloned not relative to source point.

Image Options

When you are painting using an external image, you can choose to have Mari display a preview of the image, floating over the canvas. This makes it easier to see exactly which parts of the source image Mari is cloning when you paint. This is very similar to using the **Paint Through** tool (see Painting Through an Image or Gradient).

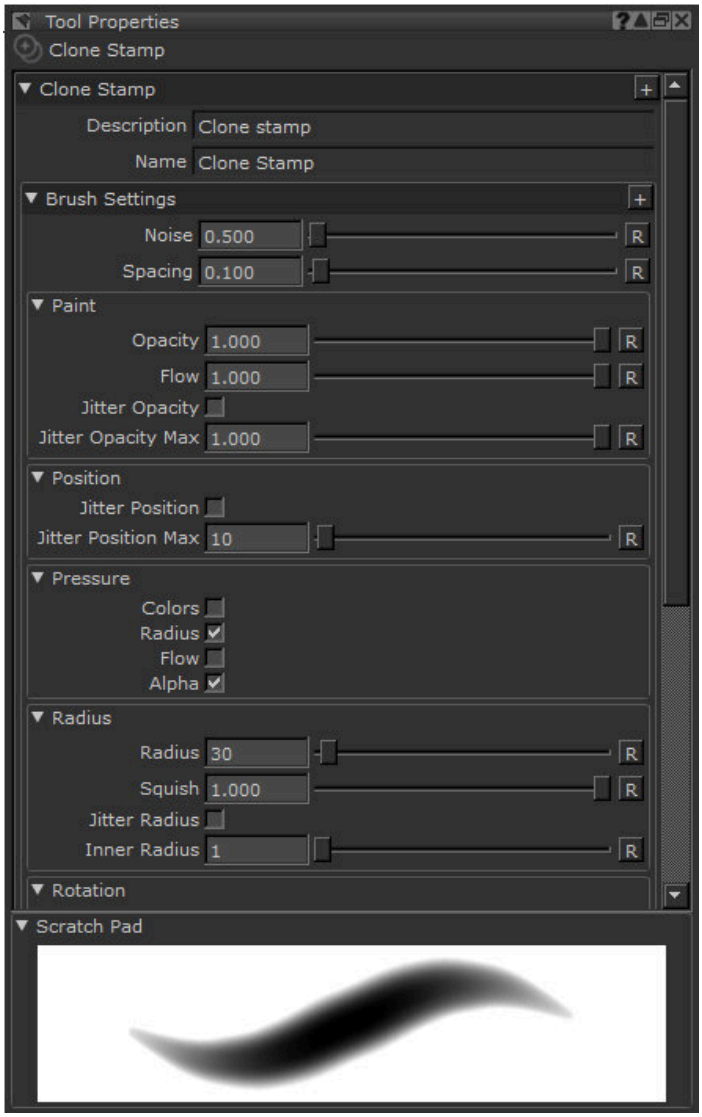
Mari can either scale the external image as you zoom in or out on the model while painting, or keep it at the same size. You control this with the **Scale Lock** setting: if on, Mari locks the image size relative to the model, at the current zoom level. As you zoom in or out, the image that you're cloning from zooms in or out too, so the cloned texture always stays at the same scale, relative to the surface you're painting on.

If **Scale Lock** is off, the image scale does not change when you zoom in or out.

Tip: Mari copies textures you clone from an external image at the resolution of the source. Mari then paints these into the paint buffer at the buffer's resolution. For example, a 4k image provides 4k source texture for cloning, which Mari can then paint into the paint buffer at a maximum of 16k (the maximum size of the paint buffer). See Controlling the Way Mari Applies Paint to Your Model for details on paint
buffer resolution.

Setting the Clone Stamp Options

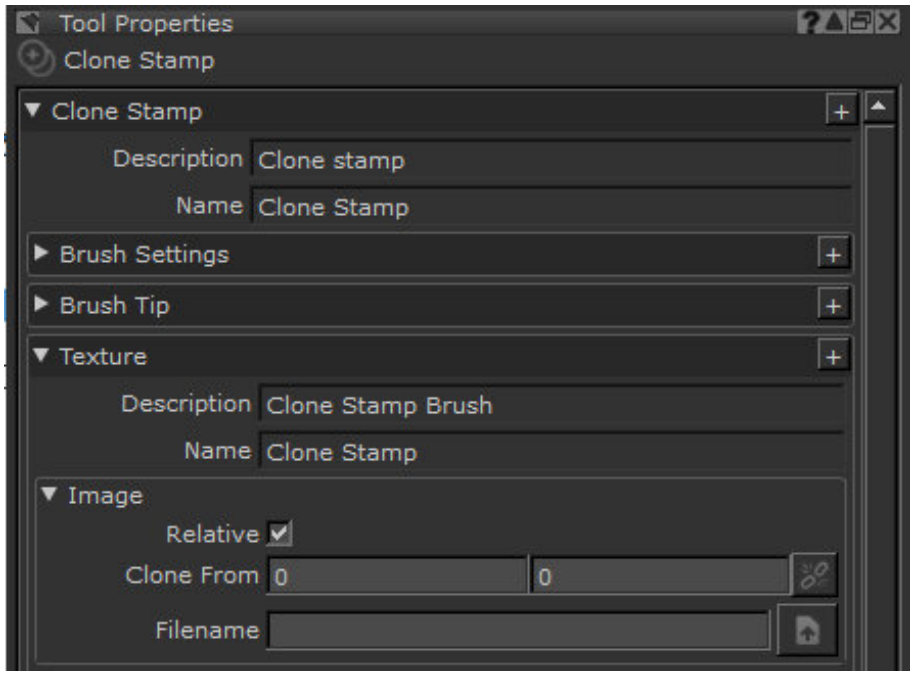
1. Start Mari and open your project.
2. On the **Tools** toolbar, click to select the **Clone Stamp** tool.
3. Open the **Tool Properties** palette.
   The current properties of the selected tool display. Scroll down to view them all.



4. To customize the brush you're using for your selected tool, set the options under the **Tip** and **Geometry** sections.

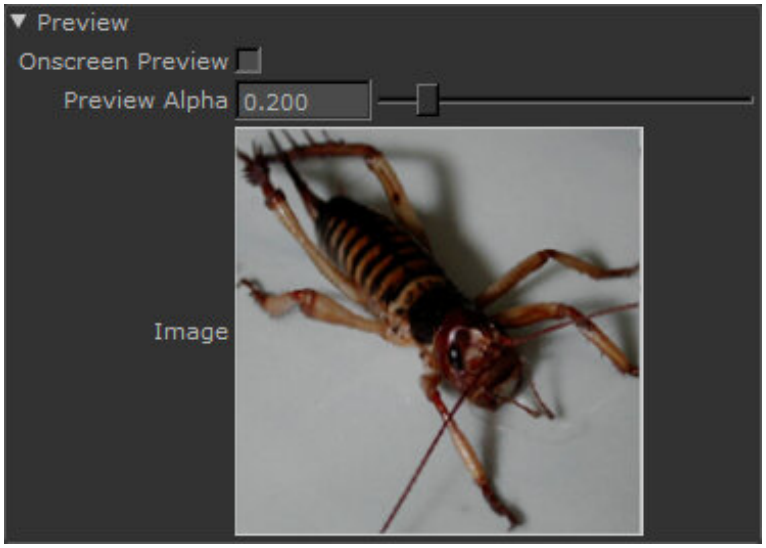
   These are the same options available for brushes in general - see Configuring Brushes.

5. To set the brush strokes to start relative to the source point, click the **Relative** checkbox under **Texture** > **Image**.



If this is not selected, the clone brush strokes always start from the source point (see Setting a Start Point for examples).

6. To turn the onscreen preview on when cloning from an external image, under **Texture** > **Preview**, click to turn **Onscreen Preview** on. Set the **Preview Alpha** to your preferred value (higher makes the preview clearer, but makes it harder to see the current paint in the channel).



As you click in the external window to change the source point, the preview moves around on the main canvas.

7. To set whether Mari zooms images relative to your view, turn **Scale Lock** on or off, under **Texture** > **Transform**.

   If this is on, Mari locks the image size relative to the model, at the current zoom level. If you zoom in or out, Mari also zooms in or out the image that you're painting through. If this option is off, the image scale won't change when you zoom in or out.

8. To set whether Mari resizes source images to fit the paint buffer, change **Reset Size** setting, under **Texture** > **Transform**:
   - **ToOriginalSize** - Displays and clones the source image in its original size, in screen pixels. For example, a 1024x1024 image takes up 1024x1024 pixels on screen. This is the default.
   - **RelativeToPaintBuffer** - Displays and clones the source image resized so that one pixel in the image matches one pixel in the paint buffer. For example, a 1024x1024 image in a 2k (2048x2048) paint buffer appears onscreen as half the size of the paint buffer.

9. To set whether Mari preserves transformations when you switch source images, change the **Auto Reset** setting, under **Texture** > **Transform**:
   - **DoNotReset** - Preserves any transformations (such as moving, resizing, or rotating) you have made to the source image that you're cloning, if you switch to using another source image. This is the default.
   - **Reset** - Resets the image to the default size, location, and rotation, each time you switch to another source image. For example, if you're using a 2k paint buffer and a 4k image, the image always appears to be twice the size as the paint buffer onscreen.

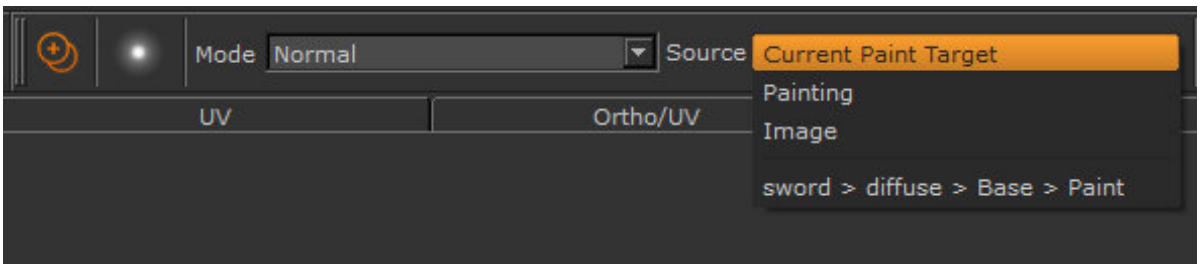
> **Tip:** Use **DoNotReset** if you want to clone several images of the same size. This is specifically useful if you want to switch between using multiple 4k textures, as it saves you having to resize the source image
> preview from the huge default every time you switch.

Now that you've set up how the clone tool works, you can get on and start painting.

Clone Stamp

To clone from an existing current layer:

1. If you haven't already, select the **Clone Stamp** tool.
2. In the **Tool Properties** toolbar at the top of the screen, select the clone source from the **Source** dropdown.

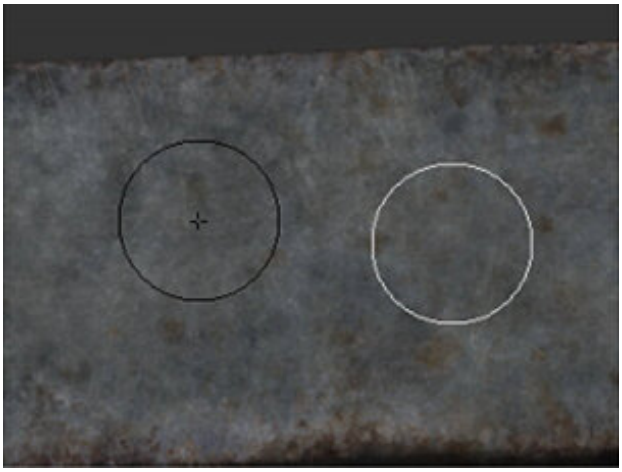


The list includes all channels in your project, plus Painting (clone from the unbaked paint) and Image (clone from an external image). The default value for Source is **Current Paint Target**.

> **Tip:** If you can't see the **Tool Properties** toolbar, open it by right-clicking in the toolbar space at the top of the window and selecting **Tool Properties** from the bottom of the dropdown list.

3. Select **Current Paint Target**, select the layer you want as a source, then hold down **Ctrl**/**Cmd** and click on the model's surface to set the source point.

   Mari highlights the source point with a cross.



4. Paint normally. As you paint, Mari copies the texture from around your source point in the source channel:
   - The white circle is your brush.
   - The black circle is the source area - as you move your brush, this circle shows you the texture you'll be cloning from.

   You can select any preset brush to use with the clone stamp tool, or edit the brush as normal (see Configuring Brushes).

To clone from the paint buffer:

1. To set the clone source in the unbaked paint (that is, the contents of the paint buffer), from the **Tool Properties** toolbar at the top of the screen, select **Painting** from the **Source** dropdown.
2. **Ctrl**/**Cmd**-click to set the source point. Paint normally:
   - As above, the white circle is your brush.
   - The black circle is the current source area.

> **Tip:** Bear in mind that if a particular area doesn't have any unbaked paint on it (that is, there is no data in the paint buffer for that area), no clone data is available from the area. It's as if you're trying to paint from an area that doesn't have anything on it.
>
> If you're trying to clone and getting nothing, check that there actually is unbaked paint on the area you're cloning from!

To clone from surface directly into the buffer:

1. To clone from the surface straight up into the paint buffer directly over the clone point, select the layer to clone from as normal. When you set the source point, hold down **Shift**+**Ctrl**, or **Cmd**+**Ctrl** for Mac, and click on the model. Paint normally. As you paint, Mari copies the area you paint from the layer surface into the paint buffer.
2. If you are cloning from the current channel and layer, you won't see any visible effects, as Mari is just copying the paint right on top of itself, but you can then switch tools and edit the paint and then re-bake. This lets you make fine adjustments to the paint already baked into a layer, or to clone sections of paint from one layer to another (for example, to clone from a displacement channel into the diffuse channel to get the basic details of the
   surface contours to paint).

To clone from an image:

1. To clone from an external image, drag the image from the **Image Manager** (or from a shelf) onto the main canvas.

   The image opens up in a separate window. You can zoom the image in and out by holding down **Ctrl**/**Cmd**+**Alt**, clicking and dragging.This sets the clone source to **Image** automatically.

2. Click in the separate image window to set the source point. The white circle in this window shows where the current clone source is, as you move the mouse.

3. Paint in the main Mari window as normal.

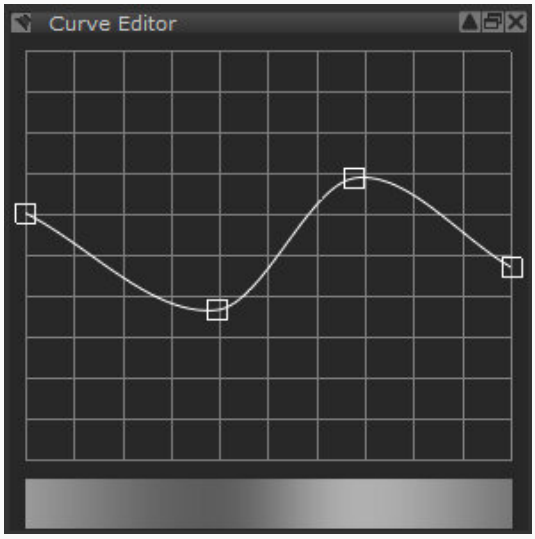
Tip: Experiment with the various options and modes of clone stamp. Try using a custom brush, with an unusual texture, lower alpha, and so on to find which settings work best for you.

Applying a Color Grade as You Clone

You can apply a color grade to your source image as you paint. You can apply the color grade to any clone operation, whether from an external image or from a channel's surface. You can:

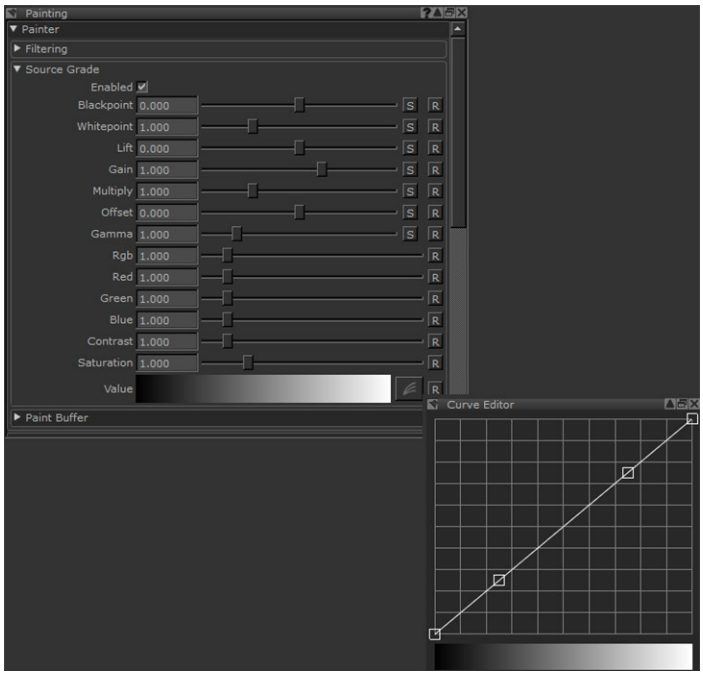
- Change the color values in the source image, either all at once, or by RGB channel.
- Alter the **Contrast** or **Saturation**.
- Use the value curve to apply a specific change to the colors. Mari grades the colors by converting the input value (on the x axis) to the matching value on the y axis.

For example:

Setting	Result
Default settings	
Blue component turned down:	
Colors at default settings, but **Contrast** turned down:	
All colors and **Contrast** at default values, but the following value curve applied: 	

To color grade the image:

1. Open the **Painting** palette and scroll down to the **Source Grade** section.



   This section lets you control the color grade for your source image.

2. Click to check the **Enabled** box.
3. You can use the controls to, among others:
   - Change the color values in the source image, either all at once (using the **Rgb** slider) or one by one.
   - Alter the **Contrast** or the **Saturation**.
   - Use the value curve to apply a specific change to the colors. Mari grades the colors by converting the input value (on the x axis) to the matching value on the y axis.

> **Note:** For more information about the **Source Grade** controls, see the Painting Palette.

4. Select the clone tool.
5. Paint some sample strokes, experimenting with the settings in the **Source Grade** section.

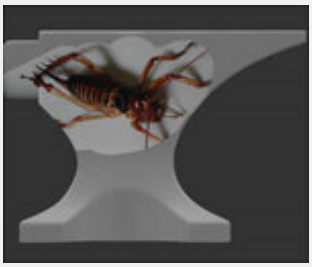
> **Tip:** Use the sliders to change the color balance in the red, green, and blue components. Try altering the contrast. Use the value curve control to change the color values in the source image. Paint a number of sample strokes to see how the color grading works.

Painting Through an Image or Gradient

Paint Through is closely related to clone stamping. The **Paint Through** tool lets you drag an image from the **Image Manager** and hover it over your model. As you paint, Mari applies the area of the image underneath your brush to the model. It's like clone stamping, except that you're always cloning directly from the preview onto the model below it.

Unpainted model.

Image overlaid, painting through.



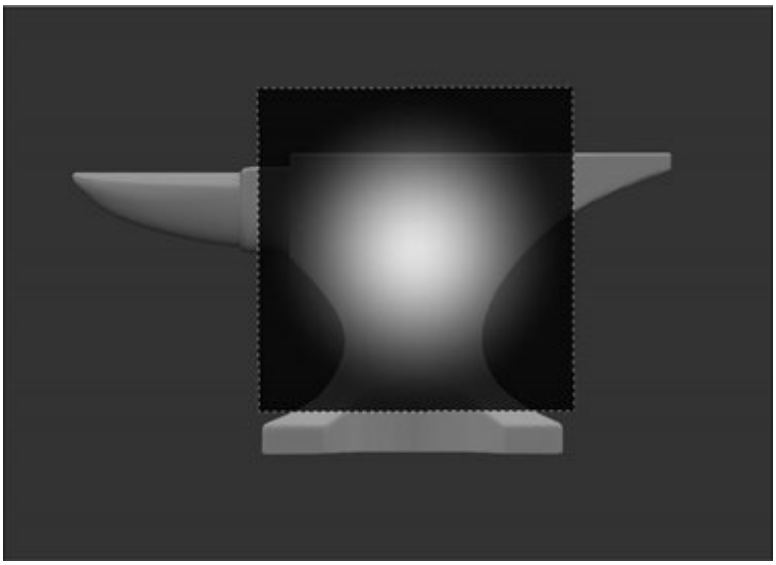
The result.

As you paint, you can move the source image around, rotate it, and resize it.

Tip: If you paint off the edge of the image, no clone data is available. You can choose to tile the source image. If this is on, when you paint off the edge, Mari repeats the image (so you always have source data).

Use the **;** (semicolon) key to toggle source image tiling on and off.

You can also use the **Gradient** tool to paint through a linear or radial color gradient instead of an image. For example, the following illustration shows a radial gradient being stamped onto the model:



Paint Through

Paint Through has two modes:

- **Paint through** - you hover the image preview over the model and stamp it down onto the surface.
- **Image clone** - you hover the image next to the model, set a source point, and clone from that point.

The **Image clone** mode is like the normal **Clone Stamp** tool, but you have more control over the source image (you can resize, stretch, and rotate it).

To lock the model and paint buffer to the source image so that Mari automatically pans or scales the model relative to the image, check the **Lock Object** checkbox in the **Paint Through** toolbar.

You can also lock the source image proportions using the **Image Scale** button, so that any changes to scale are applied proportionately to both axes. If **Image Scale** is off, you can alter the image by different amounts on the vertical and horizontal axes.

As you paint through, you can switch images. By default, when you switch to another source image, Mari applies any transformations you've made to the last source image onto the new one. For example, if you have resized the source image onscreen, moved it around, or rotated it, the next source image appears the same. You can control this behavior through the **Auto Reset** option.

You can change the algorithm Mari uses to clone from a resized texture. Set this on the **Painting** palette, under **Painter** > **Filter**. Your options are:

- **Nearest** - preserves edge detail, but gives quite "blocky" textures.
- **Bilinear** - gives good results, but can produce square artifacts at extreme zoom. This is the default setting.
- **Bicubic** - provides more rounded results, slightly blurrier but without the square artifacts.

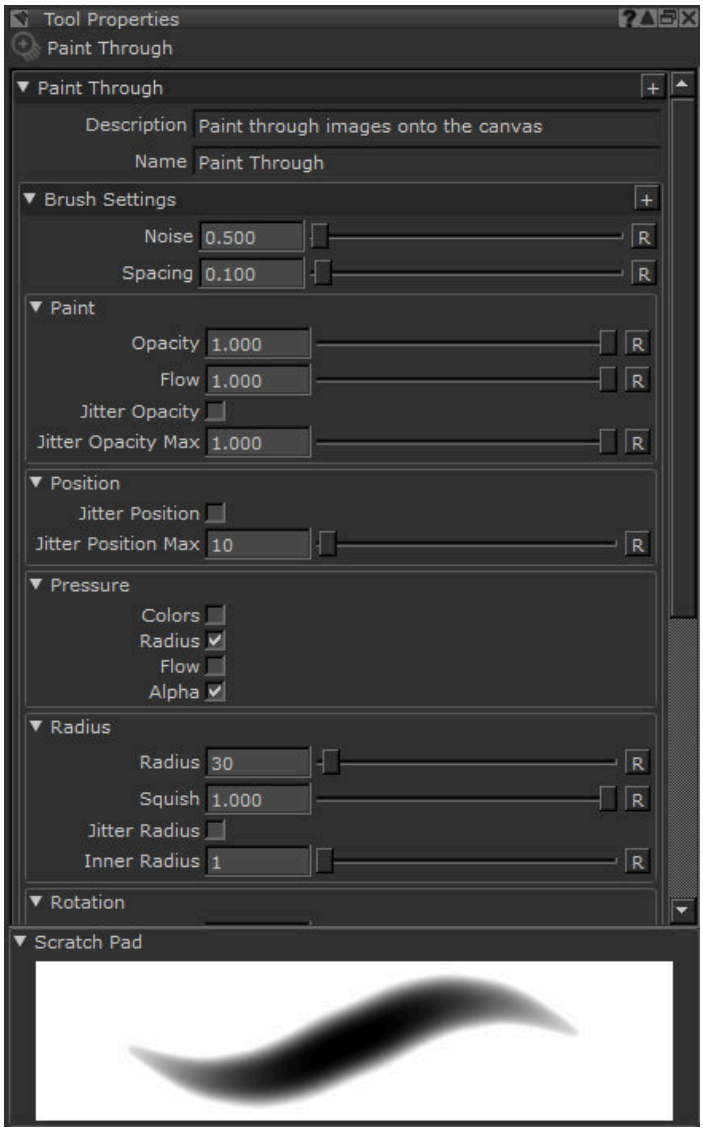
As with the **Clone Stamp** tool, you can adjust the color grade of your source image as you paint.

Tip: See Clone Stamping for details of the **Clone Stamp** tool and using color grades.

Video: Watch Using the Paint Through Tool to learn how to paint assets using images.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Setting the Paint Through Options

1. Start Mari and open your project.
2. Click to select the **Paint Through** tool.
3. Open the **Tool Properties** palette.
   The current properties for **Paint Through** display. Scroll down to view them all.

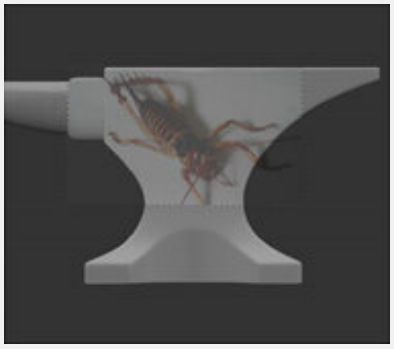


4. To customize the brush you're using for **Paint Through**, set the options under the **Brush Settings** and **Brush Tip** sections.

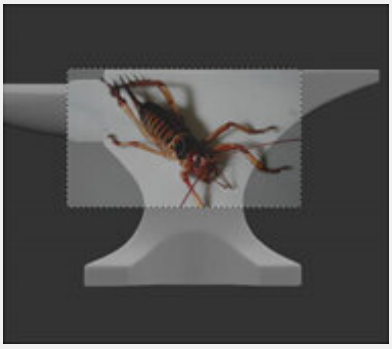
   These are the same options available for brushes in general, see Brush Properties.

5. To change the prominence of the onscreen image that you're painting through, under **Texture** > **Preview**, set the **Preview Alpha**.

   Higher values make the preview clearer, but make it harder to see the model underneath.



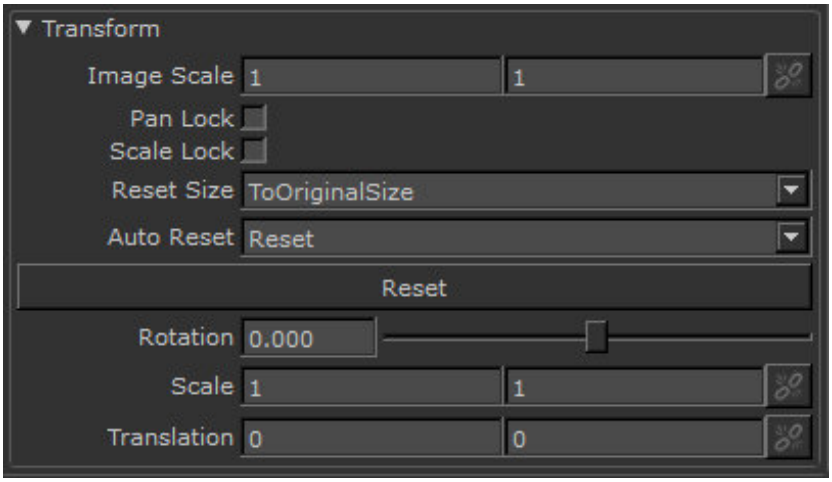
Preview alpha of 0.35.



Preview alpha of 0.75.

6. You can lock the source image proportions using the **Image Scale** button, so that any changes to scale are applied proportionately to both axes. If **Image Scale** is off, you can alter the image by different amounts on the vertical and horizontal axes.
7. Set Mari to automatically pan or scale the source image using the **Pan Lock** and **Scale Lock** settings.

   When the locks are enabled, Mari locks the image position and size relative to the model. If you pan or zoom the model in or out, the image that you're painting through pans or zooms in or out too.



   If **Pan Lock** and **Scale Lock** are disabled, the image position and scale does not change when you change the view of the model.

> **Video:** Watch a quick demo on how to use **Pan Lock** and **Scale Lock**.
> This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

8. To set whether Mari resizes source images to fit the paint buffer, change **Reset Size** setting, under **Texture** > **Transform**:

- **ToOriginalSize** - Displays and paints through the source image in its original size, in screen pixels. For example, a 1024x1024 image takes up 1024x1024 pixels on screen. This is the default.
- **RelativeToPaintBuffer** - Displays and paints through the source image resized so that one pixel in the image matches one pixel in the paint buffer. For example, a 1024x1024 image in a 2k (2048x2048) paint buffer appears onscreen as half the size of the paint buffer.

9. To set whether Mari preserves transformations when you switch source images, change the **Auto Reset** setting, under **Texture** > **Transform**:
   - **DoNotReset** - Preserves any transformations (such as moving, resizing, or rotating) you have made to the source image that you're cloning, if you switch to using another source image. This is the default.
   - **Reset** - Resets the image to the default size, location, and rotation, each time you switch to another source image. For example, if you're using a 2k paint buffer (maximum size) and a 4k image, the image always appears to be twice the size as the paint buffer onscreen.

> **Tip:** Use **DoNotReset** if you want to paint through several images of the same size. This is specifically useful if you want to switch between using multiple 4k textures, as it saves you having to resize the source image preview from the huge default every time you switch.

Now you've set up how the **Paint Through** tool works, you can get on and start painting.

Painting Through an External Image

1. Drag your source image from the **Image Manager** (or from a shelf) onto the canvas.

   The image “floats” above the canvas, ready to paint through.

> **Note:** If you do not have the **Paint Through** tool active when trying to drag an image onto the canvas from the **Image Manager**, Mari automatically switches you to the **Paint Through** tool to continue.

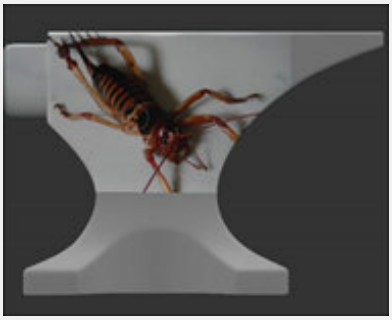
2. Get the image how you want it:

To...	You need to...
Move the image around the canvas	Hold down **Shift**, click on the image and drag it.

To...	You need to...
	Or, click on the white square in the middle of the image and drag it.
Resize the image, preserving the scale	Hold down **Ctrl**/**Cmd**+**Shift**, click anywhere on the canvas and drag. Or, hold down **Shift**, click on a side of the image and drag it.
Resize the image in one direction	Click on a side of the image and drag it.
Rotate the image	Hold down **Ctrl**/**Cmd**, click anywhere on the canvas and drag. Or, click outside the image and drag.

3. If you just want to stamp the image straight onto the model, press **'** (apostrophe).
   Mari stamps down the image, as it is currently floating over the model, onto the surface.

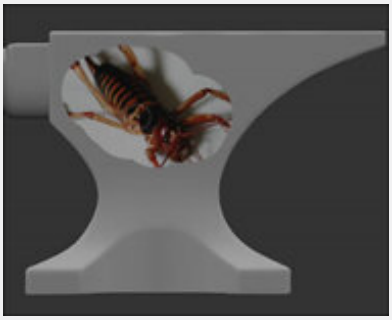
Model before painting.



Texture stamped down.

4. To paint using the brush, just click and drag to paint normally.
   As you paint, your brush paints down the texture from the image at the point you're painting.

Model before painting.



Texture painted through.

Tip: Painting with the brush lets you more subtlety control how Mari applies the paint - for example, you can use a brush with a relatively low alpha, and "build up" the paint on the model using multiple strokes.

You can apply a color grade to the source image as you paint through. Set the color grade settings in the **Source Grade** area of the **Painting** palette. See Clone Stamping for details of how the color grade settings work and the effects you can achieve. For more information about the **Source Grade** controls, see the *Painting Palette* section in the *Mari Reference Guide*.

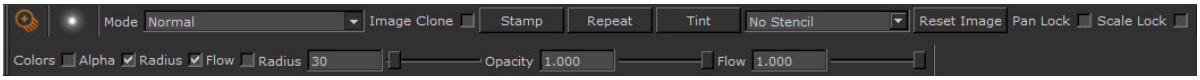
Experiment: Experiment with different brush settings and their effect on painting through. Adjust details such as opacity, alpha, and noise to affect the way the texture is painted onto the model.

5. You can apply a **Tint**, which allows you to combine a color with the image to apply a tinted version of the image using **Paint Through**.
6. You can also apply a **Stencil**. The stencil allows you to paint through an image to create a design of the image on the model.

   By default, there is **No Stencil** applied to the model.

Cloning from a Floating Image

1. In the **Tool Properties** toolbar at the top of the screen, enable the **Image Clone** checkbox:



2. Make any other changes you want to the size, rotation, stretch, or position of the source image.

   This is exactly the same as in the normal mode.
3. To move the image next to where you want to paint, **Ctrl**/**Cmd**-click on the floating image to set the source point.

   The clone source preview displays in a red circle, while your brush is the white circle.

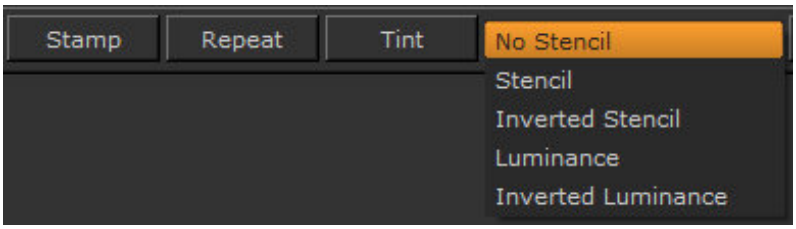
4. Paint normally.

   Mari clones the texture as you paint, taking the resolution for the cloned texture from the source image. For example, a 4k image provides cloned texture at 4k resolution.

Using Tint and Stencil Options

1. On the **Paint Through** toolbar, there are additional buttons to the right of the main options. These give you **Stamp**, **Repeat**, and **Tint** options for the **Paint Through** functionality.

2. **Stamp** is a simple button that works the same as stamping down an image using a shortcut. You can drag your image around the canvas, resize it, and rotate as usual before using **Stamp** to apply the image directly to the model in full.
3. To repeat the image you wish to apply to the model, first press the **Repeat** button before painting on any part of the model. The image repeats across the painted parts of your model. To turn this off, simply press the button again.
4. To apply a **Tint**, press the **Tint** button and select your options from the following dropdown menu.



This allows you to combine a color with the image to apply a tinted version of the image.

- By default, **No Stencil** is the selected option, which simply tints the image color and does not change any other aspect.
- Selecting **Stencil** allows you to paint through an image to create a design on the model. **Stencil** ignores the color on the stencil and follows only the alpha value.
- Selecting **Inverted Stencil** allows you to color in all parts of the model where the stencil is transparent, leaving the stencil as the model's original color.
- Selecting **Luminance** allows you to paint through an image in much the same way as you would with a stencil. However, by setting the option to **Luminance**, this allows you to use an image's luminance as the alpha when painting. This works especially well for dirt, noise, blemishes, and other similar effects.
- Selecting **Inverted Luminance** allows you to create a similar effect as **Luminance** but uses the lighter parts of the image as alpha. This can be used to pick up scratches and fine lines.

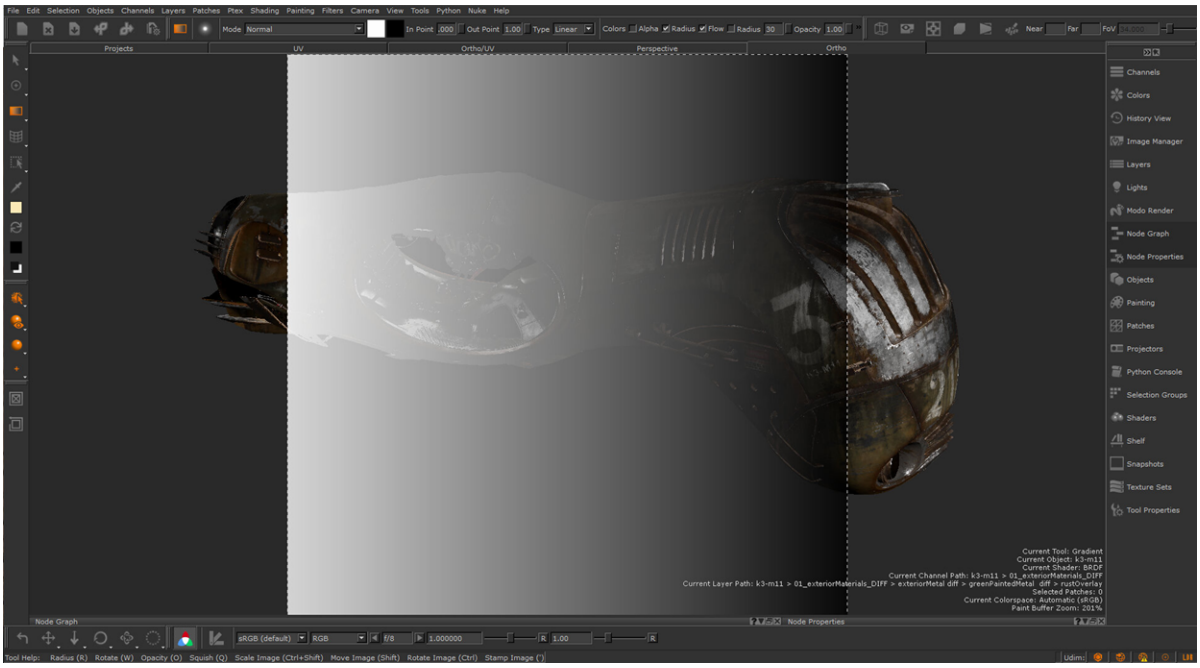
Video: To learn how to use images as stencils, watch the Using Stencils in Mari video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Painting Through a Gradient

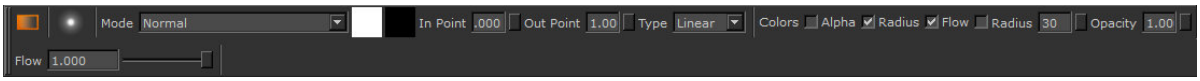
To paint through a color gradient (instead of through an image):

1. Click to select the **Gradient** tool.

   A gradient displays above the canvas.



2. To configure the gradient, change the settings on the **Tool Properties** toolbar at the top of the window.



You can set:

- The two colors for the gradient - click on the swatch to change the color.
- **In Point** - select a proportion of the gradient (left-to- right linear, or in-to-out radial) at which the starting color begins changing to the ending color (at 0.00, the transition begins at the far left linear or dead center radial; at 0.50 the transition begins halfway; at 1.00 there is no transition, it's all start color).
- **Out Point** - select a proportion of the gradient (left-to- right linear, or in-to-out radial) at which the starting color turns completely into the ending color (at 0.00, there is no transition, it's all end color; at 0.50 the
transition ends halfway; at 1.00 the transition ends at far right linear, or the outer circumference radial).
- **Type** - select from the dropdown menu whether the gradient is **Linear** (from one side to the other) or **Radial** (from the center out).

As you change the properties the gradient shown onscreen updates.

3. Size, move, and rotate the gradient, in the same way as described in Setting the Paint Through Options.
4. Paint through the gradient, in the same way as described in Painting Through an External Image.

Multi-Paint

Mari's Multi-Paint mode enables you to paint into up to eight different streams from a single **Multi-Channel Paint** node. This means you can be painting into your color stream, whilst simultaneously painting into the bump and displacement streams. This is a great tool for speeding up the painting process, and can be really helpful when you want to use texture packs, as you don't need to manually edit each stream and spend time matching up the different images.

Once you've created a Multi-Channel Paint node, you can edit and control these streams for that node via the **Multi-Paint** palette, found in the Palettes toolbar. This palette enables you to control the color or image applied to each individual stream during the painting process.

The Multi-Paint Palette's settings are stored by shader model. This means if you are using a Multi-Channel Paint node that is using **BRDF**, and you switch to a different Multi-Channel Paint node in your node graph that is using **3Delight Principled**, the panel will update to reflect the new node's settings and shader model.

You can also set up Multi-Paint Presets to save time when creating Multi-Channel Paint node. These presets save specific stream settings that will automatically populate the Multi-Paint palette when applied.

To use the Multi-Channel Paint node in your project, just merge the node into your network using a **Multi-Channel Merge** node in the same way that you would work with a **Material** or standard **Paint** node.

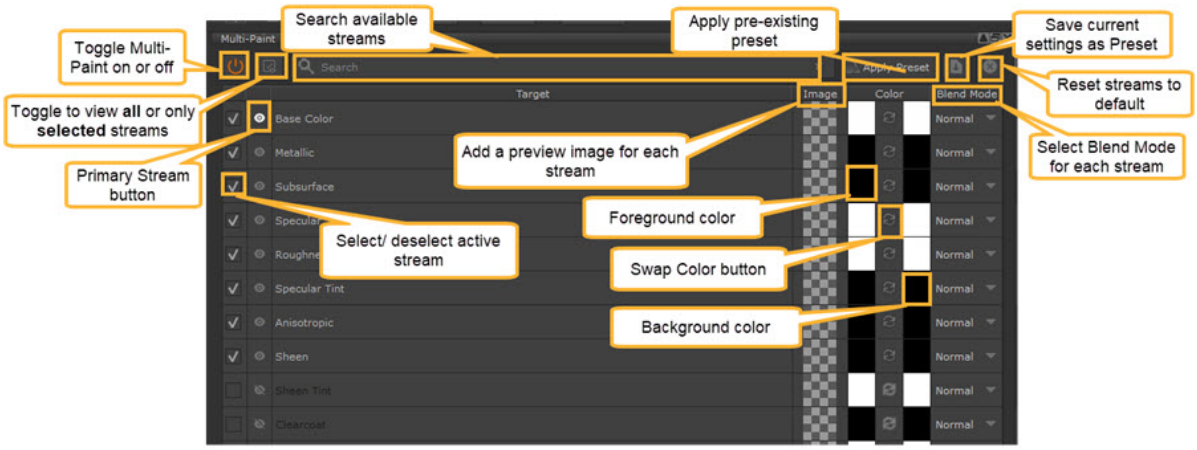
Tip: Multi-Paint mode can be used alongside **Paint** nodes, and you can switch between single-channel painting and multi-channel painting by selecting the node you want to paint on.

Experiment: Multi-Painting is a complimentary workflow that can be easily added into existing projects where you might already be using the standard paint or material network. Try adding Multi-Paint in your existing projects!

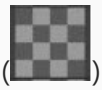
To find out more about the **Multi-Channel Paint** node, go to Multi-Channel Paint Node. The following sections of this page will cover how to utilize the **Multi-Paint** Palette.

Multi-Paint Palette

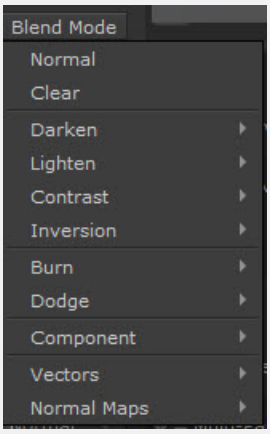
The **Multi-Paint** palette is the place to add and edit the specific streams you want to paint into. You can select a maximum of eight streams per Multi-Channel Paint node.



Using the Multi-Paint Palette

Action	Access By...	Description
Adding/ Removing Images		
Add images to the palette	• Drag & drop image from **Image Manager** • Drag & drop image from File Browser • Click the checkered box in Image column () and select image from **Image Manager** • Drag & Drop Image Group from Image Manager	Drag and Drop/ select specific images into specific streams. When you drag & drop an Image Group from the **Image Manager**, the images will auto-populate to streams based on the image names, or by the presets applied in the Image Group's Ingest Template Dropdown List.
Clear image from Palette	• Right-click > **Clear Image**	Clears the image for a specific stream. To clear all images and Stream settings, go to **Reset Streams** .

Action	Access By...	Description
Managing Streams		
Search available streams	Use the Multi-Paint Search bar	The streams available to search will correspond to the shader model defined when you created the **Multi-Channel Paint** node.
Select active streams	Select / deselect a stream by toggling the active button:	Select up to eight active streams per **Multi-Channel Paint** node to paint into.
View all / selected streams	Toggle button	Swap between viewing all streams from the shader model or the streams you have selected for Multi-Paint (up to eight).
Select Foreground / Background color	Left-click the Color tile and select color from the pop-up window.	Select the foreground and background color for 2 different painting color options.
Swap Foreground/ Background color	Toggle swap color button:	Swap between foreground and background color to change which color to use whilst painting.
Select Primary Stream	Click Primary Stream button: Active: Not active:	The primary stream defines which image or color will be displayed when you use the Paint or Paint-through tool.
Select stream Blend mode	Click **Blend Mode** dropdown for the specific	You can define the **Blend Mode** for each stream - these can be different from each other. Learn more about Blend Modes read Paint Blending Modes.

Action	Access By...	Description
	stream. 	
Reset Streams	Click Reset icon ( )	Will reset all streams to their default settings, including clearing all images.
Batch Edit streams	Left-click **Ctrl+** the streams you want to edit. Then right-click over the streams to choose the batch-editing option.	Batch edit the selected streams. The batch-editing options are: • Batch Edit Foreground Color • Batch Edit Background Color • Batch Edit Image • Batch Edit Blend Mode
Preset Workflows		
Save a Preset	Save Preset button ( )	Saves the current settings as a Multi-Paint preset. Learn more at Multi-Paint Presets.
Apply a Preset	Apply Preset button (Apply Preset)	Auto-fills the **Multi-Paint** palette with the selected Multi-Paint preset.

Multi-Paint Tools

The following painting tools are available in Multi-Paint:

Paint tools			
Select	Paint	Roller	Blur
Paint Buffer Eraser	Paint Through	Clone Stamp	Warp
Slerp	Pinup	Towbrush	Marquee Select
Transform Paint Buffer			

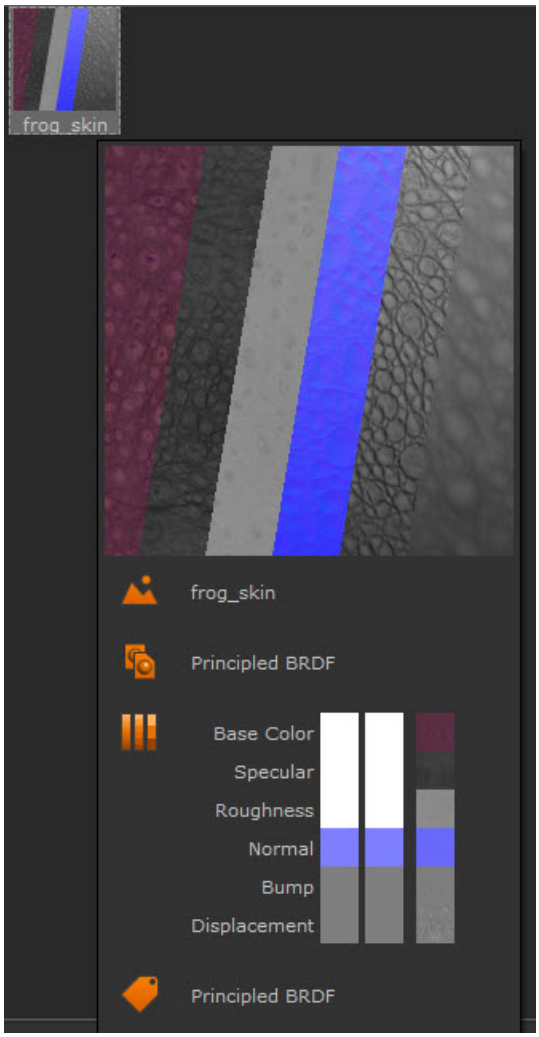
Multi-Paint Presets

You can create and save custom Multi-Paint Presets so you don't have to repeat set up every time you create a new Multi-Paint node.

To create a Preset:

1. Select your chosen streams and Color/ Image/ Blend mode settings.
2. Click the **Save Preset** icon ( ).
3. Name your Preset.
4. Select a shelf to save the Preset on.
5. Click **Save**.

Your preset will now be available when you click the **Apply Preset** button in the Multi-Paint palette. This will open a dropdown menu with all the Preset options. You can hover over the Presets to open up information on the streams and a preview of the image thumbnails.



Multi-Paint and Image Groups

You can drag image groups directly into the Multi-Paint palette and they will automatically populate the palette using image naming conventions.

Find out more information about using Image Groups with Multi-Paint at Image Groups.

Managing Images

Mari supports external images as a source for:

- clone stamping
- painting through

• picking colors in the **Color Manager**.

Mari's **Image Manager** palette stores images that are in use in the current project. This gives you quick access to the images, and very basic editing. For each image, the **Image Manager** shows a small preview and information such as the location of the image on disk. You can organize these images into **Image Groups**. Read more at Image Groups.

When you open a project, the **Image Manager** loads its images from source files on disk and keeps them in memory, ready to use. If any of those source files change after you open a project, the changes only appear in the **Image Manager** when you next open the project. If Mari can't find a source file for an image in the **Image Manager**, when you open a project, it removes that image from the **Image Manager**.

You can drag-and-drop Megascans assets from the **Texture Sets** palette to the **Image Manager** palette and use these images in your project. See Managing Megascans Assets and the Texture Sets Palette section in the *Mari Reference Guide* for more information.

You can also create new images in the **Image Manager** by:

• cropping them from existing images,

• running a script on an existing image, or

• generating them using a custom Texture Generator.

Mari stores these with the project (rather than loading them from a file on disk - unless you specifically save them on disk).

2D Paint Mode enables you to paint and edit any images in the Image Manager:

Simply select the image you wish to edit, right-click > **Send Image to 2D Paint**, or if you want to start with a fresh blank canvas, you can right-click > **Create New 2D Paint**. Mari will import a new object and import the image as a Paint node. You can then paint the image using the usual Mari tools.

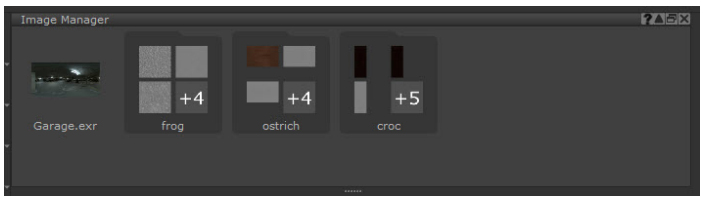
You can find more information on the Using 2D Paint Mode page.

Tip: If you need to use a common set of images for multiple projects in Mari, you can drag them to a personal shelf. See Storing Resources Using Shelves for details.

Opening the Image Manager Palette

1. Open the **Image Manager** palette:
   - right-click on the toolbar, and select **Image Manager** from the dropdown menu, or
   - from the **View** menu, select **Palettes** > **Image Manager**.

   The **Image Manager** palette displays, showing images currently in use in the project.



You can organize images in your Image Manager using Image Groups. This enables you to better organize your image without needing to scroll through a long set of thumbnails to find the one you need. Read more at Image Groups.

Video: To display the Image Manager menu when painting, press and hold **L**. Watch the demo. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Organizing images in the Image Manager

The following section covers how to organize your project images, through Image Groups and through the removing Unused Images workflow.

Image Groups

Managing unused images

Image Groups

Image Groups are a way to organize images within the Image Manager, so you can easily navigate and find your textures. Image Groups only exist within projects, so you can create specific groups dependent on what you need for your project.

Image Groups can be useful for organizing your images ready for Multi-Paint.

Creating an Image Group

To create an Image Group from images already in your Image Manager:

1. Select the images you want to include in the group.
2. Right-click and in the context menu select **Create Group**.

The images will now appear in the image manager as a Group, with preview thumbnails.

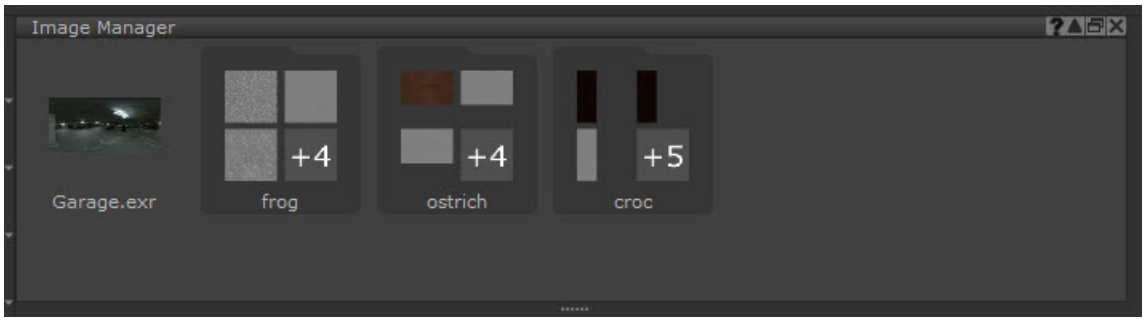
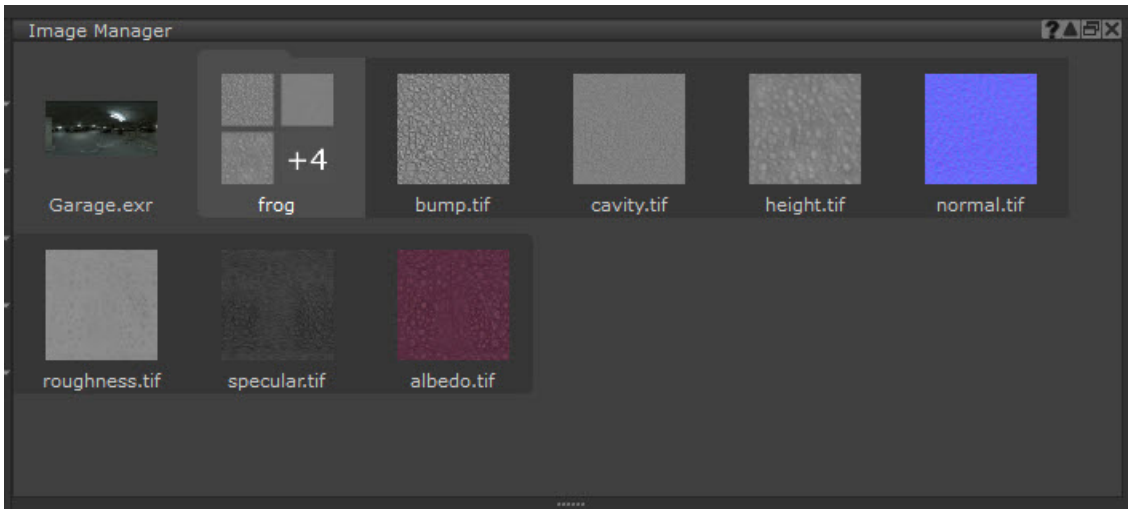


Image group opened up in Image Manager:



You can also add external images by dragging and dropping a folder from your file browser directly into the Image Manager. This will automatically create a folder with the images.

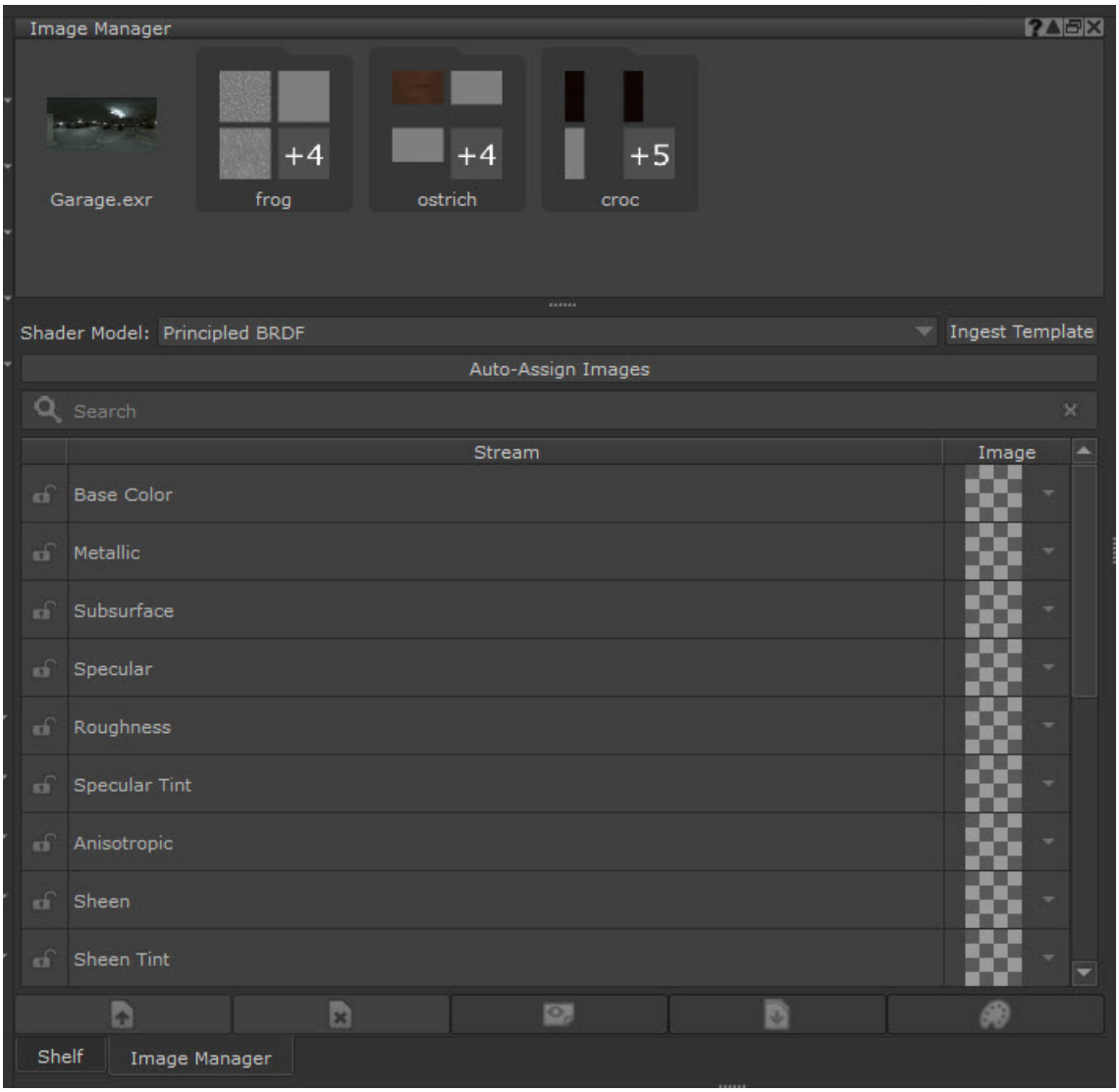
Managing Image Groups

- You can **Add to Group** and **Remove from Group** in the right-click context menu to add or remove images from group. You can also rearrange images within the Group for organization.
- You can drag and drop images that are in the Image Manager in or out of groups.
- You can rename your Image group by right-clicking the image group and selecting **Rename Item**.

Using Image Groups with Multi-Paint

If you have an Image Group you want to use with Multi-Paint, there are different methods to assign your images to specific shader streams. When you click on an Image Group, a properties panel will appear. You can use this panel to pre-assign Image Group streams ahead of inputting them into the Multi-Paint palette. Alternatively, you can let Mari automatically assign the streams based on naming conventions.

Image Manager with Image Groups properties panel:



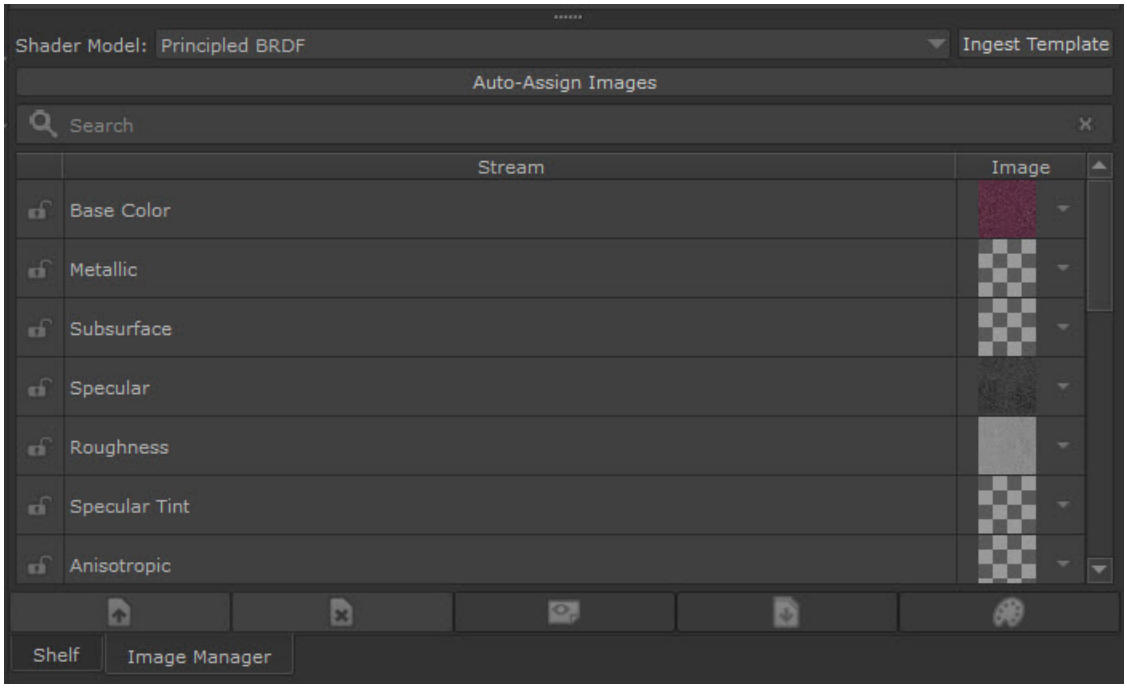
Automatically Assigning Channels in Multi-Channel Paint Node

Once you have an Image Group that you want to use for Multi-Channel painting, drag and drop the Image Group into the Multi-Channel Paint node's Multi-Paint palette. Mari will assign the images to the Multi-Paint streams based on the naming conventions of the images.

Assigning Image Groups to Shader Channels

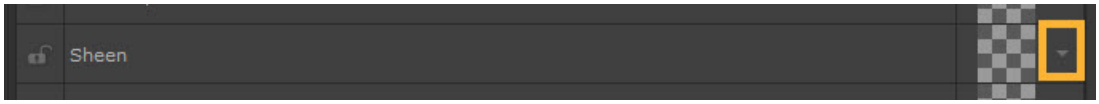
You can assign a shader model to the group and manually specify images that correspond to the appropriate stream. Once you've assigned the images, drag and drop the Image Group into the Multi-Paint palette, and the images will be projected in the appropriate streams at once.

Note: Ensure that your Image Group and Multi-Channel Paint node are using the same Shader Model for accurate assignment.



To assign a Shader Model to an Image Group:

1. Click on an Image Group. The Image Group properties panel will appear in the Image Manager.
2. Define the Shader model you want to use.
3. There are two methods to assigning images to streams:
   - **Auto-Assign Images** - Click the **Auto-Assign Images** button and Mari will automatically assign the images to the most appropriate stream based on the image's naming conventions.
   - **Manually assign images** - You can use the search bar to easily locate a specific stream. Click the arrow icon in the **Image** column to choose the image you want to assign to the stream.



4. Once you have set your images to your desired streams, drag and drop the image group into the Multi-Paint palette. You can now start painting.

Ingest Template Dropdown List

If your textures are coming from a specific vendor, you can choose the desired preset from the Ingest Template dropdown list. Mari will assign the appropriate images to each stream according to their naming

conventions.

When you click the Ingest Template button, a sub-menu opens with the following options:

- **Apply to Current Shader Model** - matches images to streams based on file names in the selected preset. This will only apply for the Shader model selected in the dropdown.
- **Apply to All Shader Models** -matches images to streams based on file names in the selected preset. This will be applied across all shader models.

Locking Streams

You can lock specific streams of an Image Group mapping by clicking the padlock icon to the left of the stream row ( ). Once locked, you can use the Auto-Assign tool and the locked stream will not be altered. You can still manually change the stream by using the image dropdown and picking a new image.

Organizing images in the Image Manager

The following section covers how to organize your project images, through Image Groups and through the removing Unused Images workflow.

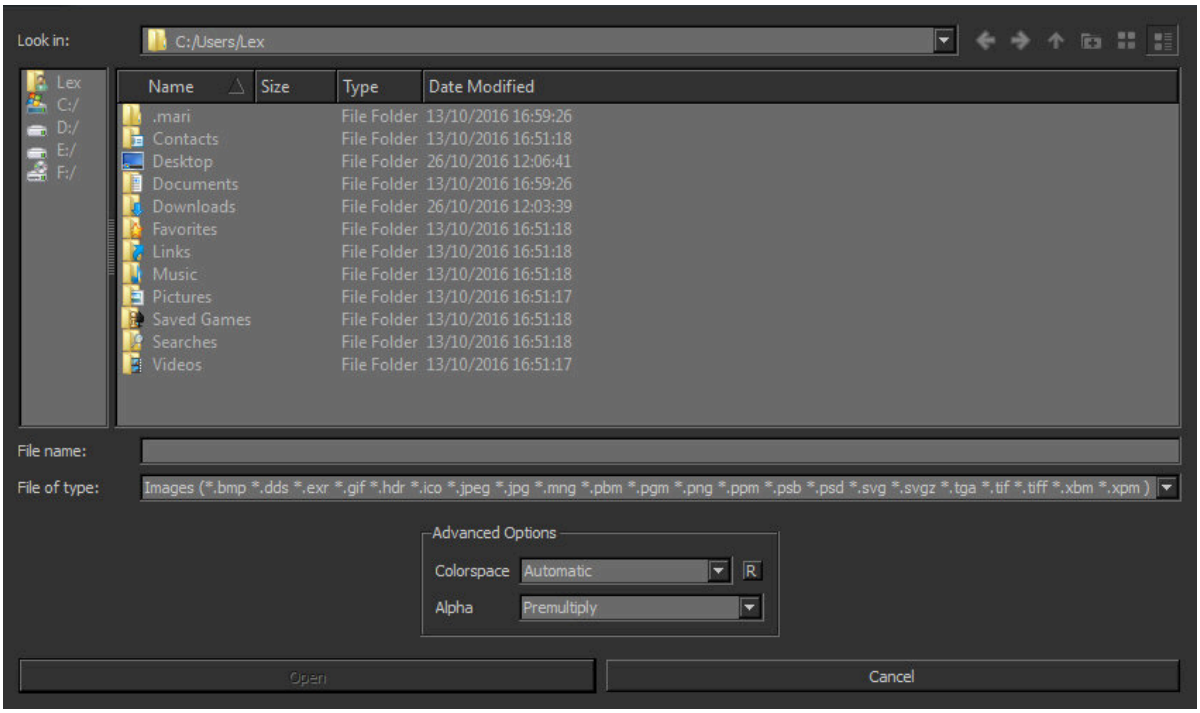
Image Groups

Managing unused images

Loading and Selecting Images

1. To load one or more images, click .

   The **Open an Image** dialog box displays.

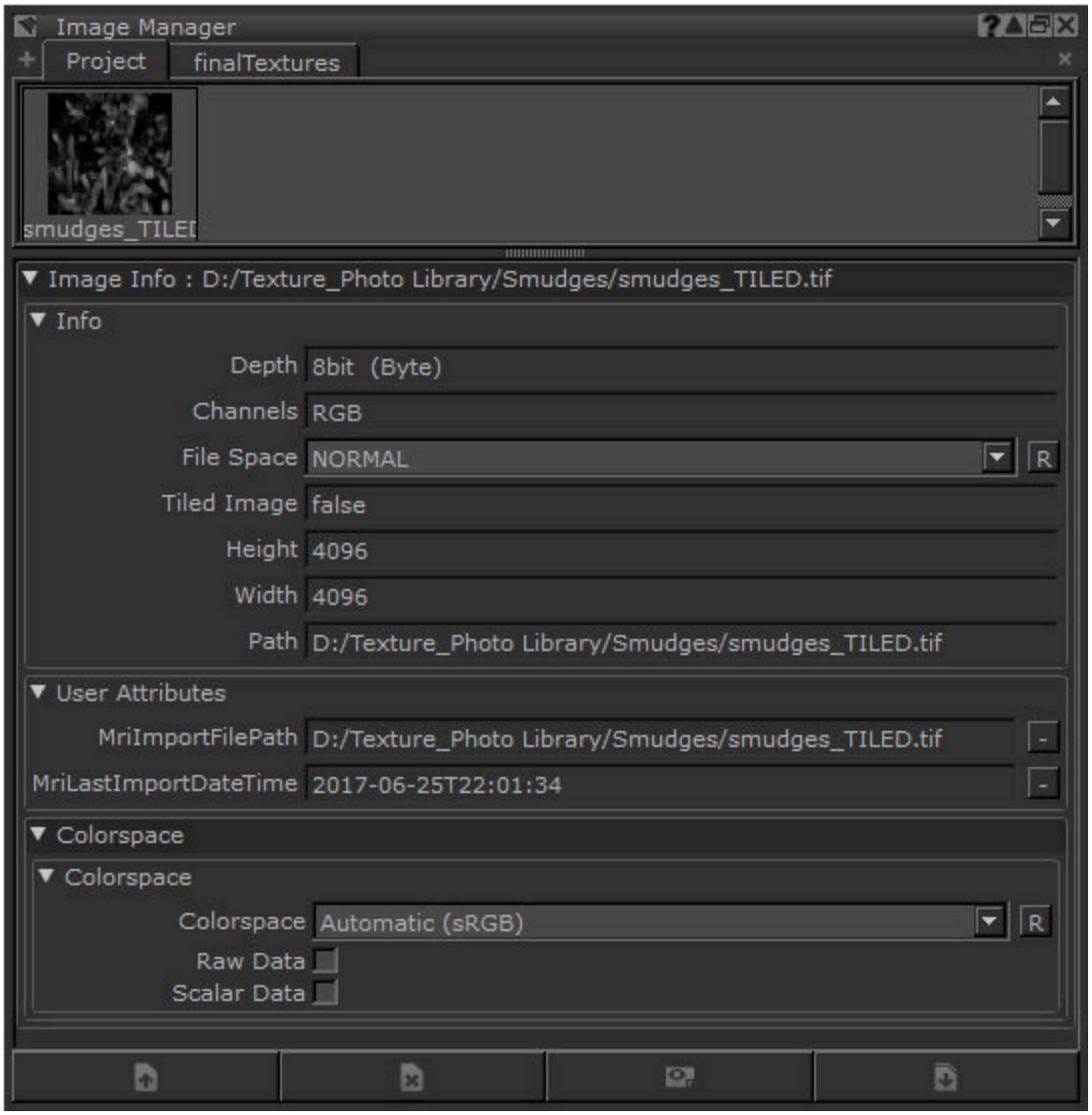


2. Select the image(s) you want to load and set the **Advanced Options:**
   - **Colorspace** - sets the colorspace to which the image data is converted.
   - **Alpha** - sets how you want the alpha channel to be handled.
3. Click **Open** once you are ready to load the image(s).

   The **Image Manager** shows a preview of the image(s). The actual image itself may be larger or smaller than the preview.

> **Note:** If you load a layered **.psd** file into the **Image Manager**, each layer is imported as a separate image.

4. Click on the preview to view information about that image.



Information displayed includes:

- Depth
- Channels
- Colorspace
- the image height and width, in pixels
- the path of the source file (from which Mari loads the image when you open the project).

Note: On Windows, file paths have a limit of 32,767 characters in length. If you load an image into the **Image Manager** that exceeds this length, the message **Invalid Image Path** is displayed in the **Path** field. This does not affect the image data or its properties, and you may continue to use it as normal.

5. To remove images from the **Image Manager**, select the image and click , or right-click on the selected image(s) and click **Close** in the dropdown menu.

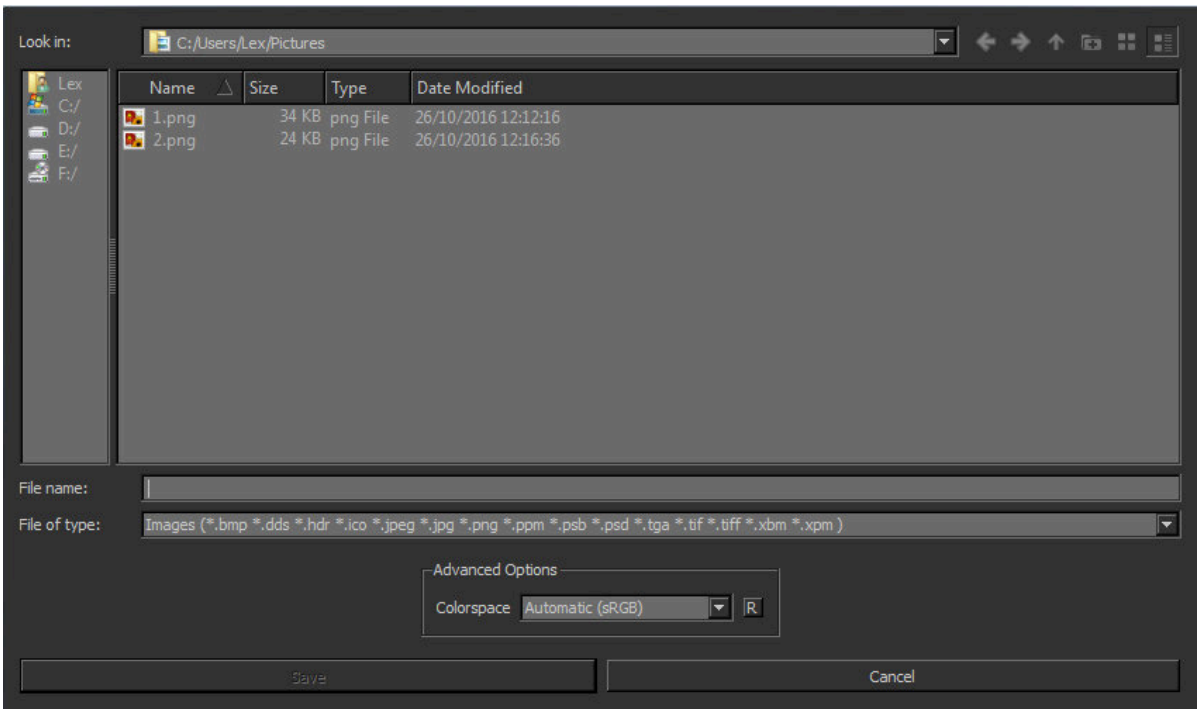
   The image(s) are removed from the **Image Manager**. This has no effect on the image source files - it just means you're not using those images in your current Mari project.

6. To view a larger version of an image, click , or select **View** from the right-click menu.

   The larger preview opens in a separate window. You can change the size of this preview by changing the size of the window. You can also crop the image from this window.

7. To save an image under a new filename, click , or select **Save As** from the right-click menu.

   The **Save As** dialog box displays.



8. Enter a new name for the image and click **Save**.

   Mari saves the image. From now on, Mari loads the image from the new filepath when you open this project.

> **Tip:** Double-clicking on an empty part of the **Image Manager** opens the **Open an Image** dialog. You can either select all the images you want and click **Open** to load them into the **Image Manager** palette, or you can drag-and-drop images into the **Image Manager** from your web browser.

Managing Megascans Assets

You can use Megascans assets (or Megascans) to paint your geometry. Once loaded, you can drag-and-drop Megascans in the **Image Manager** palette and use these images for your project.

Setting Up Megascans

Before you can start using Megascans in Mari, you need to set them up. Here is how you can do it:

1. Save your Megascans **.zip** files to a repository on your machine.
2. Unzip the Megascans **.zip** files.
3. In the **Texture Sets** palette, in the **Megascans** tab, click **Set Path** to add the location of your Megascans. Click **Choose**.

   OR

   In Mari, navigate to **Edit** > **Preferences** > **Paths** > **Megascans**. In the **Search Path** field, add the path to where your Megascans are stored. Click **OK**.
4. The Megascans assets display in the **Megascans** tab.

> **Note:** If you add new assets to the Megascans' repository on your machine, click **Refresh** to display the new assets in the **Megascans** tab.

Using Megascans

Once you have loaded your Megascans, you can start using them. In the **Texture Sets** palette, in the **Megascans** tab, use the slider to zoom in or out for a better preview of your Megascans. Select whether you want to display all Megascans assets, the atlases, or surfaces only. You can also select which Megascans category to display using the **Filters** dropdown. The more filter options you add, the more assets display in the tab.

Once you've decided which Megascans to use, click and hold on a Megascans preview and drag-and-drop it in the **Image Manager** palette, or right-click on an asset and select **Add to the Image Manager**. This adds the Megascans assets to a new tab. If you want to add Megascans to the **Image Manager**'s current tab, click and hold on a Megascans preview, press **Ctrl**+**shift** then drag-and-drop it in the **Image Manager** palette. This adds the Megascans asset to the current tab.

Tip: You can add multiple Megascans assets at once to the **Image Manager** palette.

. See the Texture Sets Palette section in the *Mari Reference Guide* for more information.

Once imported to the **Image Manager** palette, you can use Megascans as you would use images. See Managing Images for more information.

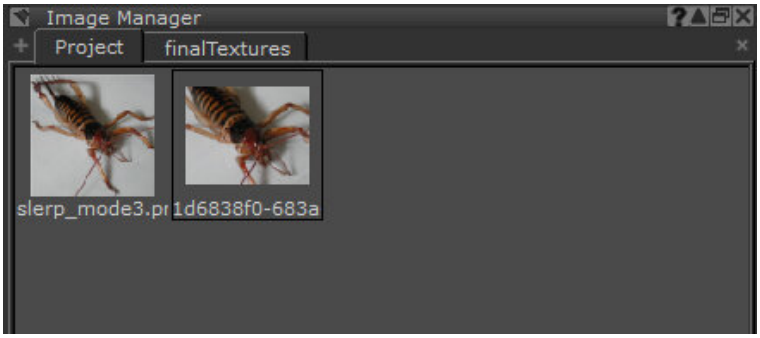
Cropping and Filtering Images

1. To create a new image by cropping an image from the **Image Manager**, click to open the image preview window.

Tip: Clicking on the image in this window sets the foreground color to the color at the spot you clicked.

2. Select the crop mode that you want to use. You can select from the following options:
   - **Arbitrary** - click and drag on an arbitrary area of the image.
   - **Fixed** - enter a fixed size for the crop box (in pixels) and drag it to the area you want to crop.
   - **Aspect** - enter a fixed aspect ratio for the crop box (in pixels) and drag it over the area you want to crop.
3. Now click and drag to select the area to crop, select the **Output Size** and click .

The selected area appears as a new image in the **Image Manager**.



Tip: Mari saves a cropped image as part of the project, not as a separate file on disk - unless you specifically save it with a new filename.

4. To apply a filter to an image, click to select it in the **Image Manager** and, from the **Filters** menu, select the filter to apply.

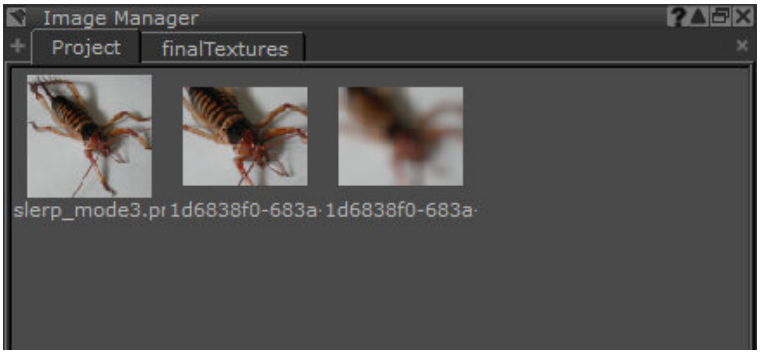
   The **Apply** dialog box displays.

5. Make sure that **Apply To** is set to **Selected Images**.

Tip: For details of the filters available and how to apply them, see Available Filters.

6. Set filter options, and click **Apply**.

   Mari applies the filter to the image in the **Image Manager**.



Tip: You can filter multiple images at once. Mari applies the same filter to all the selected images. Filtering an image in the **Image Manager** breaks the connection between the image and the original file. If you want a copy of your updated image on disk, save it with a new filename.

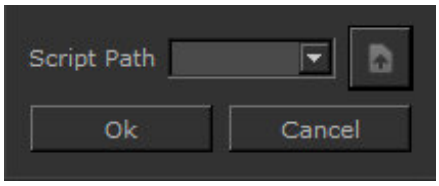
Running Scripts on Images

The **Image Manager** has a right-click option to run scripts on images. This lets you run custom scripts (for example, a Nuke or ImageMagick transform) on the images.

To run a script on an image:

1. Right-click on the image and select **Run Script** from the dropdown menu.

   The **Run Image Script** dialog box displays.



2. Type in the script command or select a script from the dropdown list. The list holds the last 10 scripts that were run.

   Use the $FILENAME variable for the name of the image file.

Tip: Mari runs the script command and then loads the $FILENAME file. So all commands must run in place on $FILENAME.

Mari runs the script, then loads the resulting image into the **Image Manager**. Initially, the previous version of the image also appears in the **Image Manager**.

Tip: You can run scripts on multiple images at once.

The initial image to the left, and the modified image to the right.

Run some basic scripts on your images. For example, try:

- **/usr/bin/convert $FILENAME -flop $FILENAME** - to flip the image horizontally.
- **/usr/bin/convert $FILENAME -blur 10 $FILENAME** - to apply a slight blur.

Using 2D Paint Mode

2D Paint mode enables you to paint and edit 2D images, which then can be easily integrated into the main project.

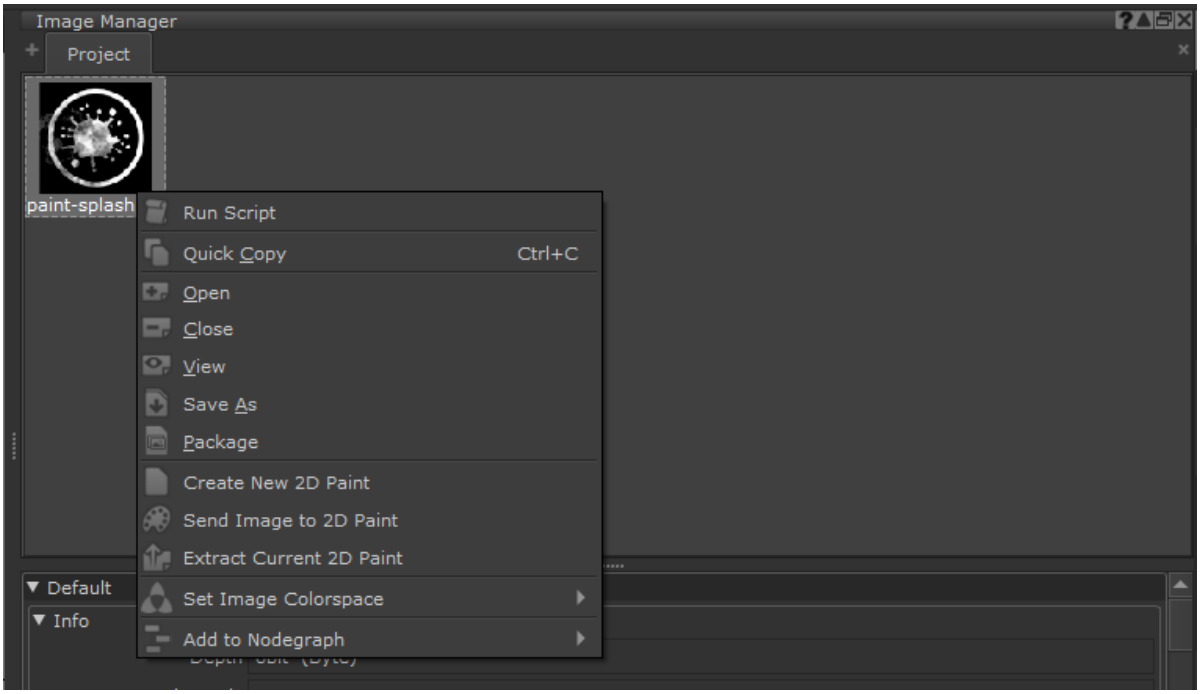
Using this tool, you can paint, flip, rotate, and scale patterns to create different texture patterns and use them as masks.

2D and 3D Node graphs and Layer stacks can be linked so that updating the 2D image automatically updates the texture in the 3D Project. To do this, you'll need to use the Image Write Node.

Access 2D Paint Mode

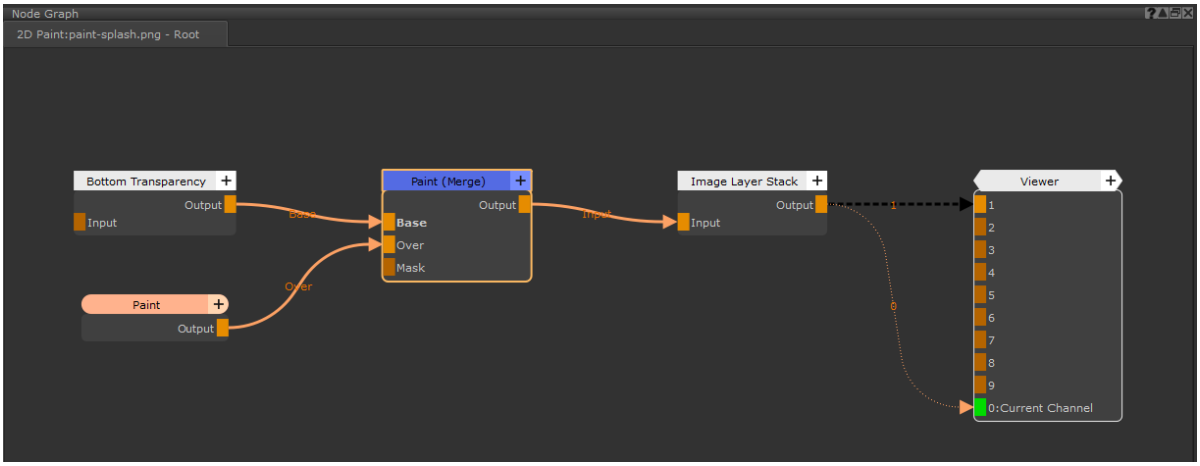
1. Open from **Image Manager**

In **Image Manager**, select an image, right click and select **Send Image to 2D Paint**.



This opens your image in the **2D** pane in the main canvas.

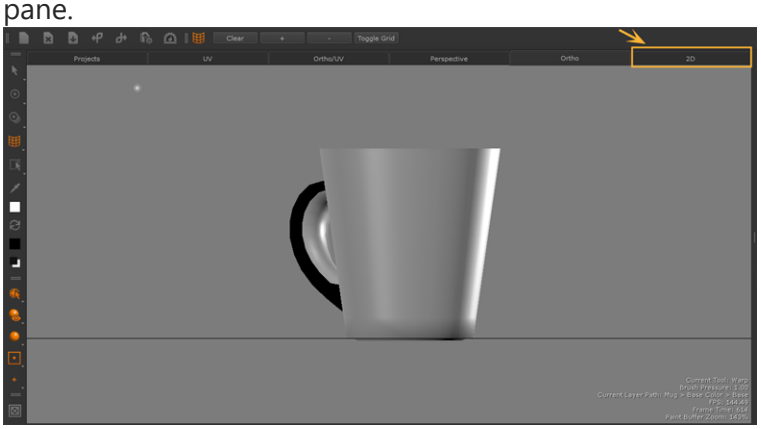
Mari will import a new object and import the image as a Paint Node. The node graph will show Bottom Transparency + Paint node (with the image imported) merged and connected to a channel node.



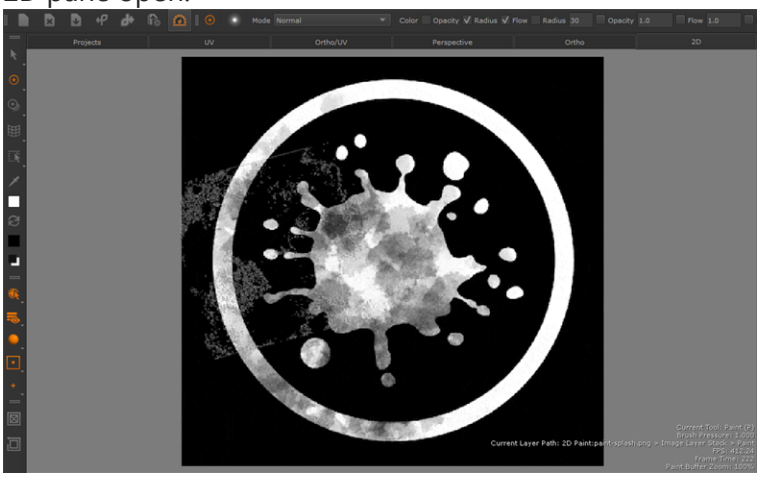
OR

2. 2D pane

Opening the **2D** pane will open an empty image ready for you to paint. Or if you already have an image opened in the 2D mode, you can re-open 2D mode easily by clicking on the **2D** pane, next to the Ortho pane.



2D pane open:



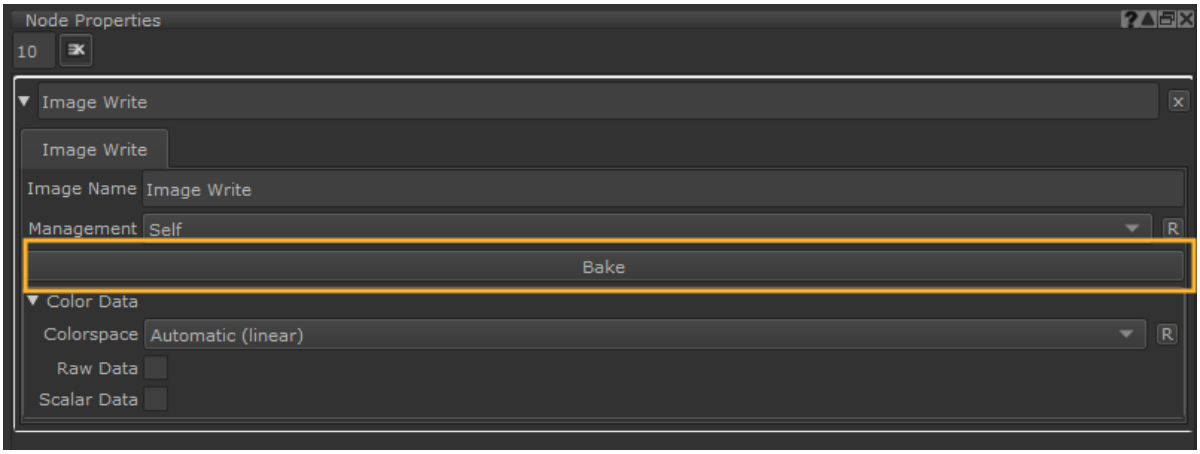
Note: Adding 2D images will also add them to the 2D tab on the Objects Palette. You can import and export the node graph as a .mrimg file from there, by using the right-click options **Export 2D Paint** and **Import 2D Paint**, which will allow you to save the 2D node tree / layer stack to disk separate to the main Mari cache.

Paint your 2D image

In 2D Paint Mode, you'll find the same painting tools as Mari's 3D canvas. These appear in the usual places in the UI. Any unavailable options that only apply to 3D are greyed out.

Update your image on your 3D object

Use an Image Write Node in your 2D node graph to export an image to the Image Manager. This will automatically update the image on your 3D object when you **Bake** it, if the image is used in any procedural nodes in 3D mode.



Save/Extract your image

By right-clicking your image in the Image Manager, and selecting **Extract Current 2D Paint**, this saves a copy of your last edited image to the Image Manager, which can then be used elsewhere.

Managing Colors

Mari lets you control both how colors display, and how you select them. You can use the color management functionality provided by the **View Transform** toolbar to specify colorspaces for different aspects of your project, apply various color profiles (preset or custom), and apply LUTs (look-up tables).

Mari always has a **Colors** palette that allows you to set the “foreground” and “background” colors that Mari uses, and the colorspace of the color pickers and swatches displayed in the project. The foreground and background colors are also displayed in the **Tools** toolbar. By default, painting tools use the foreground color.

Note: Colorspaces can be configured discretely for the various aspects of Mari. The colorspace displayed in the **Colors** palette, only affects color pickers and swatches but you can change this colorspace and others in the **Project Settings** dialog, accessible from **File** > **Settings**.

For picking colors, Mari provides:

- a standard **Eye Dropper** tool.
- in the **Colors** palette, the **Analyzer** tab, which supplements the **Eye Dropper** with a range of colors you've selected, showing their average, minimum, and maximum color values.
- the **Colors** palette, which lets you pick colors interactively from a selection of pickers (including HSV, grayscale, blends, HDR, and from images) or just by entering specific color values.
- the ability to select colors from images in the **Image Manager** or a shelf.

You can also save colors in the shelf, and access these from inside the Add Paint Node / Add Bake Point / Add Channel dialogs using the **Color Presets** shelf. See Adding a Color to a Shelf.

Note: Both the **Eye Dropper** tool and **Analyzer** tab in the **Colors** palette can pick color values from HDR images, as long as the **Paint Buffer** hasn't been clamped.

Tip: You can also change the color depth of a channel if you need to. See Channels for details.

Warning: When working on an 8-bit (byte) channel that is in a non-linear colorspace, you need to set the virtual texture's type and paint buffer's color depth to 16-bit (half) or 32-bit (float) to minimize quantization errors. This only applies to the viewer and not to exported textures.
To set the virtual texture's type, refer to the *Mari Reference Guide* in the *Mari Preferences Dialog* section. To set the paint buffer's color depth, refer to the *Mari User Guide* in the *Configure the Paint Buffer* section.

Color Remapping

When working with color management, you may want to fine tune the color grading on your textures to get your desired look. Mari has a selection of Filter nodes, such as the Gamma, Saturation, and Contrast node that enable you to have fine control over your colors. See more at Filter Nodes.

Alternatively, if there are multiple aspects you want to tweak in your colors, Mari's **Color Remap** node provides color grading without having to jump between different nodes to make sure they're all matching.

Within the Color Remap node, you can make changes to the hue and saturation, balance tricky lighting using exposure and gamma, and add eye-catching depth using advanced contrast controls. See more at Color Remap node.

About the View Transform Toolbar

The **View Transform** toolbar is provided as an efficient way to enable or disable color management and set LUTs to the painting displayed on your screen. This functionality is used to be available from within the **Color Manager (View Transform)** palette (under the preset **View Transform** color profile). This was removed in favor of the toolbar, which is conveniently placed to make quick adjustments while painting.

The **View Transform** toolbar utilizes the OpenColorIO standard originally developed by Sony Pictures Imageworks. For a more comprehensive explanation of OpenColorIO, including LUT generation and suggested workflow strategies, please visit http://opencolorio.org/.

You can quickly toggle view transform using the Toggle View Transform button on the toolbar, as well as conveniently load LUT files using the select LUT button. There are also dropdown menus available for applying colorspace conversions.

Note: To turn off color management fully, navigate to **File** > **Settings** > **Colorspace** and disable **Color Management Enabled**.

Note: For more information see View Transform Toolbar.

Picking Colors from Paint on the Model

To pick a single color from the paint already on the model:

1. Select the **Eye Dropper** tool and click on the model.
   Mari sets the **Foreground** color to the color at the point you clicked.

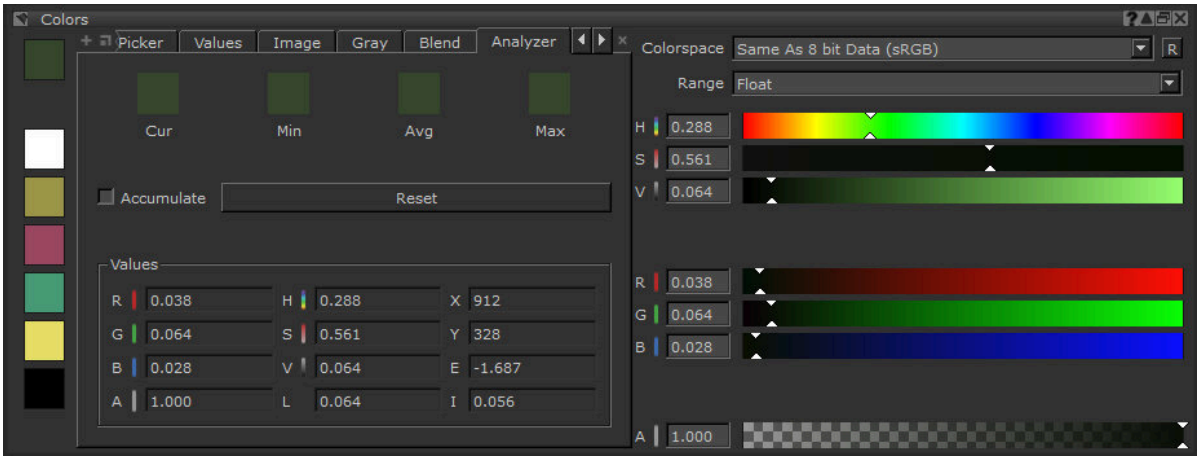
Tip: Mari sets the color to what's actually on the model's surface where you clicked. This may not be exactly the color you see on screen. For example, if you use full lighting and the area is in shade, it doesn't appear darker than it actually is.

Video: Watch a demo on how to select the foreground and background colors.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

2. Open the **Colors** palette:
   - right-click on the toolbar, and select **Colors** from the dropdown menu, or
   - from the **View** menu, select **Palettes** > **Colors**.

   The **Colors** palette opens.
3. Click the **Analyzer** tab.

   The **Analyzer** tab displays details of the currently-selected color.



The **Values** section displays the color's exact numeric value, in the selected **Range**.

Note: Picking values for HDR images outside of the range as defined by the color picking OCIO colorspace, may result in clamped values.

Video: Watch a quick demo.

Tip: If you are analyzing pixels from an HDR image and want to maintain the correct color values, ensure that you have not ticked the **Clamp** checkbox in the **Painting** palette, and that you are painting in a 16- or 32-bit channel.

4. Click the **Accumulate** checkbox.

   With this option enabled, Mari keeps track of the colors you select. As you click on the model, the **Colors** fields update to display the **Current** color selected, plus the **Minimum**, **Average**, and **Maximum** color values from the tracked series of clicks. (If you drag across the model, Mari still accumulates colors, as if it were a series of clicks.) You can then click on any of these swatches to set the foreground color to that value.
5. Click **Reset** to clear the buffer and start again.

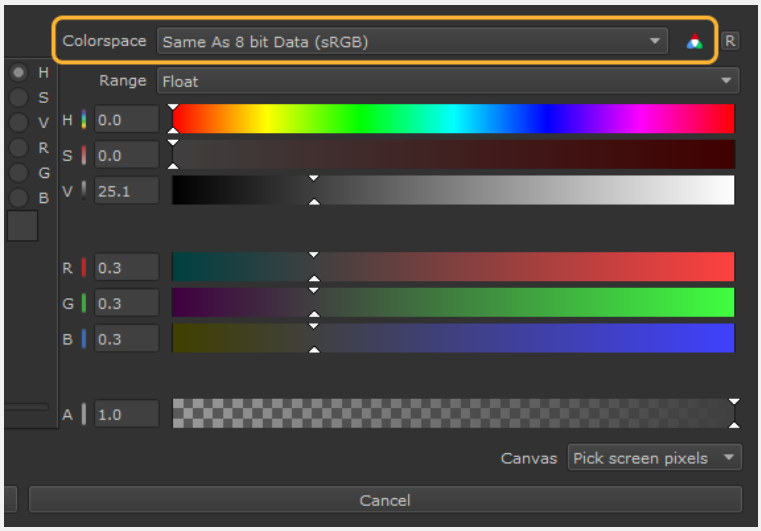
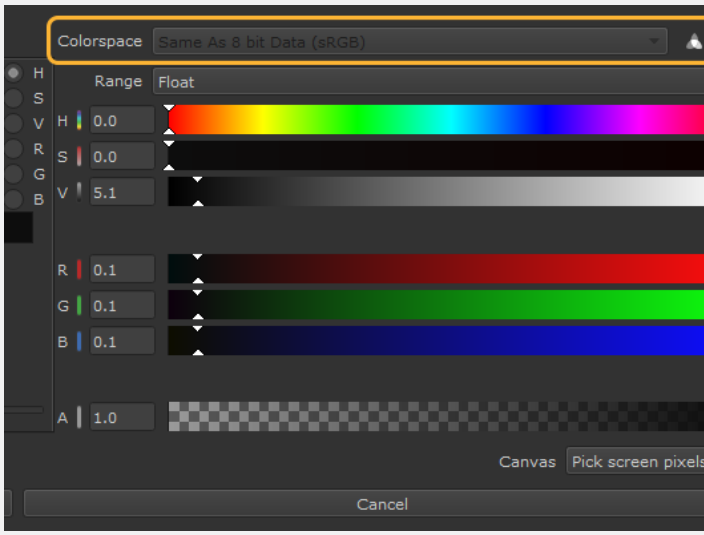
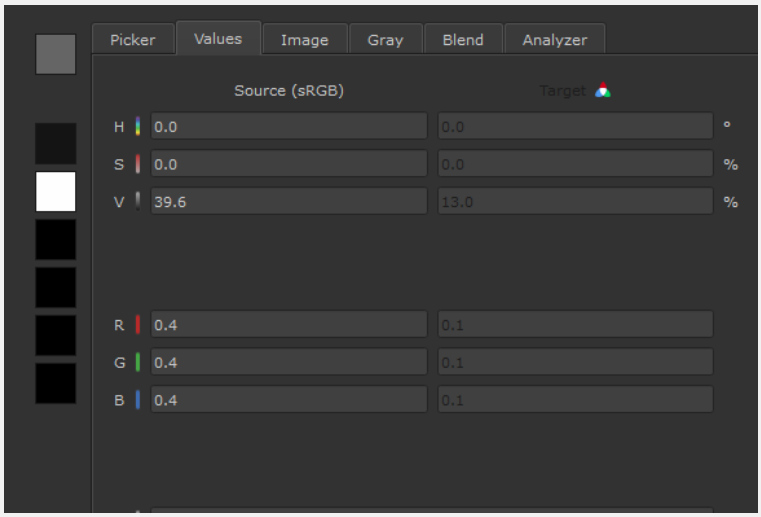
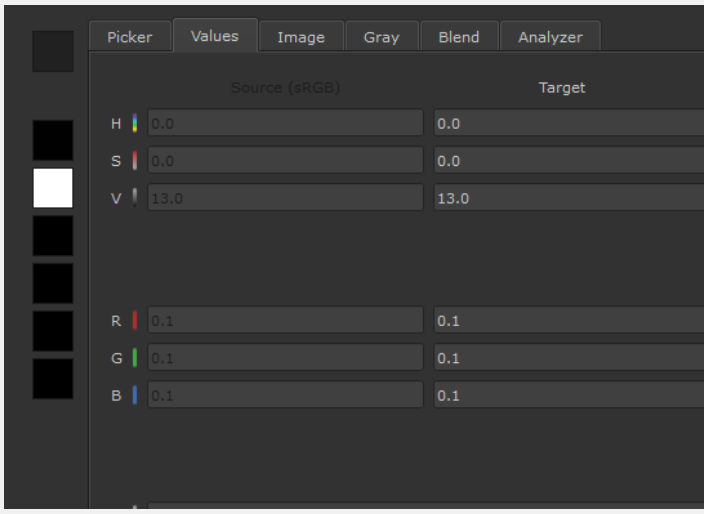
Scalar and Color Picking Modes

The color picker also allows you to pick non-color or scalar values. The scalar color picker works in the same way that the perceptual color picker does but instead of picking values in the colorspace assigned to the Color Picking role, raw RGB values are picked and color management is not applied.

The color picker samples values based on the colorspace of the active paint target. This means that if a channel, or node, you're currently painting on is scalar, you'll pick scalar data, while if the active paint target is color data, you'll pick color managed sRGB values. Your active paint target may not always be the channel or node you are viewing, so it's important to be aware of what is and isn't being edited.

Note: for more information on the differences between Color and Scalar data, see Color and Scalar Data

The color picker displays indicators when you choose either Color or Scalar data.

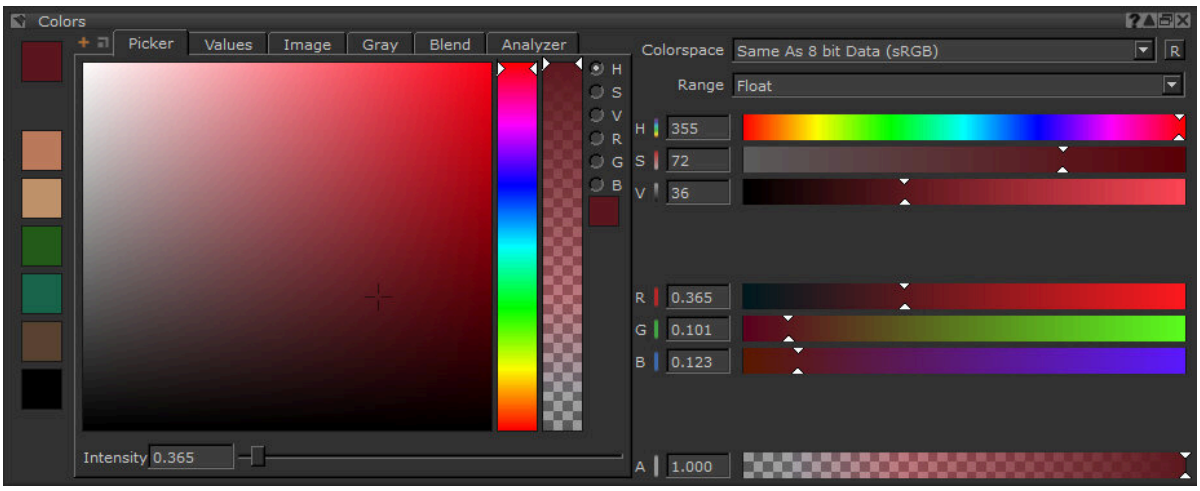
Color Data	Scalar Data
	
The colorspace dropdown is active and displays a colorspace icon	The active paint target is Scalar, so the colorspace dropdown is disabled, and the icon next to it is greyscale.
	
Values in the target column differ to the values in the source column, as the chosen color is being color managed depending on the colorspace applied to the Color Picking role. Target is also disabled, and cannot be edited.	The values in both columns match, as raw data is being chosen and no color management is being applied. The Source column is also disabled, and cannot be edited.

However, you can manually force what type of data you are picking within the color picker. This is useful for layer-based workflows or projects that contain no paint nodes, where you are unable to assign an active painting target.

Selecting Colors from a Picker or Image

1. Open the **Colors** palette:
   - right-click on the toolbar, and select **Colors** from dropdown menu, or
   - from the **View** menu, select **Palettes** > **Colors**.

   The **Colors** palette displays colors and color pickers.

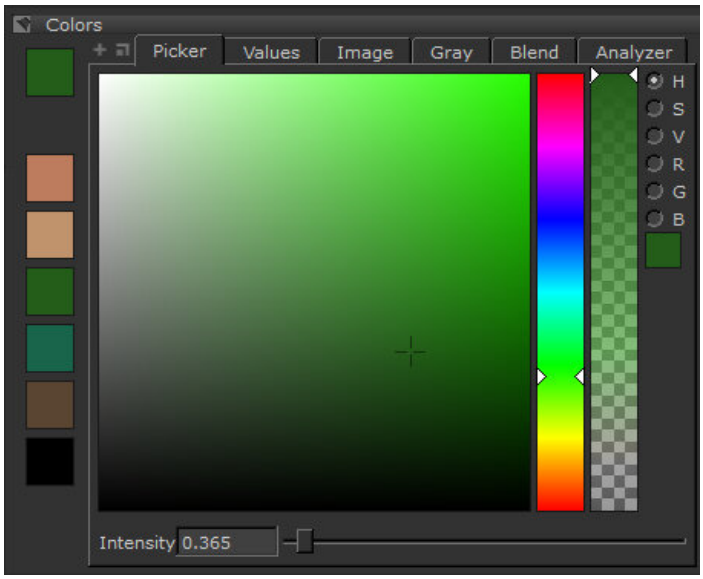


2. Select a color:
   - manually set the numeric values (either type them into the text fieldes or move the sliders) for the various color attributes - red, green, blue, alpha; or hue, saturation, and value, or
   - click in a color field in the tabbed area.

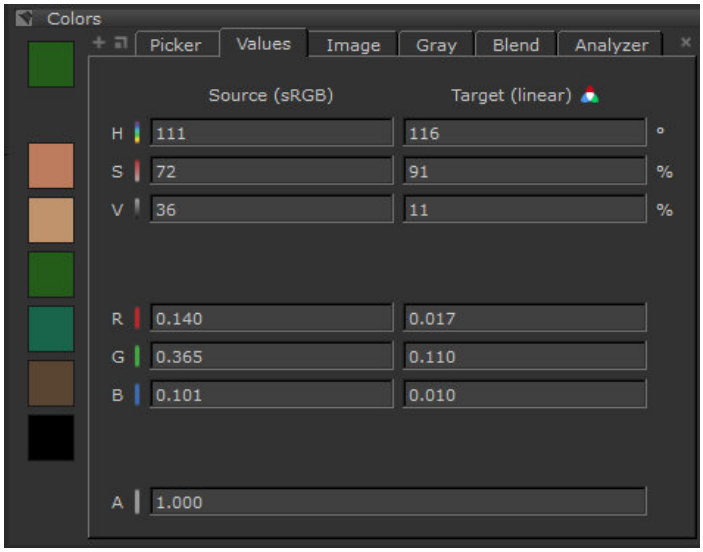
   The control dynamically updates as you change the settings.

> **Note:** Picking values for HDR images outside of the range as defined by the color picking OCIO colorspace, may result in clamped values.

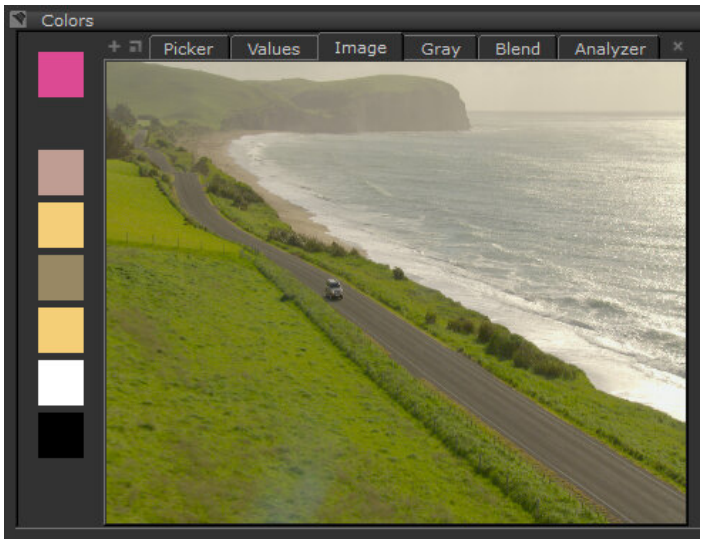
3. Click to navigate through the tabs available. These include:
   - **Picker** - to select between **Hue**, **Saturation**, **Values**, **Red**, **Green**, or **Blue** in the Color field and the Color slider. You can also set the intensity.



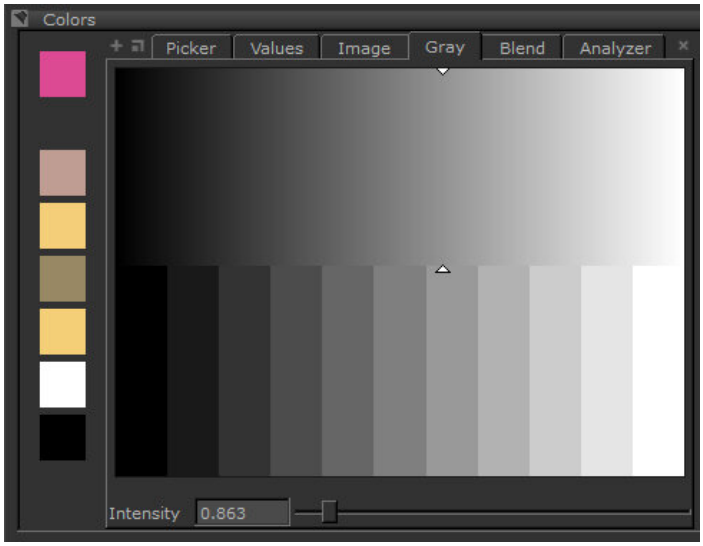
• **Values** - to pick from Byte (8-Bit) or Float (32-Bit) color values.



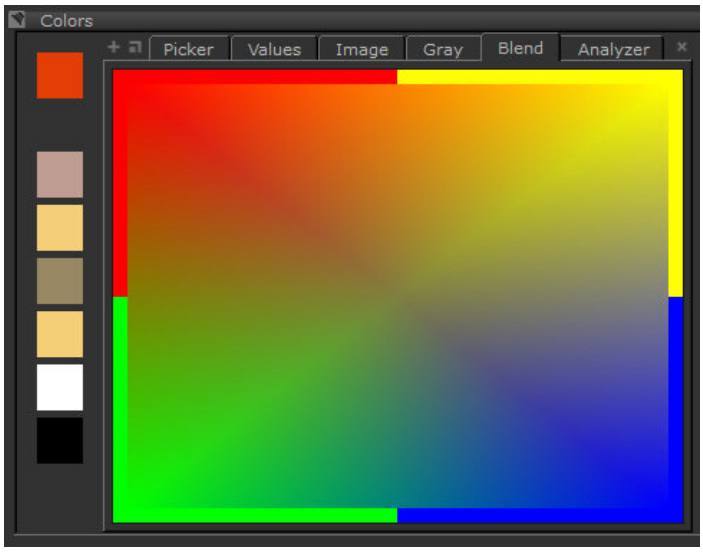
• **Image** - to pick a color from an image (right-click to load an image, or drag an image from the **Image Manager** palette).



- **Gray** - to select grayscale values (including preset gray settings, at the bottom, in steps from black to white).



- **Blend** - to pick from a field blending four colors. Click any of the color swatches (foreground color or recently used colors) and drag-and-drop them to any corners of the color field. Then click in the color field to pick from the blend.



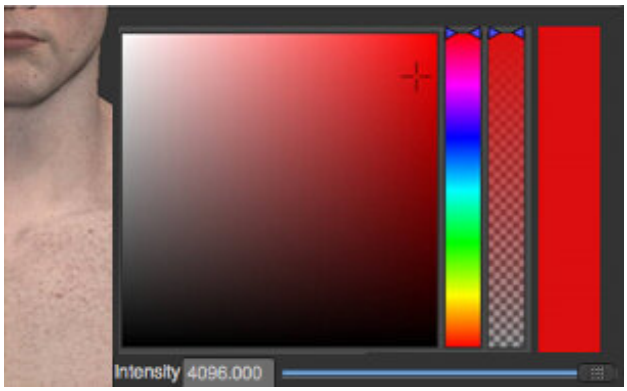
Tip: By default, Mari opens the following tabs: **Picker**, **Values**, **Image**, **Gray, Blend, and Analyzer**. You can add additional color tabs: **Color History**, **Colorspace**, and **Sliders**.

- To open other tabs, click .
- To open a tab as a floating window, click .
- You can also pick colors from images by double- clicking on them.

4. To display the Colors quick palette when painting, press and hold **J**.

Video: Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

The Colors quick palette displays under your mouse cursor. The bar at right shows the selected color (without a border - so you can compare directly with what's already painted). You can also change the intensity of the color, if you are trying to match the color intensity of an HDR image.



Tip: If you intend to paint on an HDR texture with a color from the color picker, ensure that you have not ticked the **Clamp** checkbox in the **Painting** palette, and that you are painting in a 16- or 32-bit channel.

5. To pick colors from an image directly from the **Image Manager** or a shelf, double-click on the image. When the image window opens, click within the image to set the foreground color.

Tip: See also Storing Resources Using Shelves for instructions on making a custom shelf with small icons of selected color swatches.

Try the steps below to select new colors:

- Pick a color from the **Colors** palette. Notice as you drag the cursor across a field, or slide a slider, the **Foreground** swatch changes and the RGB values change.
- Load an image into the **Image** tab in the **Colors** palette, and pick a color from it.
- Create a custom, floating **Blend** window with different colors than the defaults.

Color Management

Mari uses OpenColorIO for color management. All of the colorspaces in Mari, whether those shipped with the application or custom colorspaces loaded during project creation, are defined in OCIO config files.

Depending on the OCIO config file that you are working with, there are a number of colorspace options and roles (aliases to colorspaces) that you can set in Mari. There are also **Automatic** options, which change

depending on what file type you are working with. When you've selected the **Automatic** option the colorspace that Mari has set for it is listed in brackets.

Changes to a project's colorspace cannot be undone in Mari. You can change the colorspace back to its previous setting manually or use the reset button to restore it to the default, but you cannot undo any colorspace changes in the **History View** palette.

Note: Refer to the OpenColorIO website at http://opencolorio.org/ for additional information on OCIO colorspaces and roles.

Article: Read Q100362: Optimising your OCIO config for GPU Accelerated Color Management if you have your own custom OCIO configurations.

Colorspaces

The colorspaces are set in the OCIO config file. Mari looks to this file for the colorspaces that are listed and provides these options in all the Colorspace dropdowns available. Depending on which of the default config files that you select for your project, Mari offers a number of default colorspaces.

Mari ships with four config files: **aces**, **nuke-default**, **spi-anim**, and **spi-vfx**. There is also a **Custom** option that allows you to choose your own config file, from a designated place on your machine or network. If you set up a **Custom** config, Mari references this for colorspaces and roles, just as it would a shipped config file. However, these must be set up properly, with colorspaces and roles defined according to the OCIO standard, in order to work correctly.

Though there are four config files that are shipped with Mari, this section only covers the **nuke-default** colorspaces. However, brief descriptions of the other config files are listed below:

- The **aces** config file is a color profile with a color image encoding system proposed by the Academy of Motion Picture Arts and Sciences that allows for a fully encompassing color accurate workflow, with "seamless interchange of high quality motion picture images regardless of source".
- The **spi-anim** OCIO config file is a color profile that is in use at Sony Pictures Imageworks, and is suitable for use on animated features.
- The **spi-vfx** OCIO config file is a color profile that is in use at Sony Pictures Imageworks, and is suitable for use on visual effects work.

The **nuke-default** OCIO config file corresponds to the default Nuke color configuration. In this way, if you are using both Mari and Nuke in your pipeline, you can ensure that the color profiles match across applications. The default colorspaces that are defined in the config file are:

- **linear**
- **rec709**
- **Gamma2.2**
- **ViperLog**
- **SLog**
- **sRGB**
- **Cineon**
- **Panalog**
- **AlexaV3LogC**
- **raw**
- **sRGBf**
- **Gamma1.8**
- **REDLog**
- **PLogLin**

Roles

The roles are also set in the OCIO config file. Mari looks to this file for the roles that are listed and provides these options in all the **Colorspace** dropdowns available. Roles are used by Mari to perform a task-specific color transform without requiring you to select a colorspace by name, and they appear in all of the colorspace-related dropdowns. Depending on which of the default config files that you select for your project, Mari offers a number of default roles.

Though there are four config files with defined roles that are shipped with Mari, including the **nuke-default** roles.

Mapping Roles in the OCIO Config File

The OCIO config file is used to assign colorspaces to the roles that are listed in the **Project Setting** dialog, (see the Project Settings Dialog). To do this, each setting has a unique role name for use in the OCIO config file.

For example, to set the **Color Monitor** colorspace, you would set the OCIO role name to mari_monitor and specify the preferred colorspace. For example, to set **Color Monitor** to sRGB, add the following entry in the roles section of the OCIO config:

```
mari_monitor: sRGB
```

This can also be done using a Python script, where a target enumerator is used to refer to the role. However, the addition of OCIO role names make the process much easier.

If no colorspace is given for a role in the OCIO config, then Mari will use a fallback role. For example, the fallback role for the **Color Monitor** (mari_monitor) setting is to use the color_picking role setting. If the fallback role is also unset, then Mari will handle the assignment automatically.

The table below lists the project settings, their matching OCIO role name, the fallback role, and the Python target enumerator.

Mari Color Target Enumerator	Label in Project Settings	OCIO Role Name	OCIO Role Fallback
COLORSPACE_TARGET_MONITOR	**Color Monitor**	mari_monitor	color_picking
COLORSPACE_TARGET_SCALAR_MONITOR	**Scalar Monitor**	mari_scalar_monitor	data
COLORSPACE_TARGET_COLOR_PICKER	**Color Picking**	mari_color_picker	
COLORSPACE_TARGET_WORKING	**Working**	mari_working	scene_linear
COLORSPACE_TARGET_BLENDING	**Blending**	mari_blending	
COLORSPACE_TARGET_INT8	**8-bit Color**	mari_int8	texture_paint
COLORSPACE_TARGET_INT16	**16-bit Integer Data**	mari_int16	texture_paint
COLOURSPACE_TARGET_SCALAR8	**8-bit Scalar**	mari_scalar8	data
COLORSPACE_TARGET_FLOAT	**16/32-bit Float Data**	mari_float	scene_linear

Article: Take a look at the Q100408: Reduced color management pipeline friction with the Mari specific OCIO roles article.

Nuke Default Roles

Note: These are not the default roles applied in the project settings - Mari uses its own roles as listed above.

The **nuke-default** OCIO config file provides the following default roles:

- **color_picking (sRGB)**
- **data (raw)**
- **reference (linear)**
- **color_timing (Cineon)**
- **default (raw)**
- **scene_linear (linear)**
- **compositing_log (Cineon)**
- **matte_paint (sRGB)**
- **texture_paint (sRGB)**

If you've set a role in a colorspace field during project creation or in the **Project Settings** dialog, it also affects anywhere in your project where the colorspace has been set to **Automatic**. For instance, if the **8 bit Data** colorspace field has been set to the **data** role, which by default uses the **raw** colorspace, then any 8-bit channel with a colorspace set to **Automatic**, uses a **raw** colorspace and appears in the dropdown field as **Automatic (raw)**.

Setting How Colors Are Displayed Onscreen

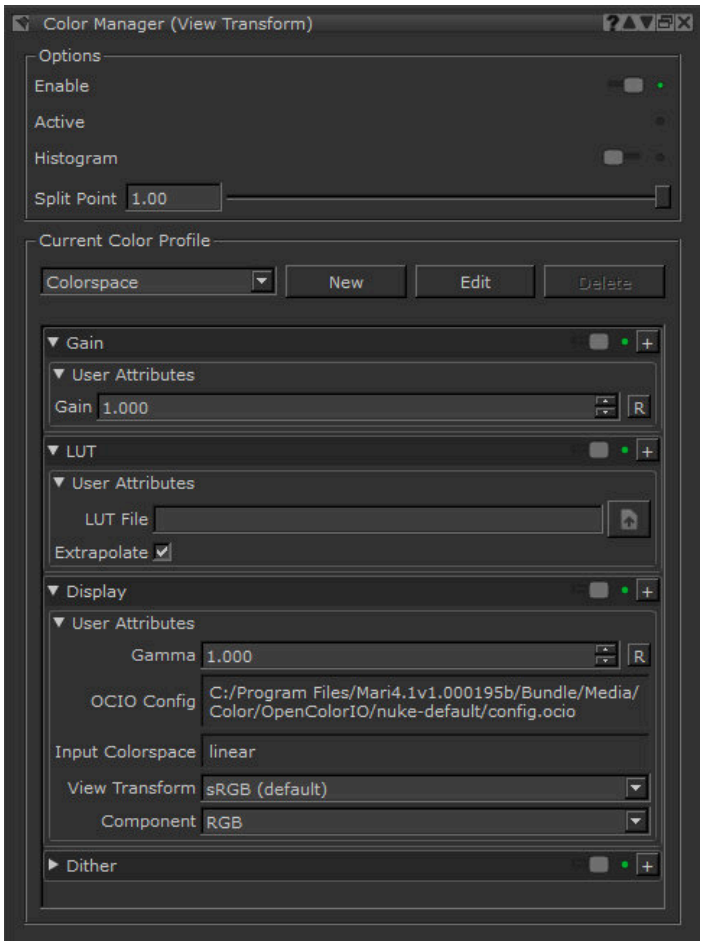
Mari sets how colors are displayed for each project separately. You can set this using either the **Color Manager (View Transform)** palette, the **View Transform** toolbar, or adjust the individual colorspaces of the project in the **Project Settings** dialog.

Note: The **Color Manager (View Transform)** palette is deprecated since Mari 4.0v1. You can display using an environment variable, see Environment Variables That Mari Recognizes.

Using the Color Manager (View Transform)

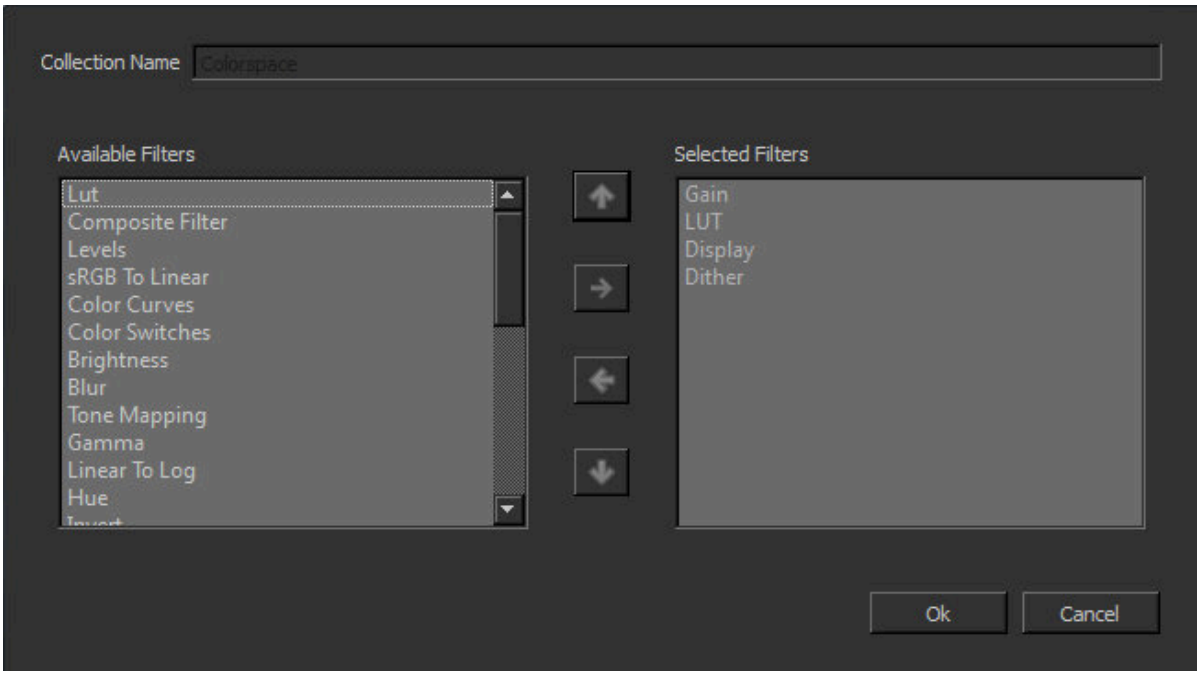
1. Open the **Color Manager (View Transform)** palette:
   - right-click on the toolbar, and select **Color Manager (View Transform)** from the dropdown menu, or
   - from the **View** menu, select **Palettes** > **Color Manager (View Transform)**.

   The **Color Manager (View Transform)** palette shows color display settings for the current project.



2. Under **Options**, select the main display options:
   - **Enable** - when selected, applies the color profile selected in the **Current Color Profile** section (see below).
   - **Histogram** - when selected, a histogram displays in the bottom left of the canvas, showing details of the color balance in the current channel.
   - **Split Point** - where (and whether) to split the canvas between color-corrected and uncorrected. The default is with the slider at the far right, at 1.00, showing the entire canvas with color correction. Moving the slider changes the split. For example, moving it halfway along (to 0.50) makes Mari display a split halfway across the screen, with color-corrected on the left and uncorrected on the right.
3. Under **Current Color Profile**, select a color profile from the list, or click **New** or **Edit** to create a new color profile or edit the current one.

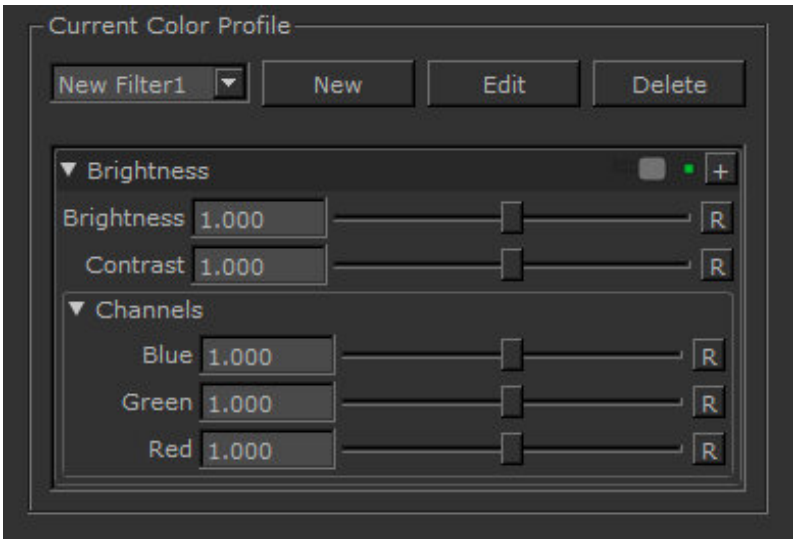
   The **Edit** dialog box displays the filters available for building up a color profile. For the **Colorspace** Color Profile, three filters - **Gain**, **LUT**, **Display**, and **Dither** - are selected by default and cannot be deleted.



4. Select filters to use from the **Available Filters** list, and click the right arrow button to move them to the **Selected Filters** list. Choose the order in which to apply the filters in the **Selected Filters** list, using the up and down arrow buttons. Then click **OK**.

   The **Current Color Profile** dialog box lists the selected filters.

5. To view and configure how Mari applies a selected filter, click to expand it, then set its properties.



6. You can delete color profiles from the list by selecting a profile and clicking **Delete**.

Tip: For more details on Mari's filters, see Filtering Images and Paint.

Using the View Transform Toolbar

1. If the **View Transform** toolbar is not enabled, click or select **Enable** in the **Color Manager (View Transform)** palette.

2. Click to select an OCIO configuration file.

> **Note:** This option is only available if color management is switched off. If you want to turn color management off, navigate to **File** > **Settings** > **Colorspace** and disable **Color Management Enabled**.

3. Set the colorspace of the current channel or image values from the **Input Colorspace** dropdown menu.
4. If you want to set up a LUT, click , choose **Select**, and browse to a file in the **Select LUT File** dialog.
5. Select the colorspace transform to apply to the scene using the **View Transform** dropdown menu.
6. View the individual channels for the scene using the **Component** dropdown menu.
7. Specify the amount of exposure adjustment applied before the view transform by either entering a multiplier (exposure value), dragging on the slider, or adjusting the F-Stop value in the **Gain** controls.
8. Finally, specify the amount of **Gamma** correction applied after the gain viewer process by entering a gamma level or by dragging the gamma slider.

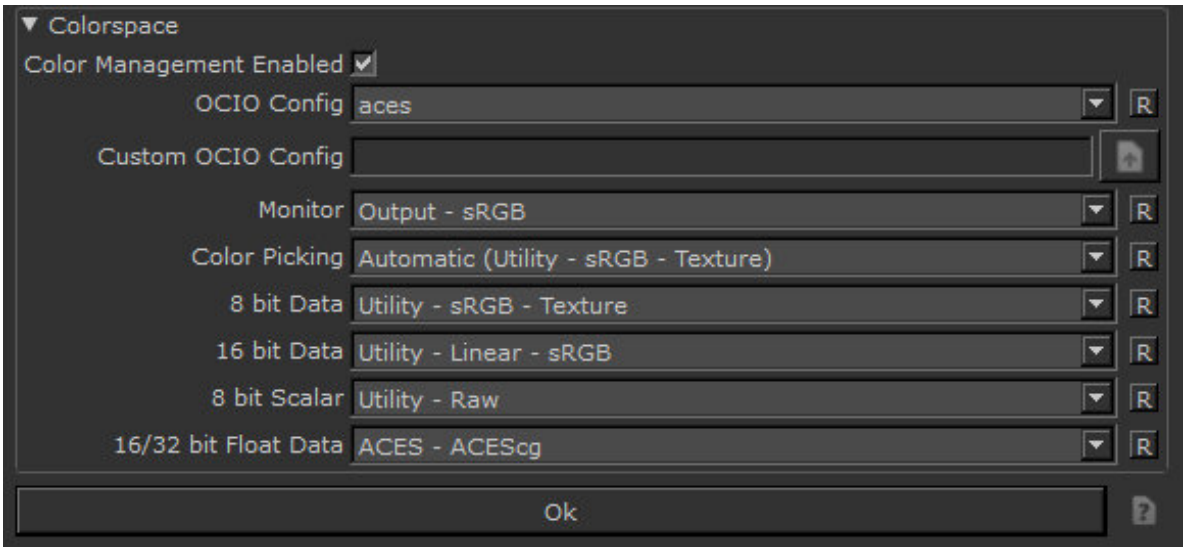
> **Note:** For more information, see the *View Transform Toolbar* .

Using the Project Settings

During project creation, you can set the colorspace you want to use for different aspects of your project. However, you might find that after you've created your project, you want to change these colorspace settings. You can do this in the **Project Settings**. Follow the steps below to adjust your settings:

1. Navigate to **File** > **Settings**.

   The **Project Settings** dialog displays.



2. Enable or disable the OpenColorIO color management using the **Color Management Enabled** checkbox.
3. Specify the **OCIO Config** file to use. The options given are shipped OpenColorIO configuration files that handle colorspace or user-defined configuration files (**Custom**) that you can set up yourself.
4. If you chose **Custom** for the **OCIO Config** field, you need to specify, or browse to, the configuration file that you want to use for this project.
5. Set the **Monitor** field to specify which is the default colorspace, that applies to thumbnails and UI elements. Though it can be whatever colorspace you want, it's recommended that this is the same colorspace as the one that you intend to use for the view transform.
6. Set the **Color Picking** field to specify which colorspace is used for all the color pickers and swatches in the project.
7. Set the **8 bit Data** field to specify which is the default colorspace for 8-bit (Byte) channels, as well as for reading and writing image files with 8-bit data.
8. Set the **16 bit Data** field to specify which is the default colorspace for reading and writing image files with 16-bit data.
9. Set the **8 bit Scalar** field to specify which is the default colorspace for masks, heights, normals, depths and, generally, any non-color image with 8-bit data. See Color Data and Scalar Data for more information.
10. Set the **16/32 bit Float Data** field to specify which is the default colorspace for 16- and 32-bit (Float) channels, as well as for reading and writing image files with floating point data.

Note: You don't need to specify all of these settings, either on project creation or while working on the project, as Mari automatically inserts default options for these fields.

Note: A 16-bit channel is in half format, which is floating-point and should take the default from the **16/32 Bit Float Data** setting, not the **16 Bit Data** option. The **16 Bit Data** option is for 16-bit integer formats and is currently only used in Mari for importing or exporting images in that format, such as some **.tiff** files. Mari currently doesn't support 16-bit integer channels.

Color Data and Scalar Data

Channels in Mari can either contain color data or scalar data. Color data holds the RGB values that define the color of the pixels in a texture. Scalar data on the other hand, defines properties of the texture such as shader inputs that define bumpiness, roughness or shininess. Essentially, color data exists to be seen, scalar data is used to calculate.

Mari allows you to switch between scalar and color channels in the **Channels Palette** (see the Channels Palette), and each channel can be viewed and customized. You normally need to apply a view transform to color data to adjust it according to your monitor's colorspace.

Scalar data's view transform defaults to raw (none) by which it is unchanged. This is because scalar data (unlike color data) is a computational resource and as such does not require color adjustment to be viewed.

Note: A scalar channel can still be viewed in Mari - think of this as a visualization of the data, drawn as if it was RGB. For example, roughness will be seen as grayscale; the darker the gray, the rougher the surface.

Both color and scalar channels have default view transform types that can be set in the in the **Project Settings Dialog** (see the *Project Settings Dialog*).

View transforms can be selected using the **View Transform Toolbar** (see the View Transform Toolbar). Each selection is remembered and it can be reset to the default using this toolbar.

Color Data

Color data in Mari consists of RGB values which describe the color of pixels in a texture. This is generally driven by a colorspace conversion to maintain a perceptually accurate result across different colorspaces and

data types. Color data is used when a channel needs to display a visual response based on a artist's exact picked color, specifically diffuse color, specular color, or emissive color.

Scalar Data

Scalar data holds raw mathematical values for processing textures as opposed to human perception of colors. Scalar data is used in shader input/channels and masking. For instance, scalar data is used to describe values of shader inputs such as the amount of bumpiness, roughness, shininess, and so on.

Scalar data is not affected by color conversions. It can relate to either coordinate or RGB values typically between 0 - 1 across three components. Scalar data is generally used for simple and compound masking, or to describe a mathematical value used to drive a shader input.

Scalar data can be perceived as grayscale or RGB.

Scalar Data: Grayscale

In the case of grayscale, when the scalar data's RGB values are the same (e.g. 0.3, 0.3, 0.3), this results in a shade of gray when viewed.

As a shader input/channel it is used to display an effect such as the shininess or reflectivity of a surface or opacity.

For masking, it can be used to create a grayscale mask, normally used to indicate surface opacity.

Scalar Data: RGB

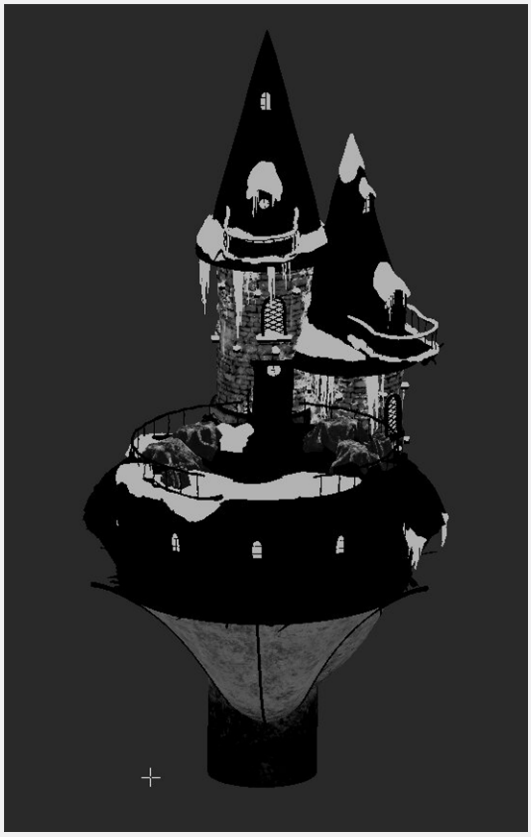
When the scalar data is RGB, the values are different for each component (unlike grayscale).

Note: RGB scalar data doesn't define true RGB color values, it simply uses the same data format (a triplet of numerical values). Even though the resulting RGB value can be perceived as color, it is simply an outcome of using the RGB components of a texel to store three mathematical values.

When using RGB scalar data for shader inputs, one example is to relate to separate coordinates in space. For world space, the value of the three components would be applied to an area in x,y,z coordinate space (for four components it would be x,y,z,w), specifically a normal or vector map. Another example is using the RGB to combine three grayscale scalar channels using their equivalent R, G, and B values to create per component mappings in a single channel.

When masking, RGB scalar data is used for co-ordinate information or channel masking, as opposed to gradient based masking. For example, to create a mask you could use your individual RGB channels to separate objects or areas of your mesh, or even individual shader inputs within your project.

Diffuse color channel affected by color management -
Scalar Data option is disabled.

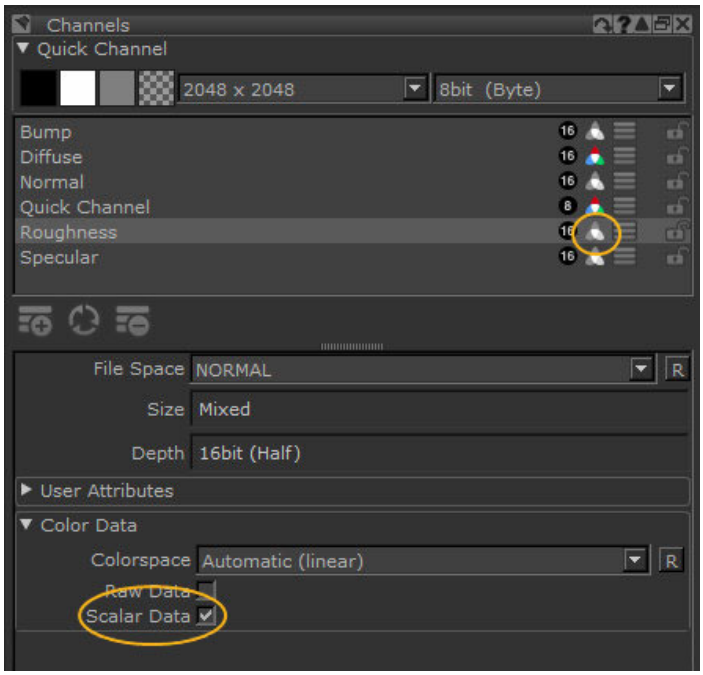


Specular data channel not affected by color management -
Scalar Data option is enabled.

Choosing Scalar Data

By default Mari always assumes data is color, unless specified as scalar.

Scalar data is used in the **Image Manager Palette** and **Channels Palette**, see the Image Manager Palette and Channels Palette. Scalar data can be enabled with the **Scalar Data** checkbox and in the **Channels** palette's channel list this icon / indicates whether the channel is tagged for scalar data.



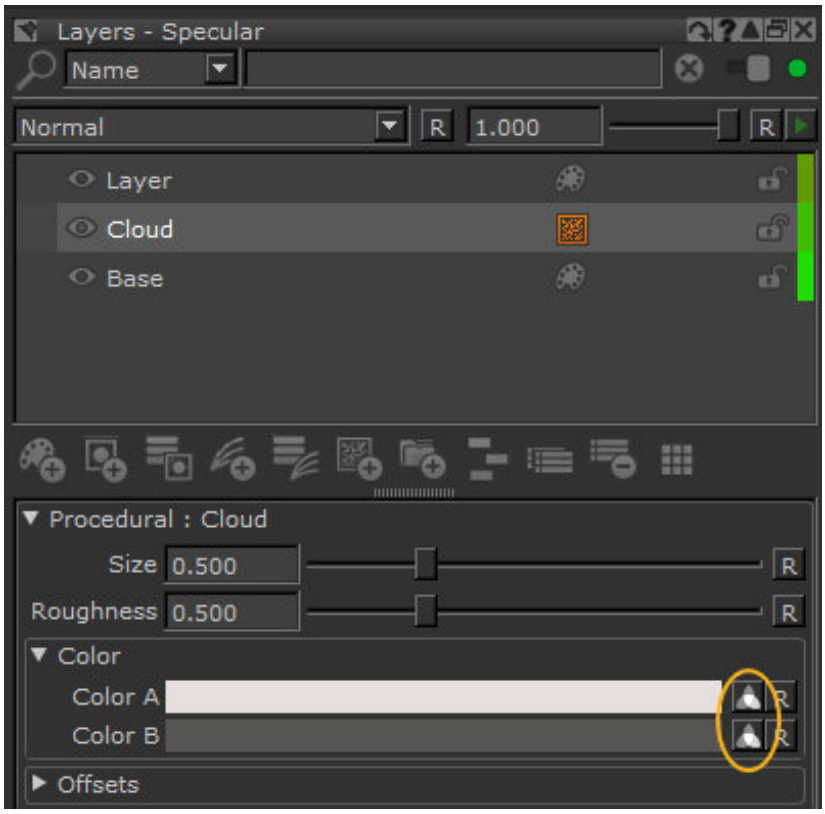
Scalar data is also used in the **View Transform Toolbar** (see the View Transform Toolbar), which is used to display a number of options for managing your monitor's colorspace. Color and scalar data are represented by the following icons:

- - The data is considered as color data and is being color managed.
- - The data is considered as non-color data, specifically scalar data, and has no color management applied.

View Transform toolbar

> **Note:** When viewing nodes that connect to scalar streams of Multi-Channel nodes, Material nodes, or Shader nodes, the View Transform switches automatically to scalar mode.

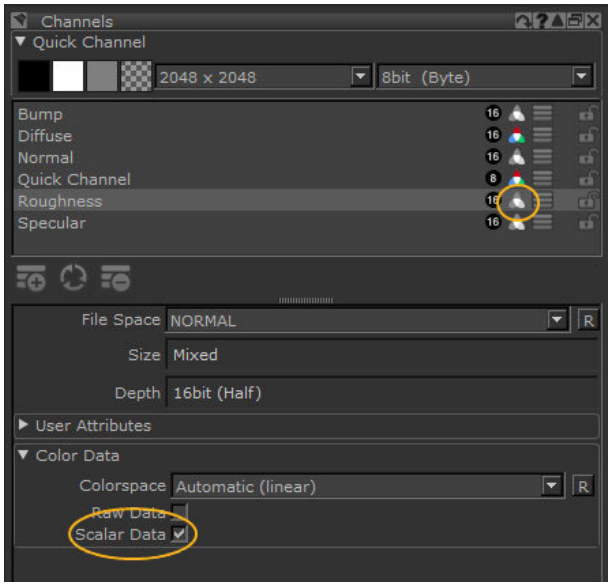
Some procedurals contain attributes that require a color for configuring their output (see Adding Adjustments and Procedurals and Layer Types and Examples). In these procedurals, the color controls contain a toggle that sets the context in which the procedural effect is to be used for Scalar or Color data. By default, Mari detects the channel's color data type and sets the toggle accordingly for layers created through the **Layers** palette. For Procedural nodes created through the Node graph, this color/scalar toggle defaults to color. Procedural nodes that are to be used within masks and scalar channels have to be set to scalar manually.



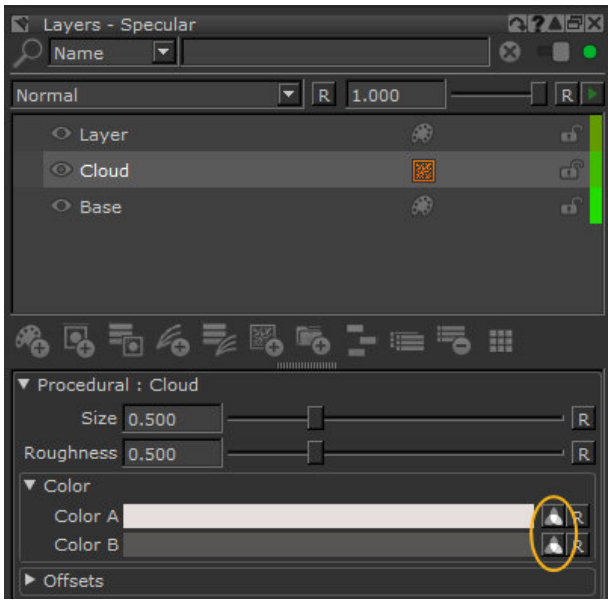
Procedural in the **Layers** palette

Using Scalar Data

Scalar is generally used to create effects such as bumpiness, roughness, shininess, reflectivity, masks, and so on. If you are painting scalar data, make sure to use scalar channels, source images, and procedurals. Some use examples below:



Channels palette - **Scalar Data** is enabled



Cloud Procedural - Scalar data toggle

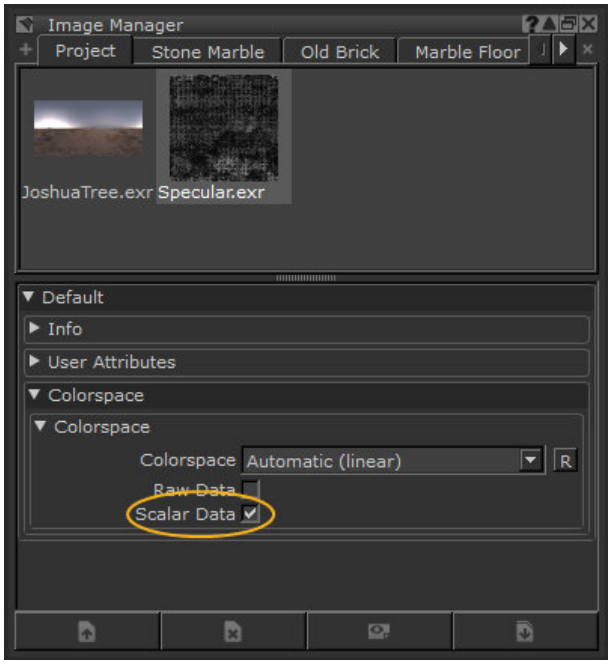
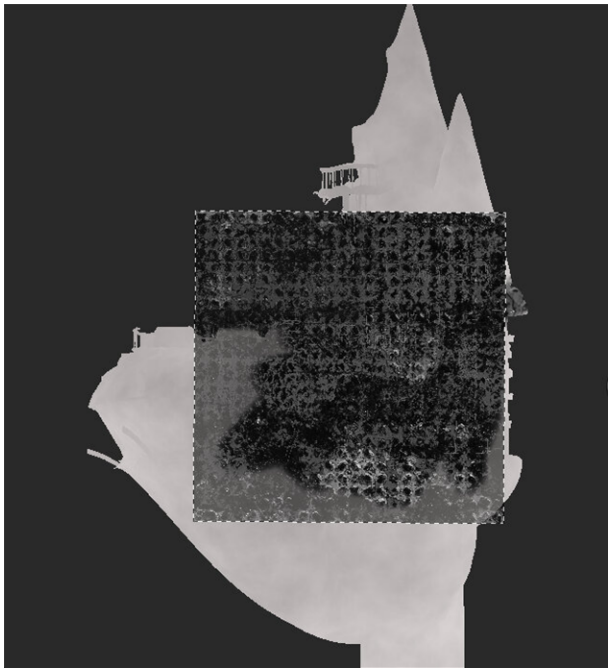


Image Manager palette - **Scalar Data** is enabled



Paint Through tool - Target channel has **Scalar Data**

Advanced Color Management

In addition to the basic controls for color management, Mari also has an advanced color management mode. This mode is for those users who feel confident about their color management abilities, and who know exactly what setting they want to change and why.

Warning: As changes to a project's colorspace, and color management in general, cannot be undone in Mari, it's imperative that you first consider whether you strictly need to operate in the advanced color management mode.

Tip: If you want to work in the advanced color management mode without risking damage to your project, you can make a copy of the config files to a location on your machine or network before making any changes to the original file.

Enabling Advanced Color Management

To enable the advanced color management mode:

Set the following environment variable: **MARI_COLORSPACE_USER_INTERFACE_MODE=2**

If the environment variable already exists, change it from **1** to **2**. Once the environment variable is set to **2**, the advanced color management options are active and exposed within Mari.

Note: Any changes you make to colorspace while advanced color management is enabled persist even after you have reverted back to the basic color management mode.

For more information on the **MARI_COLORSPACE_USER_INTERFACE_MODE** environment variable, refer to the *Mari User Guide* in the *Environment Variables That Mari Recognizes* section.

Advanced Color Management Options

Once you've enabled the advanced color management within Mari, there are a number of controls that are exposed in the UI, that are not otherwise visible in the basic color management mode. These controls vary, depending on where they are in the UI, but every place that colorspace is usually visible has additional options. Some of the options you may find are:

- **Working/Working Colorspace** - the colorspace for painting, lighting, applying filters, and similar operations.
- **OCIO Config** - a list of either user-defined configuration files (**Custom**) or shipped OpenColorIO configuration files that handle colorspace.
- **Custom OCIO Config** - if you chose **Custom** for the **OCIO Config** file choice, specify the path to the configuration file.
- **Output Colorspace** - the colorspace for the output.
- **Raw Data** - when enabled, prevents the data from being converted to another colorspace.

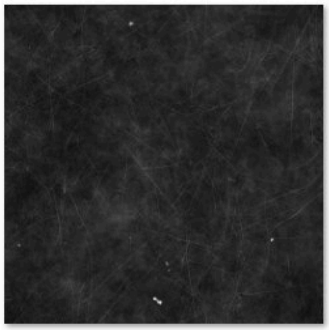
Though changes you make to these colorspace options while in the advanced color management mode persist whatever color management mode Mari is in currently, the options above are no longer visible when the basic mode is re-enabled.

Managing Materials in Mari

Materials are created from multiple texture files, procedurals, or plain colors that can be used to accurately represent how light interacts with real-world materials. This method of shading is called Physically Based Rendering (PBR). You can apply these materials to your assets for a more realistic result as well as establishing a base line look for your asset before adding more bespoke paint and crafting details.

Metal07_col.jpg

Metal07_nrm.jpg


Metal07_rgh.jpg

Texture files Thumbnails

Mari Material Thumbnails

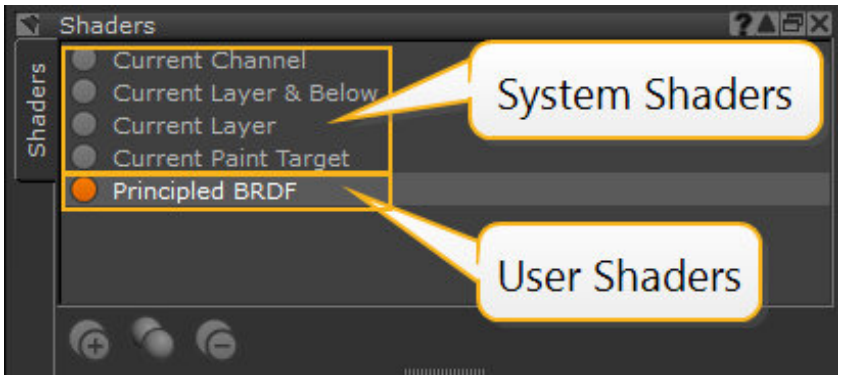
Materials are created based on shader models, which are fundamental to how material system works. Materials are shared across all channels making it easier for you to work.

Shader Models

Material nodes/layers are built from a single reference shader model. Shader models are 'metadata' that represents the inputs associated with a given shader and the properties of those inputs. Shader models are used by Mari when creating multi-channel layers and their associated nodes in order to generate the appropriate input/output streams and drive various user interfaces to match the requested shader. Shader models are derived from the User Shader.

There are two types of shaders in Mari:

- **User Shaders** - created from standalone shaders provided by vendors and Foundry. User shaders are used to show an approximation of the final rendered look of the actual shader.
- **System Shaders** - allow to view different elements of the scene separately, for example, you can look at different layers such as shininess, color, or scalar. It is less heavy to work with system shaders.



Shaders palette

Note: To create custom shader models you need to register your own custom shader, refer to the **Help** > **SDK** > **Custom Shader API** documentation for more information.

These are the shader models properties:

- **name** - full input name.
- **prettyName** - property's label that appears in the UI.
- **isScalar** - indicates whether an input represents scalar data (raw mathematical values) as opposed to color-managed data (RGB values). See Color Data and Scalar Data.
- **shortName** - abbreviated versions (4 characters maximum) of the full input names, which are used when the UI space is limited.

Note: The shader model **.xml** files are located in **C:\Program Files\Mari 8.0v1\Bundle\Media\Nodes\Simple\Lighting\Standalone**

Using Materials

You can create materials using the **Material Ingest Tool** (see the Material Ingest Tool Dialog section in the Mari Reference Guide) or the Node Graph, apply them using the **Layers** palette or the Node Graph, and store them using the **Shelf** palette, see Adding Materials to a Shelf.

- Refer to the Creating Materials with the Material Ingest Tool to create your materials.
- Refer to Working with Materials in the Layers Palette to use the **Layers** palette workflow.
- Refer to Working with Materials in the Node Graph to use a Node Graph workflow.
- Refer to **Adding Materials to a Shelf** and **Mari Preferences Dialog** to add custom thumbnails to your materials in the **Shelf**.

Example Material Design Project

Mari includes an example material design projection in the **Help** menu that contains a finished material setup for you to explore, backwards engineer, and tweak inside the Node Graph. This is great if you are new to material creation in Mari, and want to learn more about Node Graph material workflows. See Node Graph and Navigating Inside the Node Graph for more information on how to navigate the Node Graph.

The example material.

Creating the Example Material Design Project

1. Click the **Help** menu.
2. Click **Create Example Material Design Project**.
   Mari extracts and unzips the material design project inside the bundled Mari files.
3. In the **Projects** tab, double-click the newly created **MaterialDesign_** project.
   The material design project is opened.

Creating Materials with the Material Ingest Tool

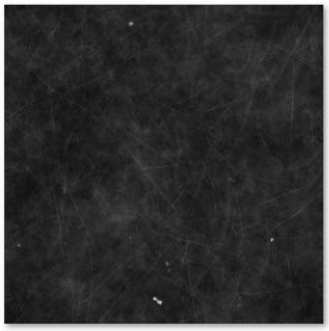
Materials are created from multiple texture files, procedurals or plain colors, which can be used to accurately represent how light interacts with real-world materials. This method of shading is called Physically Based Rendering (PBR). You can app ly these materials to your assets for a more realistic result as well as

establishing a base line look for your asset before adding more bespoke paint and crafting details. The Material Ingest Tool allows you to create materials from texture files. Once you acquired these texture files, you can start the ingestion.

Here are examples of texture files:

Metal07_col.jpg

Metal07_nrm.jpg



Metal07_rgh.jpg

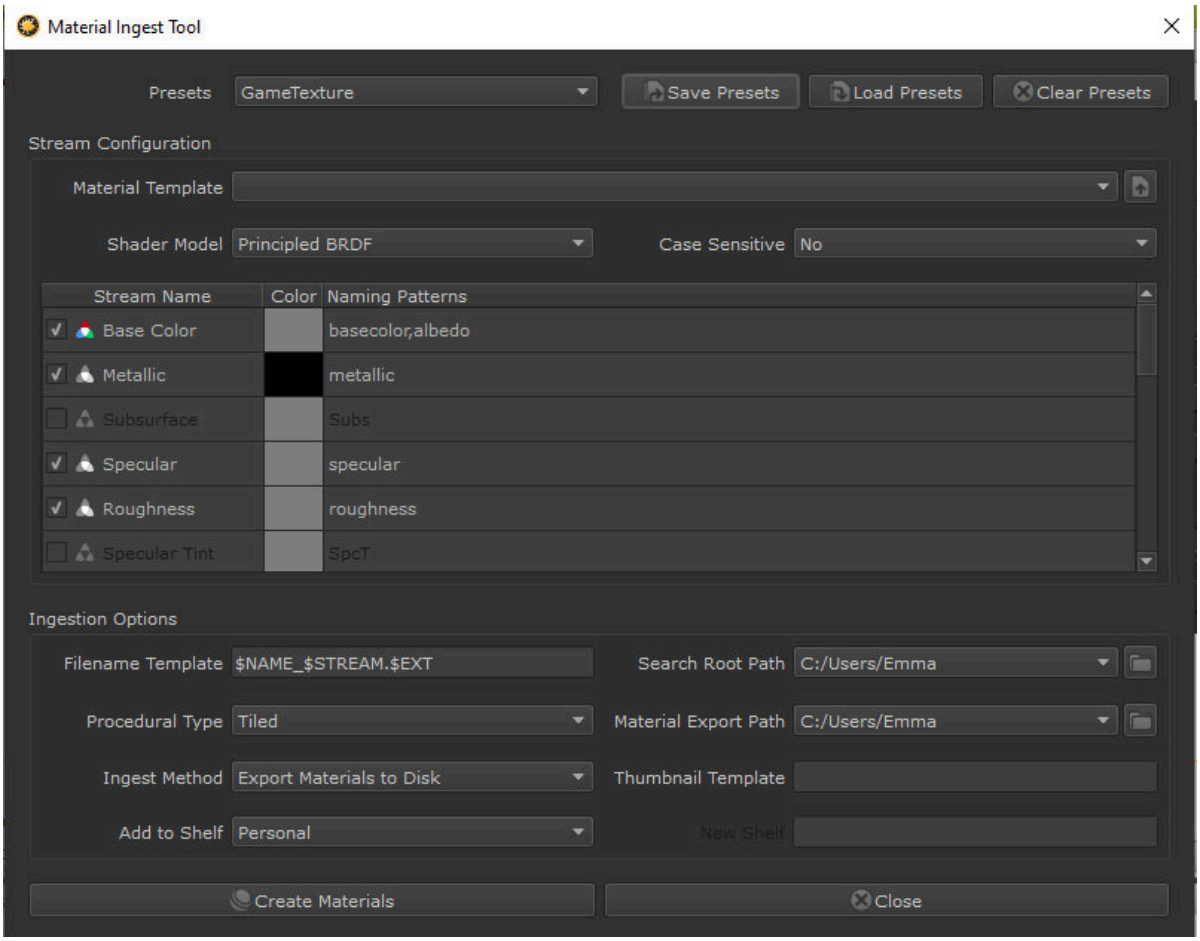
Notice that these files contain a material name, in this case **Metal07**. They also contain a stream name (shader input) to describe the intended target shader input, in this case: **col**, **nrm**, and **rgh**. They all have the same extension (**.jpg**). This naming pattern information is important to know before creating your materials so that you can fill in the material preset properties correctly.

Mari provides a set of predefined materials (presets), which are built for the Principled BRDF shader, in the **Shelf** palette, in the **Mari Materials** tab (see Storing Resources Using Shelves), but you can also create your own manually using the Node Graph (see Working with Materials in the Node Graph) or the Material Ingest Tool.

To create materials using the Material Ingest Tool:

1. Navigate to **Tools** and select **Material Ingest Tool**.

   This opens the **Material Ingest Tool Dialog**, see the Material Ingest Tool Dialog section in the *Mari Reference Guide*.



2. Select a preset from the **Presets** list according to which texture files you are using.

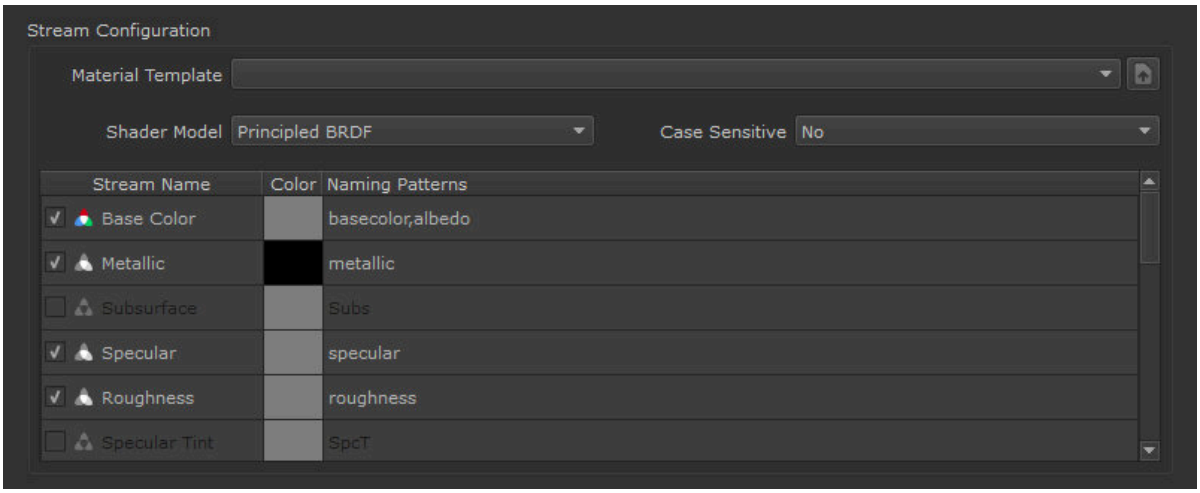
> **Note:** The **Presets** dropdown list contains the names of all preset files found in **C:\Program Files\Mari 8.0v1\Bundle\Media\Settings\MaterialIngestPresets**
> To load custom presets from another location, use the **MARI_MATERIAL_INGEST_PRESETS_SEARCHPATHS** environment variable. See Environment Variables That Mari Recognizes.

OR

Click the **Load Presets** button if you already have a material preset saved on disk, then browse to the **.mip** file and click **Open**.

This loads the preset properties in the **Stream Configuration** and **Ingestion Options** sections.

Stream Configuration section:



1. In the **Stream Configuration** section, you have the option to select a **.mma** file from the **Material Template** dropdown, or from disk using the button. This step is entirely optional.

   This material is used to build the template structure of the new material. Once an **.mma** file is picked, a validation pass is performed on the material to determine the **Shader Model**, the **Procedural Type** (defined by a Tiled or Triplanar node in the **.mma** file with the same name as the stream short name) and any missing streams.
2. In the **Stream Configuration** section, select a **Shader Model**.

   The table containing the stream properties updates accordingly.

> **Note:** If you have selected a **.mma** file in the **Material Template**, the **Shader Model** is locked to the shader model found within the **.mma** file.

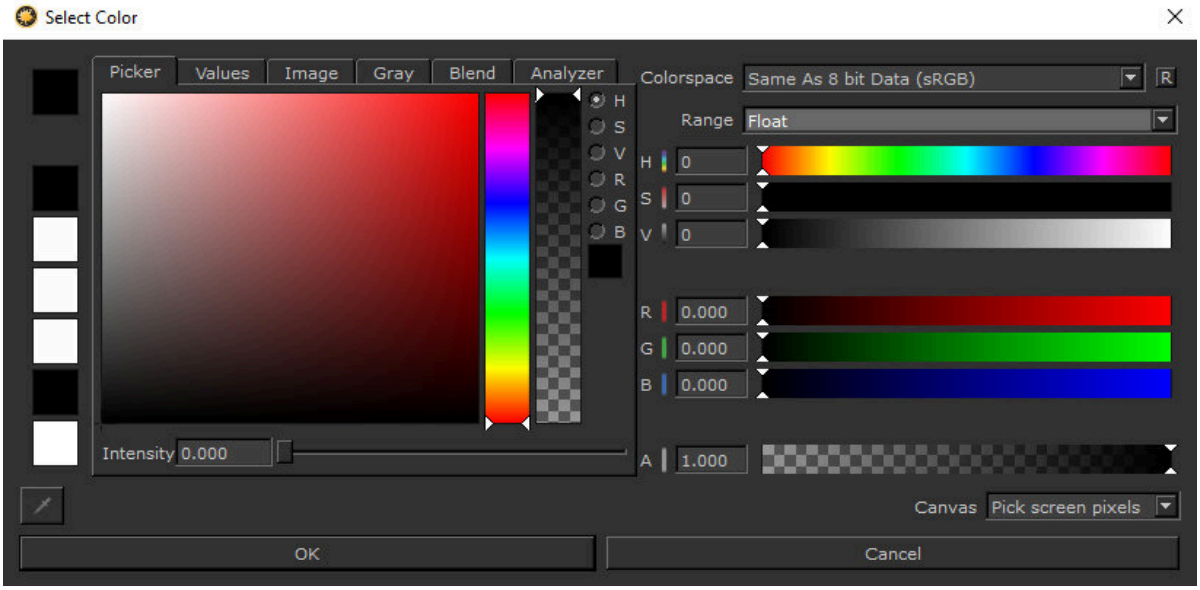
> **Note:** The **Stream Configuration** table is used to match the texture files' naming patterns to the shader input they are targeted for.

3. Select whether the stream naming patterns is case sensitive if your texture files follow varying capitalization conventions.
4. In the **Stream Name** column, disable the checkbox if you want to skip the creation of any procedurals for that stream.

   To disable multiple streams at once, **Ctrl**+click the required rows, then right-click and select **Uncheck Selected**. Likewise, select **Check Selected** to enable the streams.
5. Double-click the color swatch to set a color for the streams in case a matching texture file cannot be found. To set a color for multiple streams at once, **Ctrl**+click the required rows, then right-click and select **Set Color**.

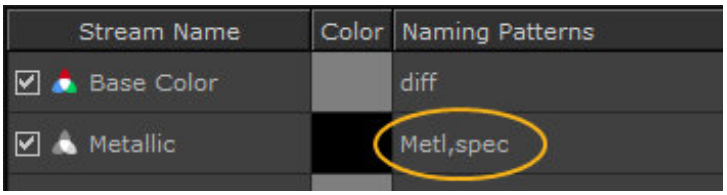
   This opens the **Select Color** dialog. Pick a color and click **OK**.

Tip: Drag the **Select Color** dialog window to expand it.



Note: A Constant procedural is created for Scalar streams. If the RGB values of the Scalar stream's color are unique, a Vector procedural is created.
It is recommended to set the default color of your Shader's Normal stream to **0.5**, **0.5**, **1.0**, which is the default for a flat normal map. This way, it does not affect the normals at this value.

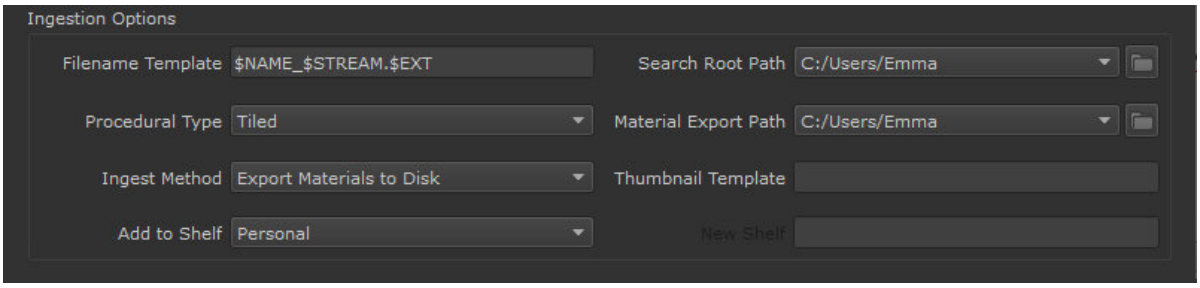
6. By default, the **Naming Patterns** use the streams' short names, but you can change this by double-clicking the name and editing it. If you need to specify more than one naming pattern, use a comma (**,**) to separate them.



To reset the Naming Patterns to their default, **Ctrl**+click to select them, then right-click and select **Reset to Short Names**.

Note: It is recommended to enter the names in the order of importance as when multiple naming patterns are found, the first matching pattern in the row is used and the rest is ignored.

Ingestion Options section:



1. Depending on the naming pattern information of your texture files, set the **Filename Template**. The default value is **$NAME_$STREAM.$EXT**
   For example: **Metal_col.jpg**

   Here are the template options:

   Required:

   - **$NAME** - specifies the name of your material. You can use any set of alphanumeric characters including underscores, spaces, dashes, and periods.
   - **$STREAM** - indicates where in the file name the pattern specified per stream should be matched.

   Optional:

   - **$EXT** - specifies any supported image extensions.
   - **$CHAR** - ignores any arbitrary characters.
   - **$NUM** - ignores any numbers.
   - **$ANY** - ignores any alphanumeric characters including underscores, spaces, dashes, and periods.
   - **$SEP** - specifies the system's path separator, forward slash **/** or backslash ****.
   - **NUMCHAR** can be used together to match against textures with their resolution in the filename, for example: **myMaterial_color_4k**

> **Note:** You can use multiple file templates by separating them with the pipe **|** symbol. If the first template is matched, the rest is ignored.
> To search for texture files in subfolders, you can use the backslash **** symbol, for example, to match **myMaterial\PNG\diffuse.png** use the template **$NAME\$CHAR\$STREAM.$EXT**

2. Browse to select a **Search Root Path** from where the Material Ingest Tool searches for collections of files matching the specified patterns.

   This searches through the sub-folders of the selected path.
3. Choose the **Ingest Method**, whether to export your materials to disk, build them in your project, or display the texture sets found with your specified **Search Root Path** and specified **Filename_Template**.

   If you have selected **Export Materials to Disk**, browse to or enter the **Material Export Path** in the field on the right-hand side.

If you have selected **Display Search Results Only**, the **Material Export Path** and **Add to Shelf** options become disabled. The **Create Materials** button becomes **Find Materials**.

4. In the **Procedural Type** field, select which image based procedural node to generate materials with:
   - **Tiled** - uses one image to tile across the model.
   - **Tri Planar Projection** - uses the same image to project onto the surface of your model from three different directions (Top, Front, and Right).

> **Note:** The **Procedural Type** field is disabled if you have selected an **.mma** file in the **Material Template**. The Tiled or Tri Planar Projection node found in the **.mma** file is used.

> **Note:** Select **Tri Planar Projection** if you don't want patterns to be a tiled repeat or if you are working in a PTex project. PTex projects cannot use Tiled as they have UV seams while Tri Planar doesn't.

5. In the **Add to Shelf** field, select which shelf to add your materials to. You can add them to the **Menu**, **Personal**, or **Project** shelves or you can create a new shelf. Select **Do Not Add to Shelf**, if you don't require to add them to a shelf.

> **Note:** Materials are added to the **Personal** shelf by default.

If you have selected **Add to New Shelf**, enter the name of the **New Shelf** in the field on the right-hand side.

6. Depending on the naming pattern information of your texture files, set the **Thumbnail Template** to specify the naming convention for your thumbnail files. If there is criteria inside this field, Mari searches for a thumbnail image path using the matching template criteria.

   Here are the template options:

- **$NAME** - any set of alphanumeric characters that represent the name of your material.
- **$EXT** - this is replaced by any supported image type.
- **$CHAR** - this can be used to ignore any arbitrary characters.
- **$NUM** - this can be used to ignore any numbers.
- **$ANY** - this can be used to ignore any alphanumeric characters, including underscores, spaces, dashes, and periods.
- **$SEP** - this is replaced by the system's path separator (****).

  If **$NAME.$EXT** has been specified in the **Thumbnail Template**, a thumbnail image path found with the same name as a texture set should be assigned together with the material. For example, if you have three

files: myMaterial_diffuse.exr, myMaterial_specular.exr, myMaterial.png, and **Thumbnail Template** is set to **$NAME.$EXT**, then myMaterial.png is defined as the thumbnail for the found texture set: myMaterial.

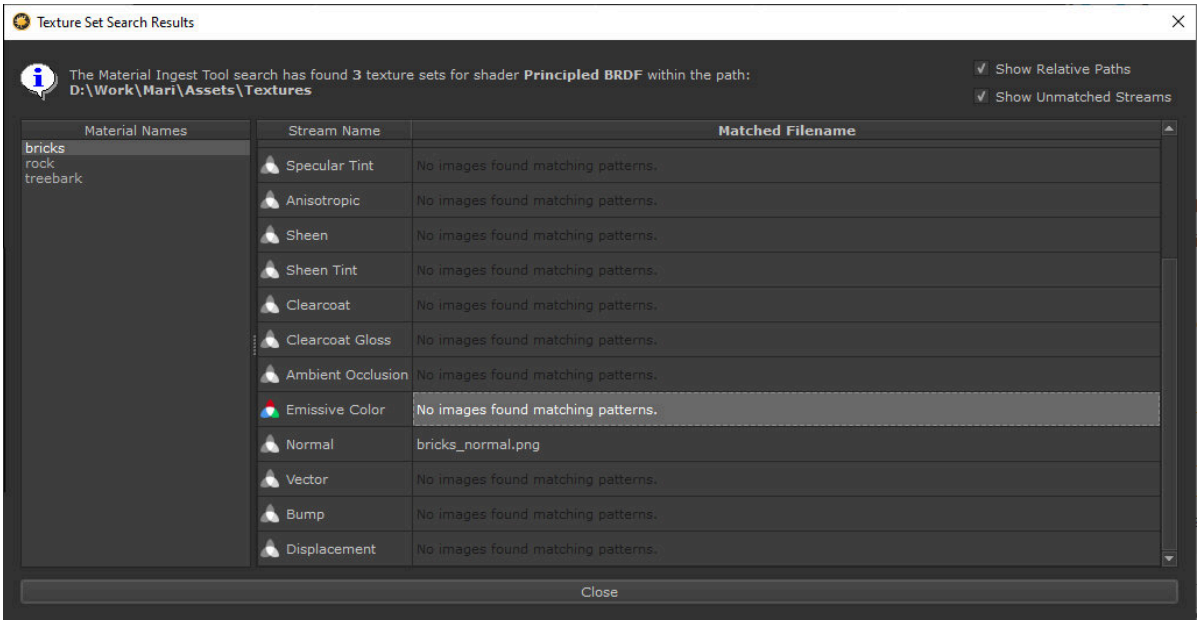
> **Tip:** Ensure the thumbnail adheres to your monitor display color space so there are no color space complications when the material displays in the **Shelf**.

7. Click **Create Materials** to generate materials. If you have selected **Display Search Results Only** in **Ingest Method**, go to step 8.

   The ingestion starts and you can see its progress on the progress bar. Once completed:

   - If you have selected **Export Materials to Disk**:
     - Your materials are exported in **.mma** files to the specified location.
     - If you chose to add your material to a new shelf, the material displays in the selected shelf.
   - If you selected **Build Materials in Project**, your materials are added as Tiled or Tri Planar Projection nodes in the Node Graph.
   - If you have selected **Display Search Results Only**, this button is displayed as **Find Materials** instead, and clicking this opens a new window displaying all the found texture sets for that material with the matched filename
8. If you have selected **Display Search Results Only** in **Ingest Method**, click **Find Materials**. Otherwise, go to step 9.

A **Texture Set Search Results** dialog opens.



Note: Materials are available across your Mari projects that use the same shader as the one selected.

Tip: If you'd like to save your preset for future use, click **Save Presets** and browse to the location where to save it.
This saves your preset as an **.mip** file.

9. If you need to create more materials, repeat the same steps, otherwise click **Close** to exit the **Material Ingest Tool Dialog**, see the *Material Ingest Tool Dialog* section in the *Mari Reference Guide*.

Note: The **Material Ingest Tool** dialog remembers the last settings used.

Note: To apply materials to your scene refer Working with Materials in the Layers Palletto or Working with Materials in the Node Graph.

Material Ingest Tool Templates and Examples

Under the **Material Template** property in the **Material Ingest Tool** dialog, you can select an **.mma** file from disk to use as a template structure when building a new material. Mari comes packaged with 3 template examples for you to use:

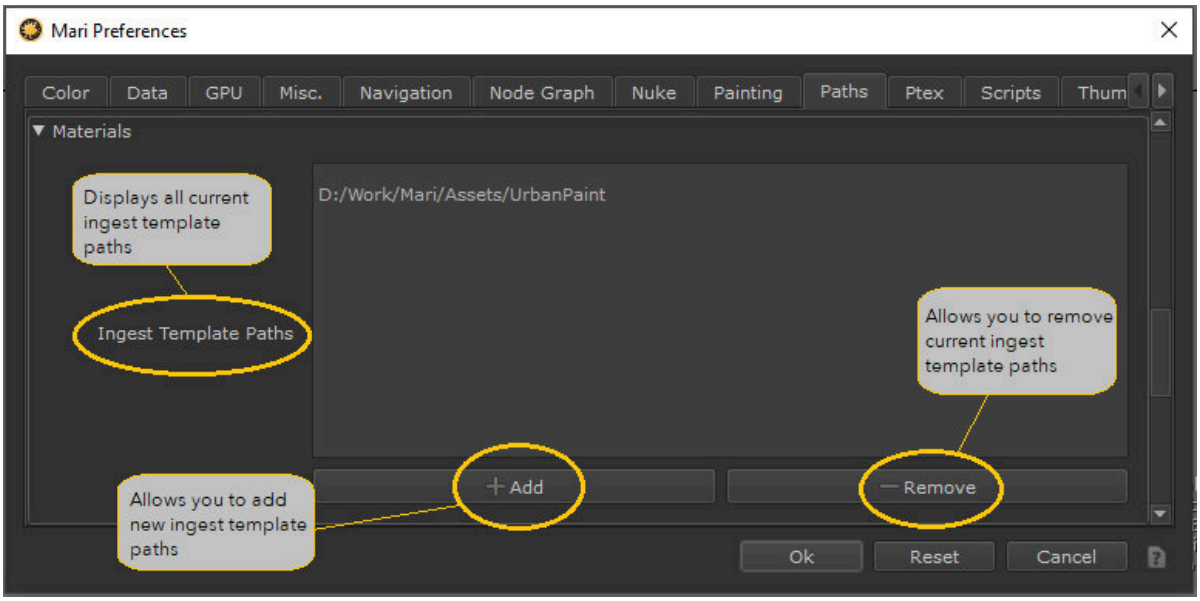
- **GlossinessToRoughness.mma**
- **NormalsFormatInvert.mma**
- **GlossinessToRoughnessAndNormalsFormatInvert.mma**

Once an **.mma** file is selected from the **Material Template** property, the **Shader Model** and streams become locked to the specified template structure.

Adding Ingest Template Paths

You can add material template paths through the **Mari Preferences** under the **Paths** tab. Any **.mma** files found in the specified paths are displayed in the **Material Template** dropdown of the Material Ingest Tool.

1. Click **Edit** > **Preferences**.
2. Click the **Paths** tab.
3. Scroll down to the **Materials** section.



The **Materials** section of the **Paths** tab in the **Mari Preferences**.

4. Click **Add** to open up the **Add Directory** dialog.

   You can select a directory containing template **.mma** files.
5. Click **OK** to add the path to the **Ingest Template Paths**.
6. To remove any paths from the **Ingest Template Paths** field, select the path in the **Ingest Material Paths** field and click **Remove**.

> **Note:** Any paths defined in the **Ingest Template Paths** can be overridden by the following environment variable: **MARI_MATERIAL_INGEST_TEMPLATES_SEARCHPATHS**.

Working with Materials in the Layers Palette

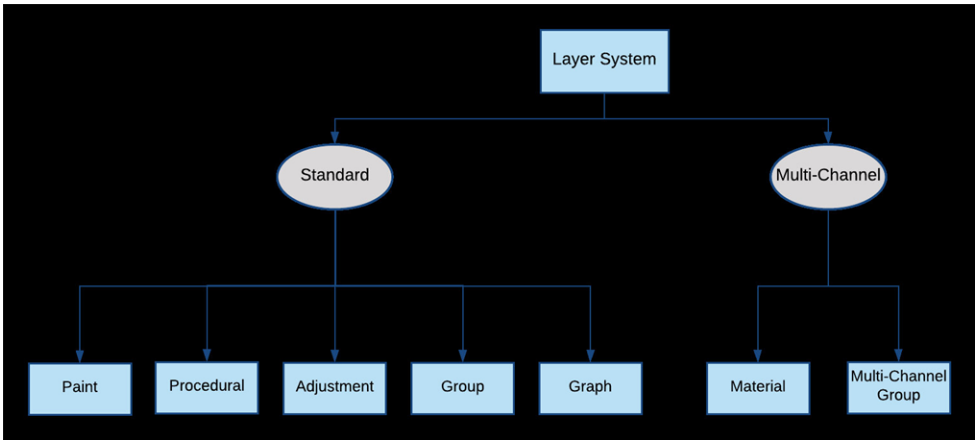
Once you've created your materials or if you are using the default Mari materials stored in the **Mari Materials** tab of the **Shelf** palette, you can apply materials to your scene using the **Layers** palette.

Note: Mari Materials located in the **Shelf** palette only work with the Principled BRDF shader.

Note: To find out how to create material presets using the Material Ingest Tool, refer to Creating Materials with the Material Ingest Tool, or if you prefer to create your own manually using the Node Graph, refer to Working with Materials in the Node Graph.

Layer System

Materials are shared across all channels and the way to add them to your projects is with multi-channel layers. Multi-channel layers are either a material or a multi-channel group. Here is a diagram showing you the layer system with standard and multi-channel layers using the **Layers** palette.



Adding Materials to Your Scene

You can add materials, which are a type of multi-channel layer, through the **Layers** palette or by dragging materials from the **Shelf** palette and dropping them on the asset directly.

Note: To properly apply materials to your scene, ensure that:
• You've selected the required user shader in the **Shaders** palette, for example, **Principled BRDF**.
• You have channels created and connected to the appropriate inputs of the shader that match the streams/material outputs of your materials. See Creating Multiple Channels from Presets for auto-creating and connecting channels. To check the streams/material outputs of your materials, in the **Layers** palette, click the Material icon of the material layer. This opens the material subgraph in the **Node Graph** palette.

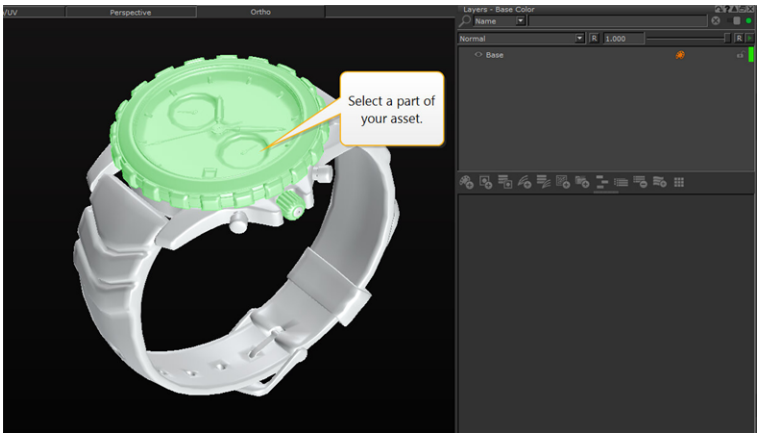
Note: To get the best visual result, ensure you have an environment light. See Configuring the Lighting.

Adding Materials to Your Assets from the Shelf Palette

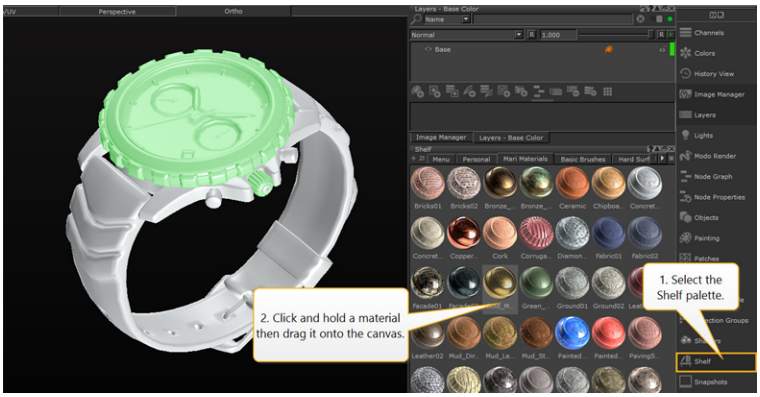
Dragging a material from the **Shelf** palette to the canvas applies it to the current object. Alternatively, you can use a selection tool to add the material with a mask of the selected region.

Note: Dragging a material from the **Shelf** palette to the layerstack applies it to the current object. This method does not allow you to add the material with a mask of the selected region.

1. In the **Tools** toolbar, select the **Select** tool then choose a part of your asset on the canvas to which you want to apply your material.
   This highlights the selected part in green.

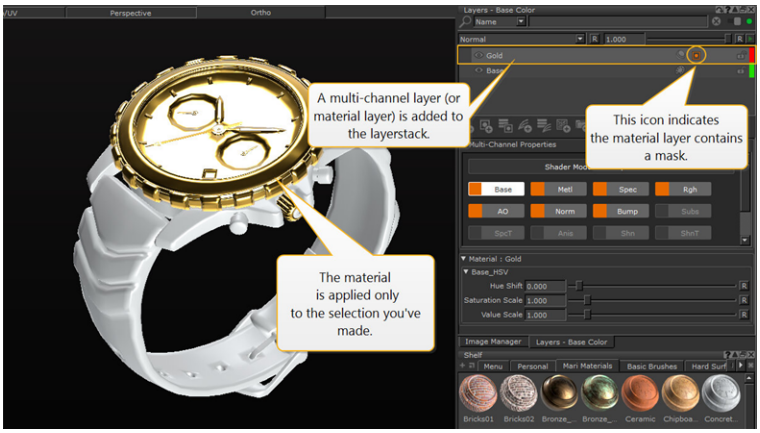


2. In the **Palettes Toolbar** (see the Palettes Toolbar section in the *Mari Reference Guide*) in the **Shelf** palette, navigate to the tab where your materials are stored or select the **Mari Materials** tab, then click and hold a material and drag it onto the canvas.



This adds the material with a mask to the selection only.

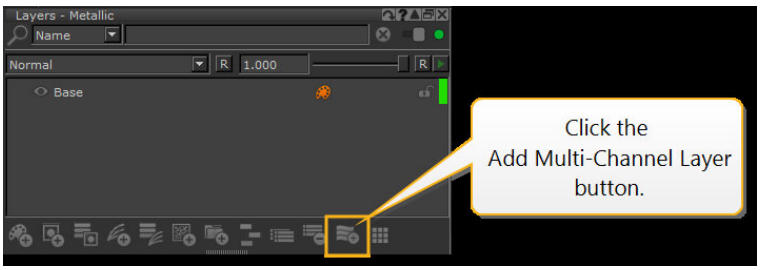
In the **Layers** palette, this adds a multi-channel layer with a mask applied on the selected part of your asset.



Adding Multi-Channel Layers through the Layers palette

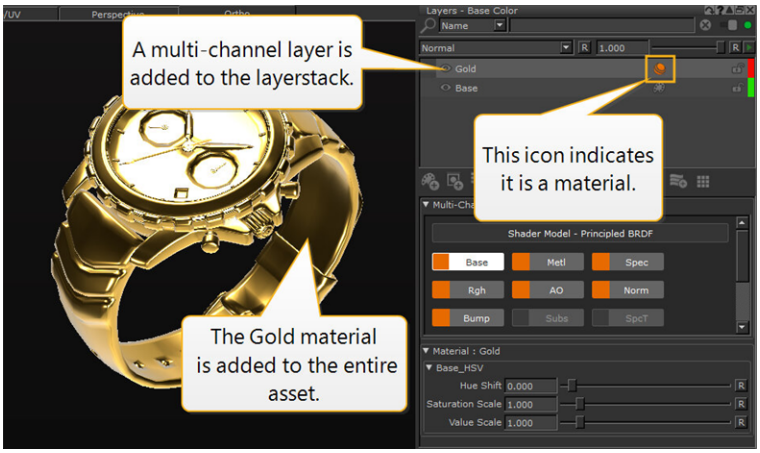
You can add materials to your scene using multi-channel layers from the **Layers** palette:

1. In the **Layers** palette, click the **Add Multi-Channel Layer** button.



2. Navigate to **Materials** > **Mari Materials**, or any other shelves to which you added materials, then select a material.

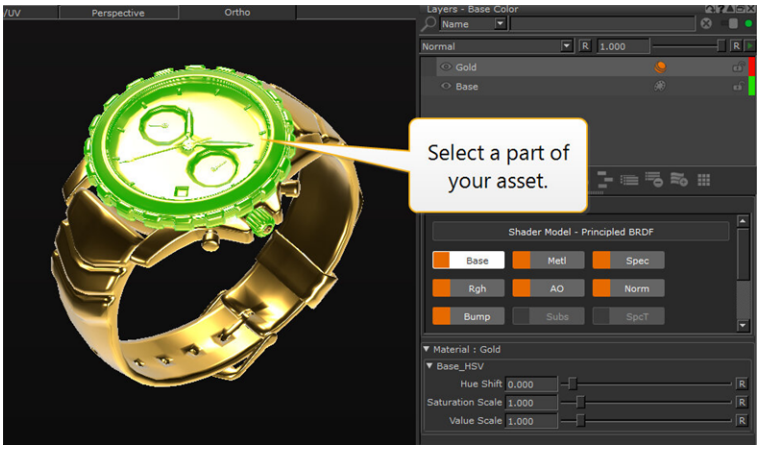
   This applies the material to the entire asset (see canvas) and adds a multi-channel layer to your layerstack.



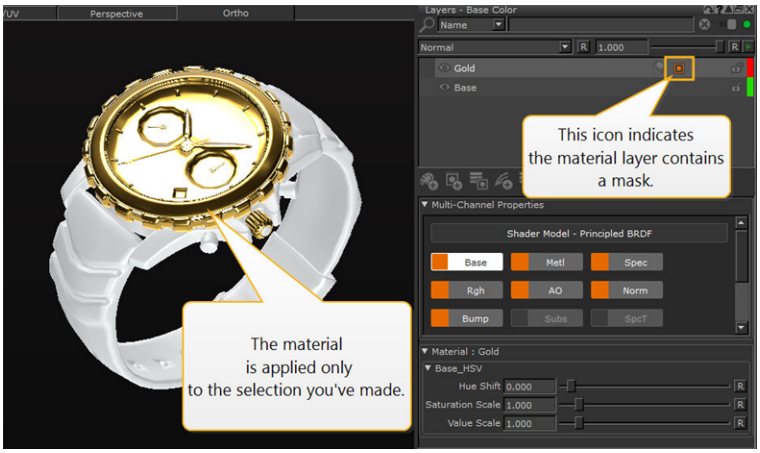
Note: A material layer is not a paintable layer but you can use masks to choose how to apply the material to your asset.
You can also convert a material layer to a paintable layer. In the **Layers** palette, right-click on the required layer and select **Convert to Paintable**. See the **Layer Functions** section in the *Mari Reference Guide*.

If you want to apply a material to parts of your asset only, you can use masks:

3. In the **Tools** toolbar, select the **Select** tool then choose a part of your asset on the canvas to which you want to apply your material.
   This highlights the selected part in green.



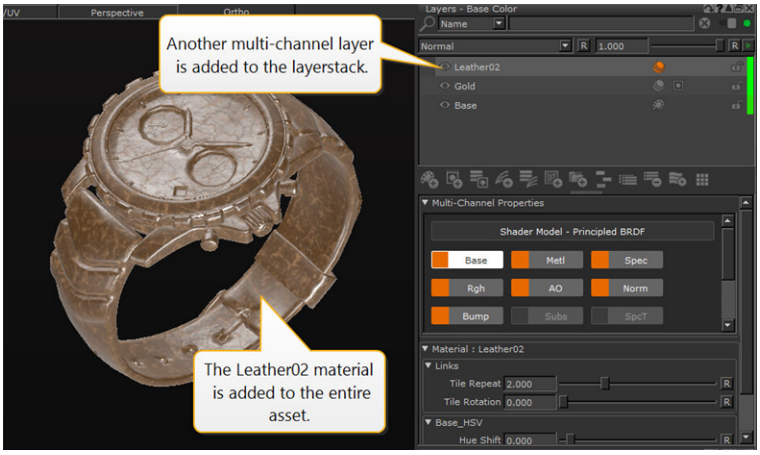
4. In the **Layers** palette, right-click on the required multi-channel layer and navigate to **Layer Mask** > **Add Mask** > **From Selection**.
   This adds a mask to the multi-channel layer, applying the material to the selection only.
   This mask is used across all sibling streams/channels.



To add another material, to the bracelet part of the watch for example, do the following:

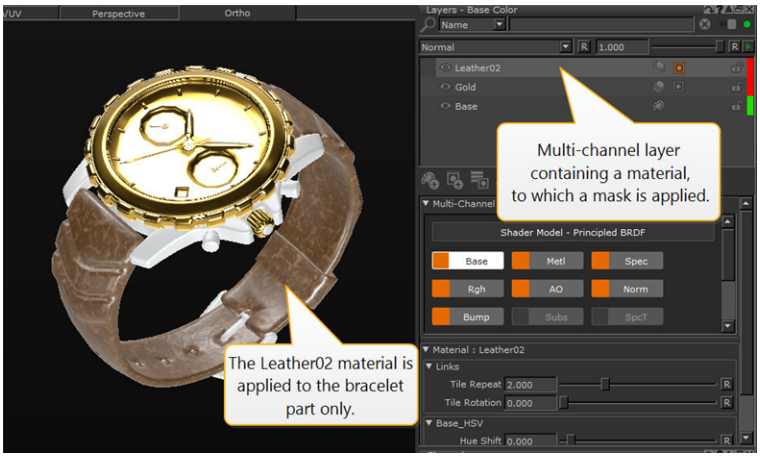
5. In the **Layers** palette, click the **Add Multi-Channel Layer** button and navigate to **Materials** > **Mari Materials**, or any other shelves to which you added materials, then select a material.

   This applies the material to the entire asset (see canvas) and adds another multi-channel layer to your layerstack.



6. On the canvas, select the bracelet part.
7. In the **Layers** palette, right-click on the new multi-channel layer.
8. Navigate to **Layer Mask** > **Add Mask** > **From Selection**.

   This adds a mask to the multi-channel layer, applying the material to the bracelet part only.

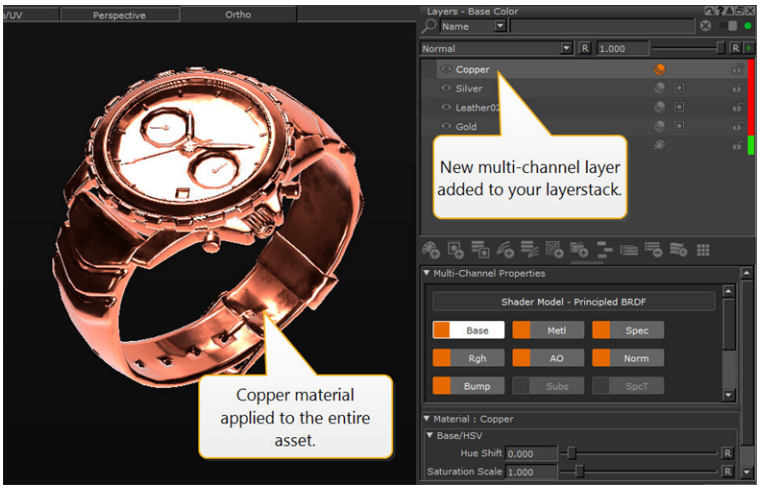


Using Pass-Through with Multi-Channel Groups

Multi-Channel Groups work the same way as standard groups except that they are meant to work with materials, which are shared across all channels. See Groups, Pass-Through, and Mask Stacks for more information on standard groups. You can use Multi-Channel Groups to build material stacks for instance, and use most layer masking operations.

Pass-Through mode can be used with multi-channel groups the exact same way as with standard groups. The steps below show you one way to use the Pass-Through mode with materials contained in a multi-channel group.

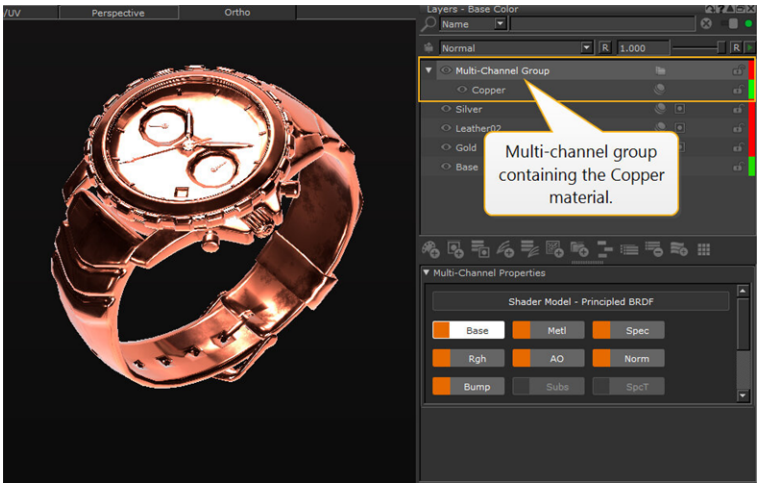
1. In the **Layers** palette, add a new material on top of your current materials. Click the **Add Multi-Channel Layer** button, navigate to **Materials** > **Mari Materials**, or any other shelves to which you added materials, then select a material.
   In this example, the Copper material is used.



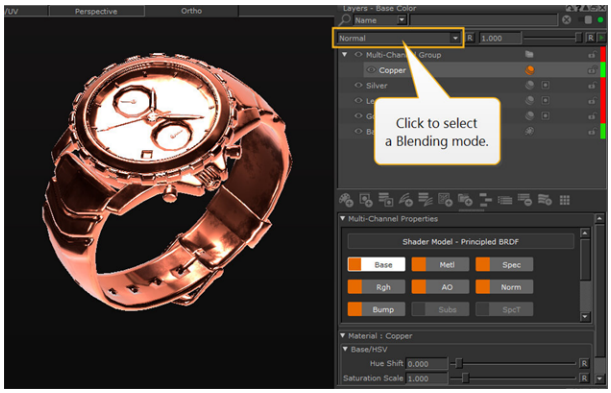
2. In the **Layers** palette, select the material layer at the top of the layerstack, then click the **Add Multi-Channel Layer** button and select **Group**.

Tip: If you want to add an empty multi-channel group, make sure that you haven't selected any layers in the layerstack, otherwise it creates a group with the selected layer(s) in it.

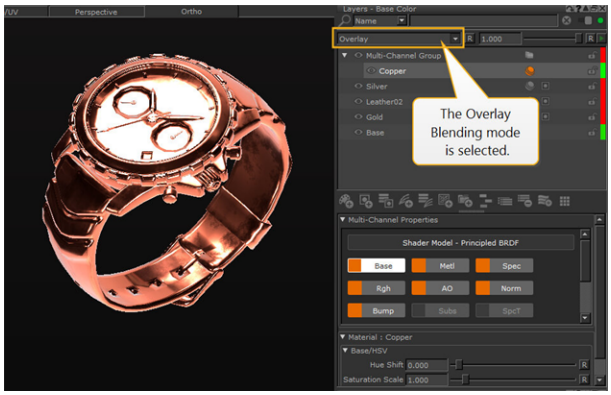
This adds a multi-channel group to your layerstack, which contains the Copper material layer.



3. Select the Copper material layer and choose a blend mode. For this example, navigate to **Contrast** > **Overlay**.



The blending mode is set to Normal.



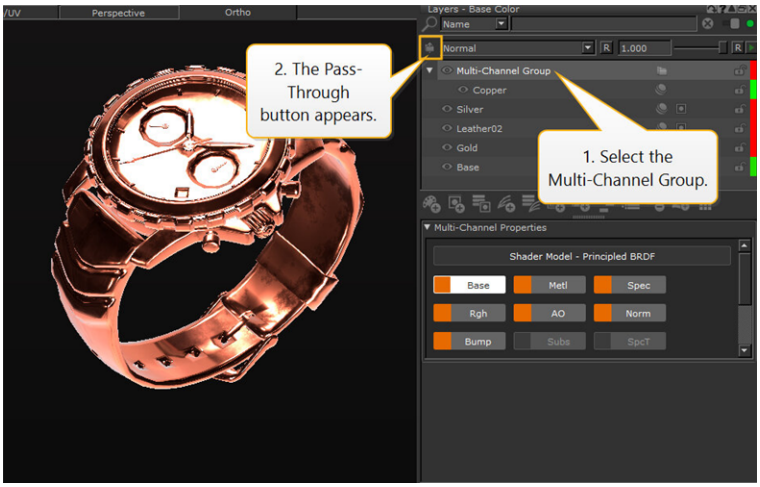
The **Overlay** blending mode is selected.

Since the Copper material layer is in a multi-channel group, the **Overlay** blending mode is not applied to the layers below, as it is applied to the bottom of the stack within the multi-channel group, which is transparent.

To pass-through the blending mode properties to the layers below:

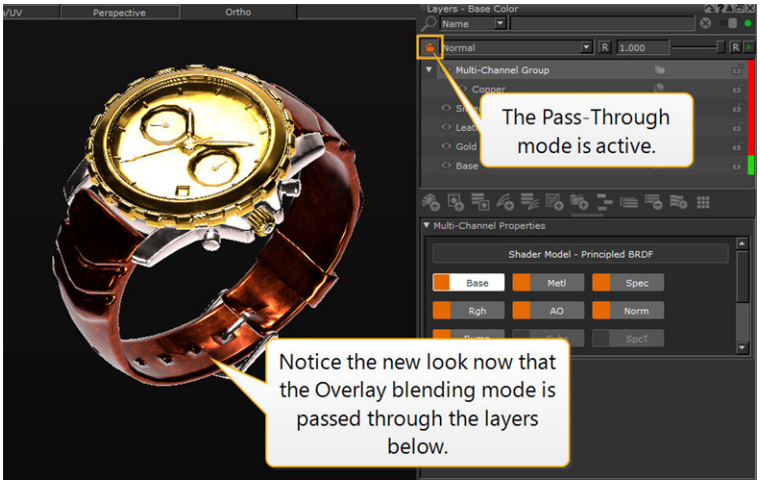
4. In the **Layers** palette, select the multi-channel group.

The Pass-Through button appears at the top of the **Layers** palette.



5. Click the Pass-Through button.

   The Overlay blending is passed through the layers below.



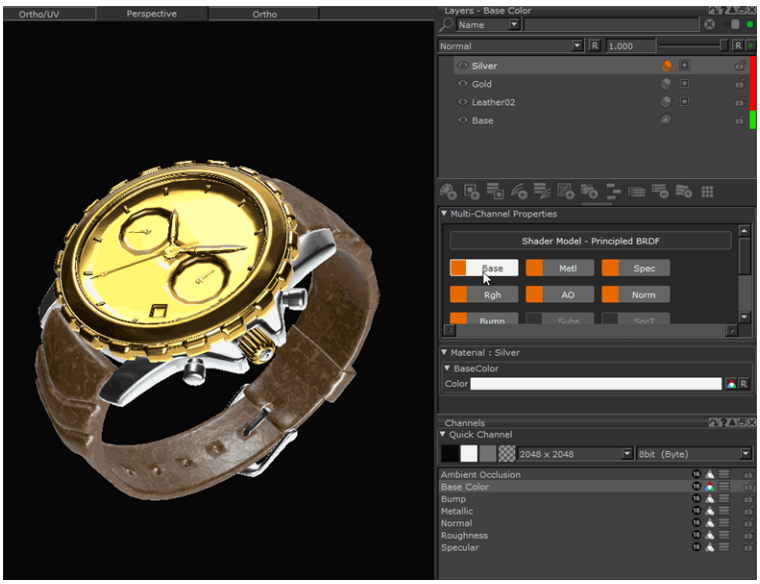
Note: The Pass-Through mode works per channel/stream and is not shared across all channels like materials are.

Controlling Multi-Channel Properties

From the **Layers** palette, in the **Multi-Channel Properties** section, you can switch between channels and enable or disable streams/shader inputs.

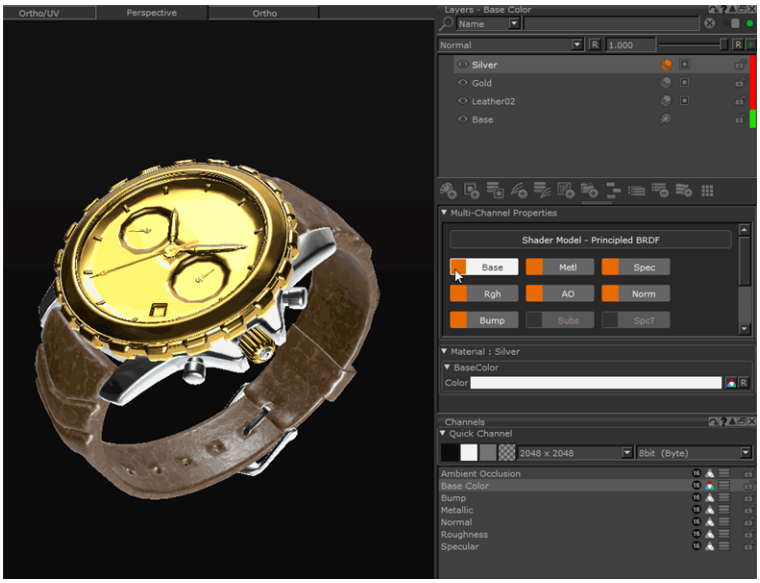
Switching Channels

In the **Layers** palette, in the **Multi-Channel Properties** section, click on the channel name to select the channel.
Notice that the channel is selected in the **Channels** palette.



Enabling and Disabling Streams/Shader Inputs

In the **Layers** palette, in the **Multi-Channel Properties** section, click the orange toggle button to disable the streams/shader inputs, and click the gray toggle button to enable them. Alternatively, in the layerstack, **Shift**+click the Eye button to enable/disable the stream of the current channel.
In the example below, the streams/shader inputs of the Silver material are being enabled and disabled.
Notice the changes on the canvas.

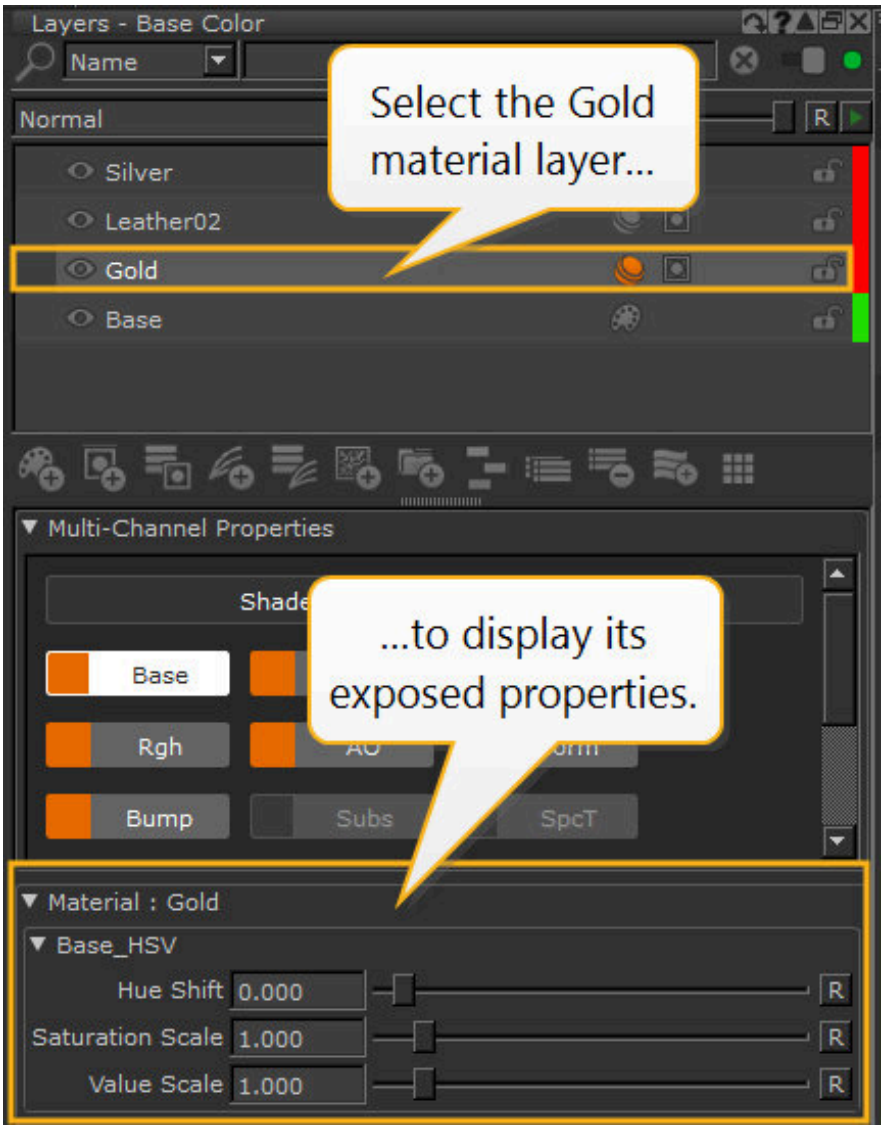


Note: This does not affect the material layer's visibility and only changes the state of the selected streams/shader inputs, which is reflected on the canvas.

Controlling Material's Exposed Properties

The materials' exposed properties or controls are displayed at the bottom of the **Layers** palette under the **Multi-Channel Properties** section.

Click the required material layer (or also known as multi-channel layer) to display its exposed properties or controls. You can then change the values of the properties with the different controls. For example, use the slider to change the **Hue Shift** value of the **Gold** material.



Note: With the Node Graph workflow, you can expose the properties of the nodes contained in your material node and then link them to create controls to change the Material node's properties. See Working with Materials in the Node Graph for more information.
If you are using the Node Graph workflow, the materials' exposed properties are displayed in the **Node Properties** palette. See Working with Materials in the Node Graph.

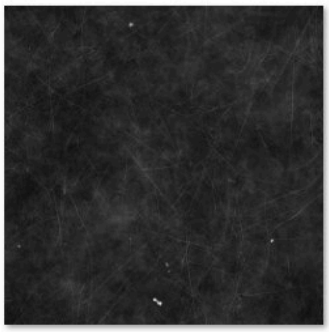
Working with Materials in the Node Graph

Materials are created from multiple texture files, procedurals, or plain colors, which can be used to accurately represent how light interacts with real-world materials. This method of shading is called Physically Based Rendering (PBR). You can apply these materials to your assets for a more realistic result as well as establishing a base line look for your asset before adding more bespoke paint and crafting details.

Here are examples of texture files:

Metal07_col.jpg

Metal07_nrm.jpg



Metal07_rgh.jpg

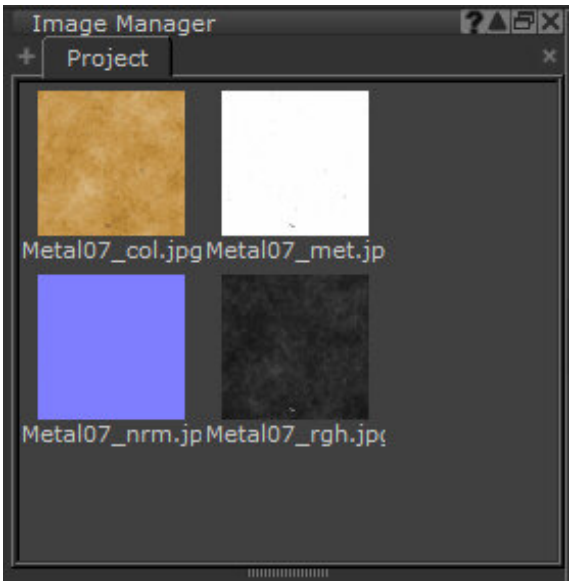
You can create your materials manually using Material nodes and use them for texturing your assets. You can then export Material nodes as **.mma** files to share your materials with other texture artists.

> **Note:** If you only want to create a material from a PBR texture set without any additional nodes, you can use the Material Ingest Tool to automatically create materials. Refer to Creating Materials with the Material Ingest Tool.

Creating a Material Node from Texture Files

You can use a material template node called **Material** to create materials in Mari.

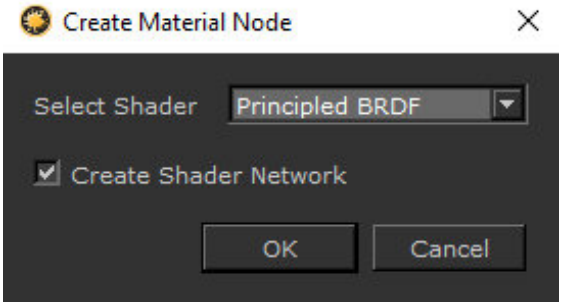
1. Make sure that your texture files, which you need to create your material, are imported in the **Image Manager** palette.



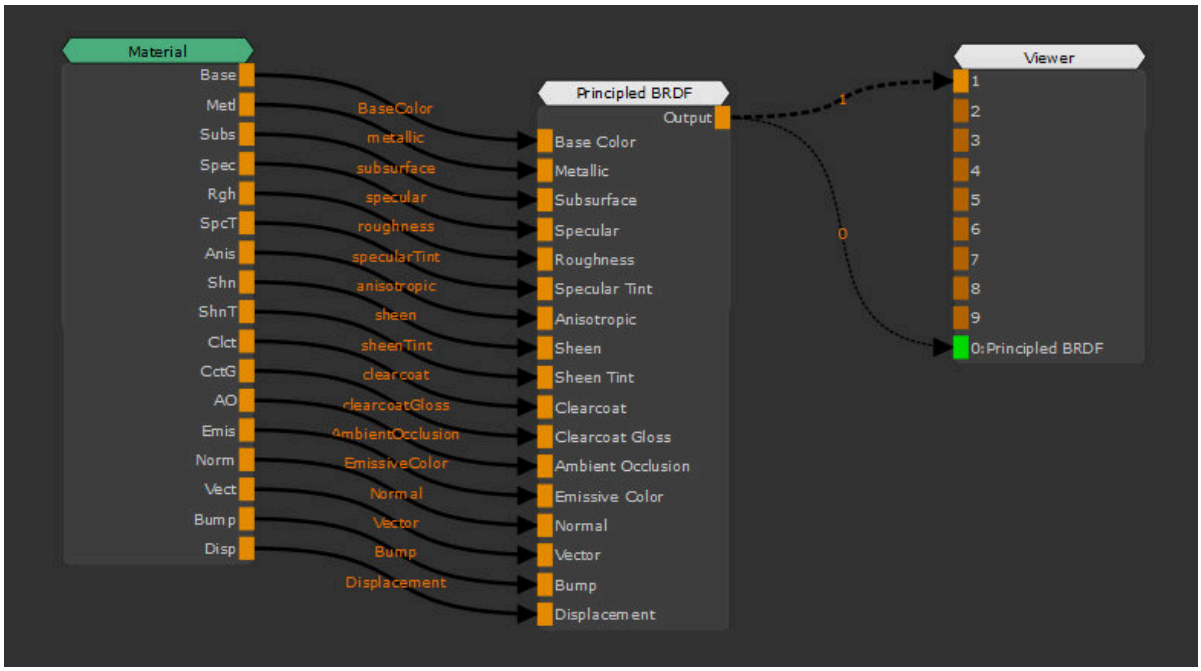
2. In the Node Graph, make sure you haven't selected any nodes.
3. Press the **Tab** key and start typing 'Material'and select **Material** from the list of matches.

   Alternatively, right-click in the Node Graph and from the menu that opens, navigate to **Nodes** > **Misc** > **Material**.

   This opens the **Create Material Node** dialog.
4. In the **Select Shader** dropdown, select the appropriate shader model for your material pipeline and enable the **Create Shader Network** checkbox if you want a new shader created to preview your material on the canvas.



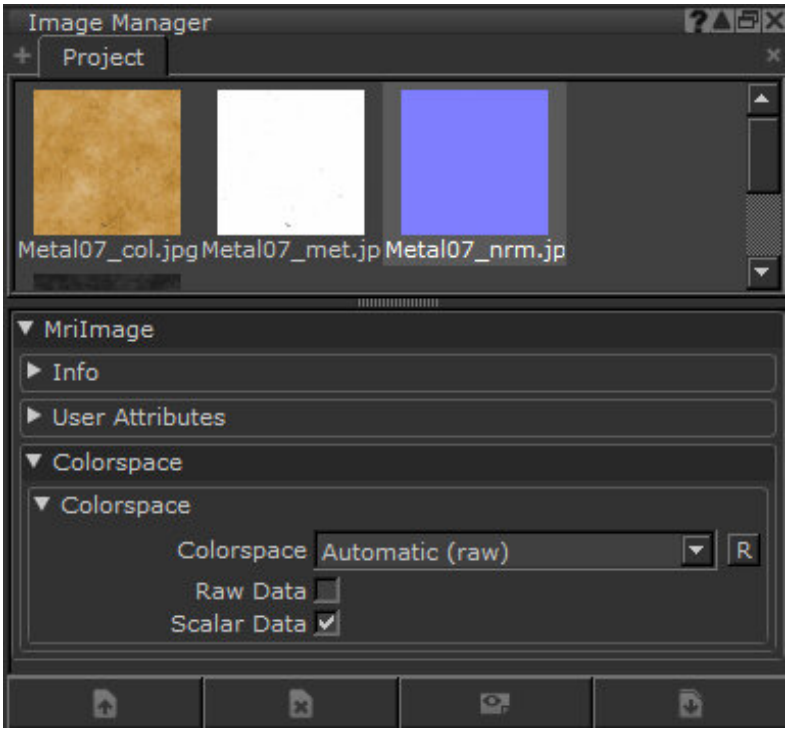
A Material node and a shader node are created in the Node Graph. The Material node automatically batch connects to the shader node.



5. Select the Material node and press **Ctrl**+**Enter**.

   This opens the Material node in a subgraph and displays the material inputs.

6. In the **Image Manager** palette, tag your texture files as **Scalar Data** except the color textures, in this example: **Metal07_col.jpg**.
   See Color Data and Scalar Data for more information on scalar data.



7. From the **Image Manager** palette, drag each image making up your texture into the Material node's subgraph and connect them to the relevant material stream outputs.

> **Tip:** First navigate to the required stream/material output and then drag the image. Once you dragged the image, you can rename the Tiled node for clarity.

> **Note:** For image based textures, you can use **Tiled** or **Tri Planar Projection** nodes. **Paint** nodes are not supported for materials.

Your **Material** node is created, the texture is applied to your asset.

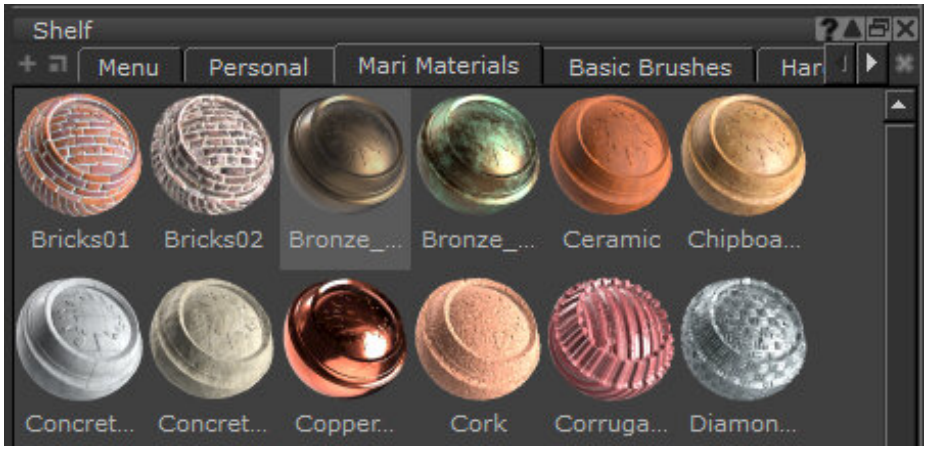
> **Video:** Watch Connecting Texture Files to Material Node's Streams.

You can now export your material node, see Exporting a Material Node.

> **Note:** To apply materials to your asset, refer to Applying a Material to an Asset.

Creating a Material Node from an Existing Material

In the **Shelf** palette, from the **Mari Materials** tab or a custom tab containing materials, drag a material into the Node Graph.



This creates a Material node.
The images making up the material are imported in the **Image Manager** palette in a tab called **Material**

Images. If the material is solely made from procedurals, no images are added to the **Image Manager** palette.

> **Note:** To apply your material to your asset, refer to Applying a Material to an Asset.

Exposing Material Properties Using the Promote Button

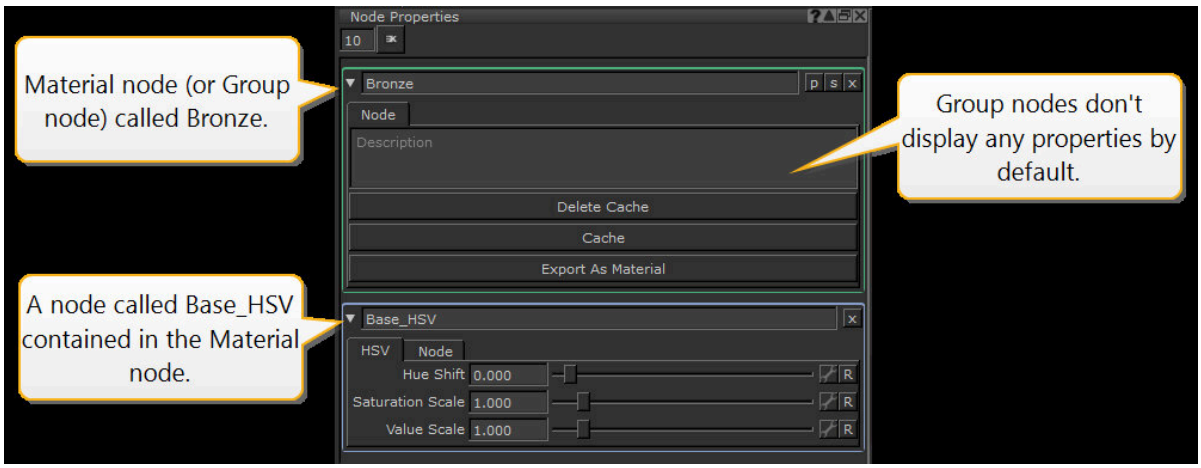
A Material node is a type of group node and you can expose node properties of a Group node by clicking the Promote button. See Using Backdrops, StickyNotes, Groups, and Gizmos for more information on Group nodes.

Group nodes don't display any properties by default, other than those shared by all nodes. See Working with Node Properties for more information.

1. Enter a Group node and then double-click the required nodes within this Group node.

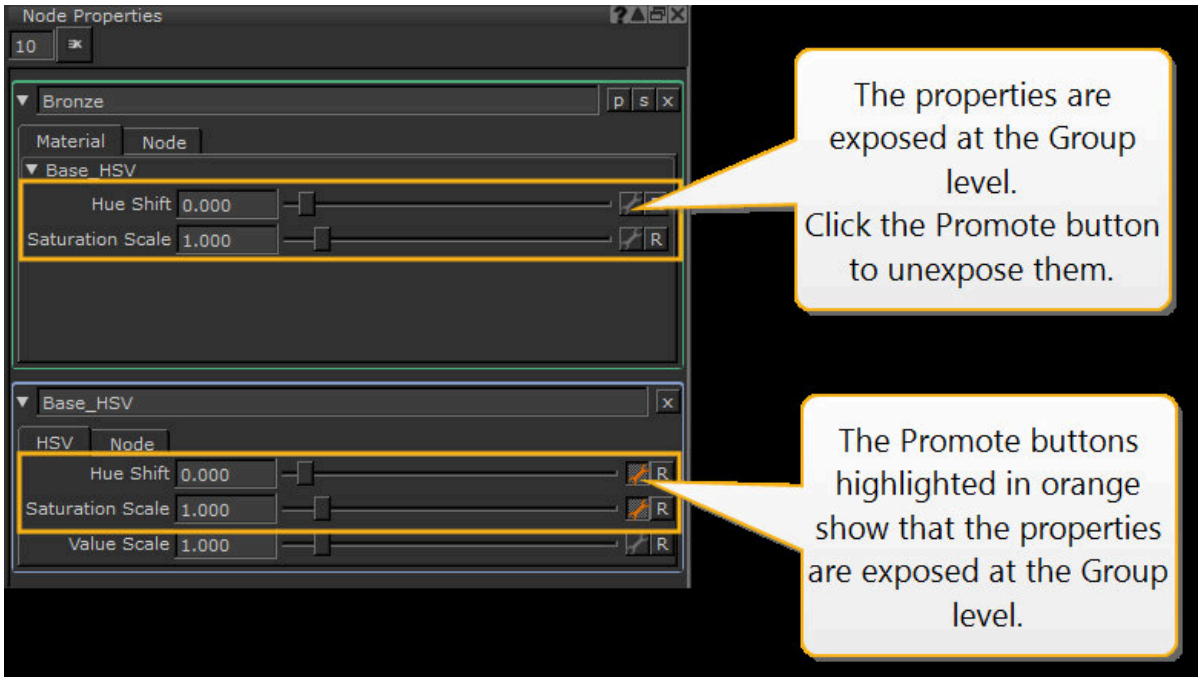
   > **Note:** To enter a Group node, **Ctrl**+double-click the Group node to open its subgraph. Alternatively, select the Group node and press **Ctrl**+**Enter** or in the **Node Properties** palette, click the **S** button.

   The properties of the nodes display in the **Node Properties** palette.



2. Enable the Promote button.

   The Promote button turns orange and the node properties are exposed at the Group node level.



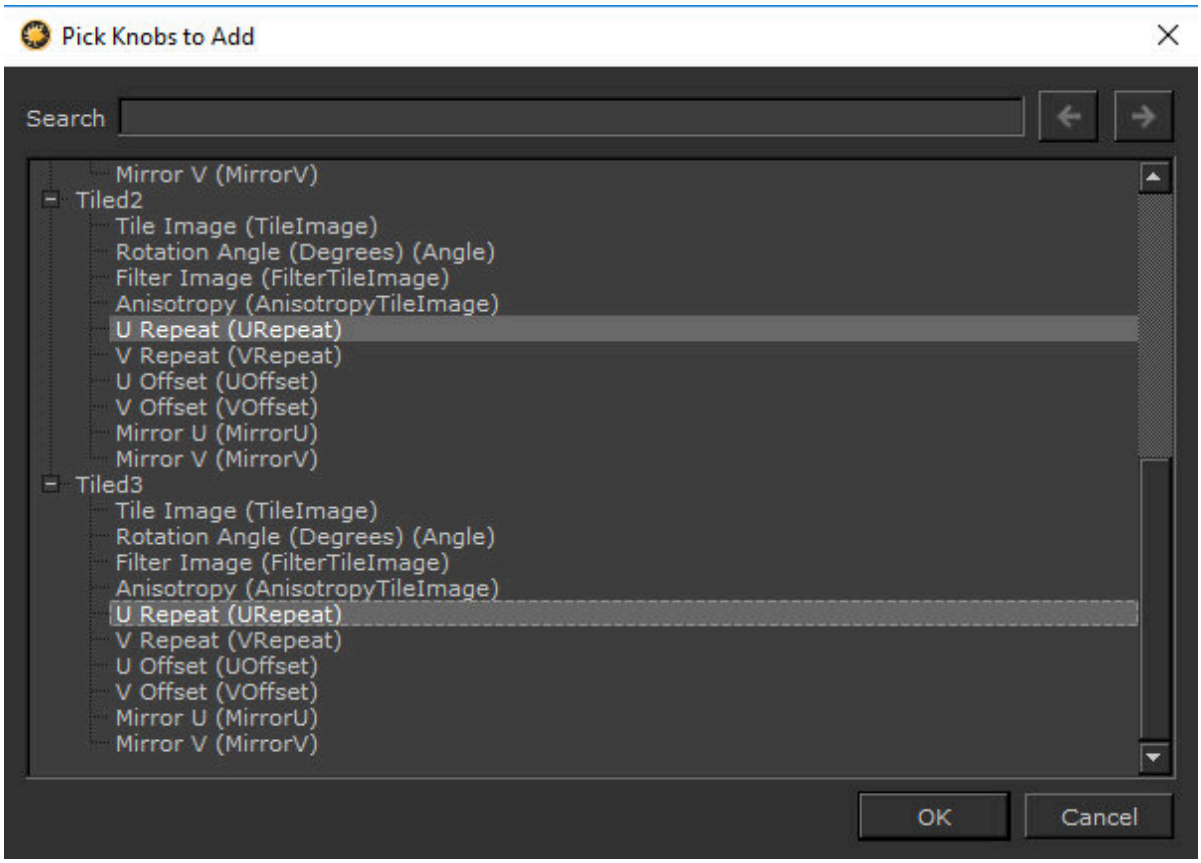
Disable the Promote button to unexpose the node properties.

> **Note:** You can link related exposed node properties together under one control (see Creating Knobs in a Material Node) but if you unexpose one or more node properties that are part of this control, this deletes the control at the Group node level as the control becomes irrelevant.

Creating Knobs in a Material Node

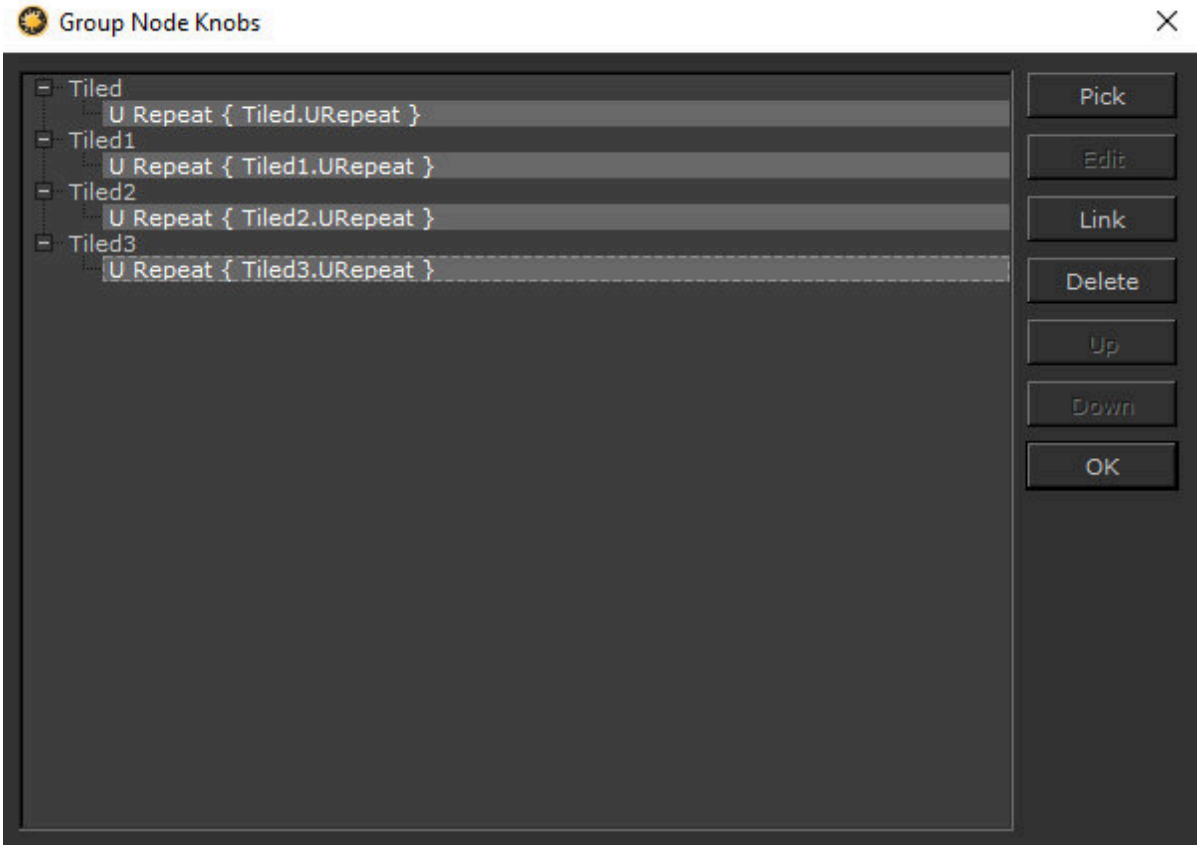
Similar to a Group node you can add knobs (controls) to your Material node, which is a type of group node, using the **Group Node Knobs** dialog. You first need to expose the properties of the nodes contained in your material node and then link them to create controls to change the Material node's properties.

1. In the **Node Properties** palette, in the Material node's properties, click the **P** button.
   This opens the **Group Node Knobs** dialog.
2. Click **Pick**.
   This displays the node's properties of all the nodes in that group.
3. If you haven't exposed properties for that node, select the related properties and click **OK**.
   If you already have exposed properties, skip this step.



This exposes the node's properties at the Material node level.

4. Select the properties you want to group under one control and click **Link**.

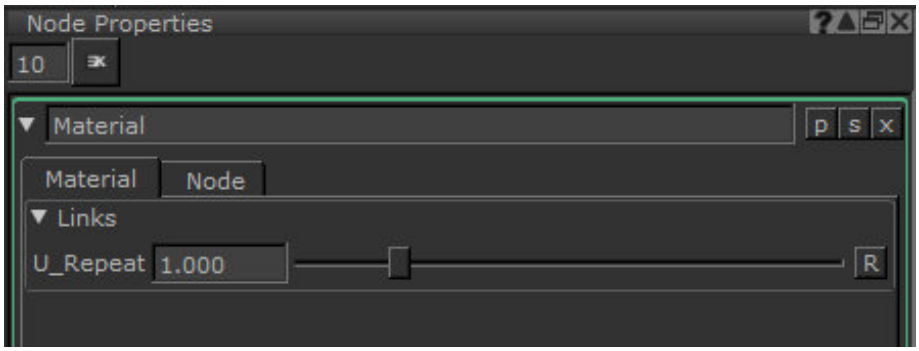


This opens the **Link Knobs** dialog.

5. Enter the name of the new knob and click **OK**.

Note: Spaces are not allowed in knob names.

The exposed properties are linked under one knob (control) at the Material node level in the **Node Properties** palette.



6. Click **OK** to close the **Group Node Knobs** dialog.

> **Note:** Refer to *Customizing Group Properties* and *Linking Exposed Properties* on the Using Backdrops, StickyNotes, Groups, and Gizmos page for more details.

Copying and Pasting Node Knobs

You can copy and paste nodes within Material nodes that have promoted attributes. For example, you can have a Tiled node within a Material node that has its **Rotation Angle (Degrees)** property promoted as a knob on the Material node. Copying (**Ctrl**+**C**) and pasting (**Ctrl**+**V**) the Tiled node into the same Material, or a different Material, copies the promoted attribute over depending on where the node is pasted to, and whether the source node has links. See the table for a breakdown on this behaviour.

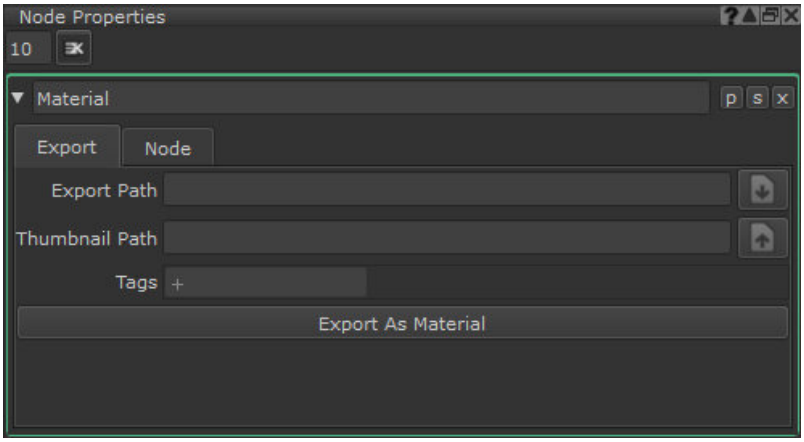
Action	Result	Comments
Pasting a node to the same Group *If original node's knob is promoted*	Newly pasted node's attribute is promoted but unlinked.	The name of the pasted node matches that of the copied source, using number prefixes to identify the different nodes.
Pasting a node to the same Group *If original node's knob is promoted and linked*	Newly pasted node's attribute is promoted and linked to the copied node's linked knob.	The name of the pasted node matches that of the copied source, using number prefixes to identify the different nodes.
Pasting a node to a different Group	Newly pasted node's attribute is promoted but unlinked.	The name of the pasted node matches that of the copied source, using number prefixes to

If original node's knob is promoted		identify the different nodes.
Pasting a node to a different Group *If original node's knob is promoted and linked*	Newly pasted node's attribute is promoted and unlinked as the original node is not in the same Group.	The name of the pasted node matches that of the copied source, using number prefixes to identify the different nodes.
Pasting a node to a different Group *If original node's knob is promoted and linked to other nodes in the clipboard*	Newly pasted node's attribute is promoted and unlinked from the copied nodes not in the same Group, but linked to the sibling nodes pasted into the same Group.	The name of the pasted node matches that of the copied source, using number prefixes to identify the different nodes.
Pasting a node to the root Node Graph	All knobs linked or otherwise are removed.	

Exporting a Material Node

You can export your materials as **.mma** files to share with other texture artists.

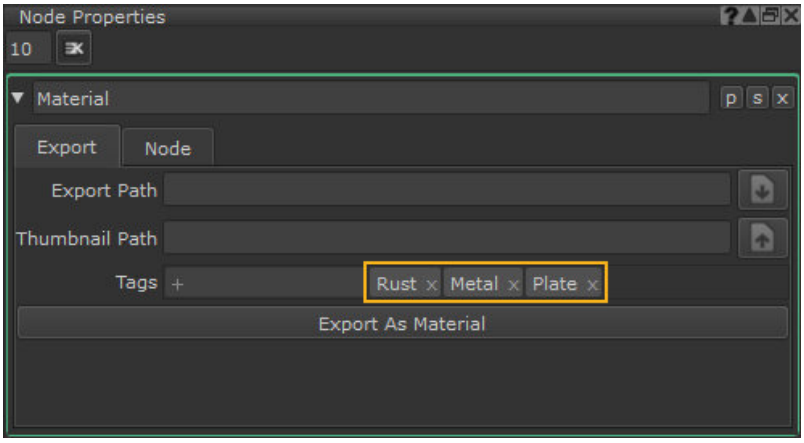
1. In the Node Graph, double-click the Material node.
   This opens the Material node's properties in the **Node Properties** palette.



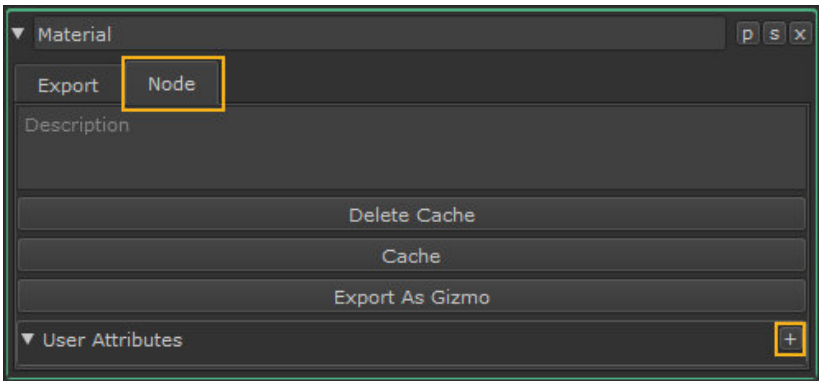
The Material **Node Properties** panel.

2. Select the **Export** tab.
3. Specify a file path in the **Export Path** field, or click to open a file dialog and browse to the export location.
   The default **File Name** is the Material node name.
4. Specify a thumbnail image in the **Thumbnail Path** field, or click to open a file dialog and browse to the thumbnail image location.
   If you don't specify a **Thumbnail Path**, Mari creates one for you before the material is exported.
5. Add metadata tags to your material, if required. Start typing a description into the field to display a list of existing tags or create your own and press **Enter** to add the tag.
   Tags enable Mari to display information about an **.mma** file before you apply the material. For example, if the material represents a rusty metal surface, you could added descriptive tags as shown to identify the material more easily.

> **Note:** Mari includes Python API commands to return information from tags in **.mma** files without actually importing the file. In Mari, go to **Python** > **Documentation** and search the index for **tagList()**for more information.

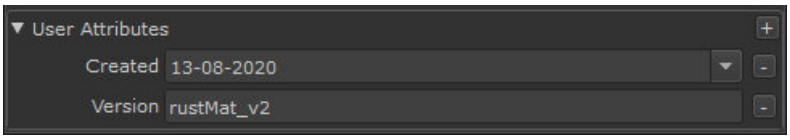


6. You can also add **User Attributes** to Material nodes in the **Node Properties** > **Node** tab.
   Click the icon to add an attribute.



User Attributes allow you to add information such as database references and version control information for asset management systems, such as Shotgun.

7. Choose from a **Boolean**, **Date**, **Float**, **Integer**, or **String** attribute and give the attribute a name.
8. Click **Add** to create the attribute. For example, you could add creation date and version information using the **Date** and **String** attributes.



9. On the **Export** tab, click **Export As Material**.
   The **Material** node is exported as an **.mma** file to the specified **Export Path** location. The **.mma** file contains all texture files used in Tiled or Tri Planar nodes within the Material node.

Importing a Material (.MMA File)

You can import materials (**.mma** files) in the Node Graph using the following methods:

> **Note:** Mari includes Python API commands to return information from tags in **.mma** files without actually importing the file. In Mari, go to Python > **Documentation** and search the index for **tagList()**for more information.

You can import a material file directly in the Node Graph by dragging the **.mma** file from your file browser into the Node Graph.

This creates a Material node.
The images composing the material are imported in the **Image Manager** palette in a tab called **Material Images**. If the material is solely made from procedurals, no images are added to the **Image Manager** palette.

OR

1. Right-click the Node Graph and navigate to **File** > **Import Material**.
   This opens the **Import Material from Archive** dialog.
2. Browse to the material location, select an **.mma** file, and click **Open**.
   This creates a Material node. The images composing the material are imported in the **Image Manager** palette in a tab called **Material Images**. If the material is solely made from procedurals, no images are added to the **Image Manager** palette.

Applying a Material to an Asset

Once your Material node is added to the Node Graph, you need to connect it to your scene to apply it to your asset. You can use two methods:

- Batch connection - where Mari connects multiple sibling pipes with one action. This speeds up the process.
- Manual connection - where you need to connect nodes manually.

Note: Material nodes must match the Shader Model of your project in order to apply them to your assets, meaning that material nodes' shader inputs must match the intended target shader input.

Note: To properly apply materials to your scene, ensure that:
• You've selected the required user shader in the **Shaders** palette, for example, **Principled BRDF**.
•You have channels created and connected to the appropriate inputs of the shader that match the streams/material outputs of your materials. See Creating Multiple Channels from Presets for auto-creating and connecting channels.

Note: To get the best visual result, ensure you have an environment light. See Configuring the Lighting.

Batch Connection Method

To apply a material to an asset using the Mari's batch connection method:

1. Ensure a Material node is placed in the Node Graph.
2. Select any Channel merge node.
3. Press the **Tab** key and start typing 'Multi', then select **Multi-Channel Merge**.
   This automatically batch connects the Multi-Channel Merge node to the Channel merge nodes and Channel nodes.

4. Connect any Material node's output to its corresponding Multi Channel Merge node's **Over** Input. For example, **Metl** to **Over Metl**.
   This automatically batch connects the Material node's outputs to the Multi Channel Merge node's **Over** inputs.
   This applies the material to your asset.

Note: You can also use the batch connection method to connect multi-channel nodes to Output nodes when working inside Material nodes. For more information, see Connecting Multi-Channel Nodes.

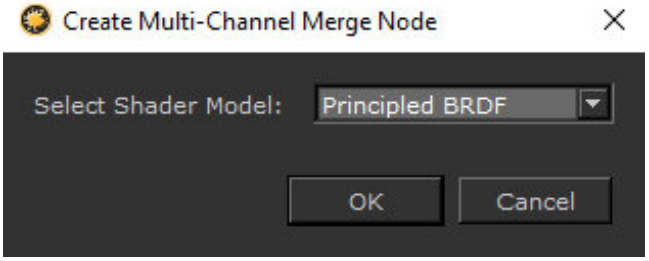
Video: Watch Batch Connection.

Note: A material node is not a paintable node but you can use masks to choose how to apply the material to your asset.

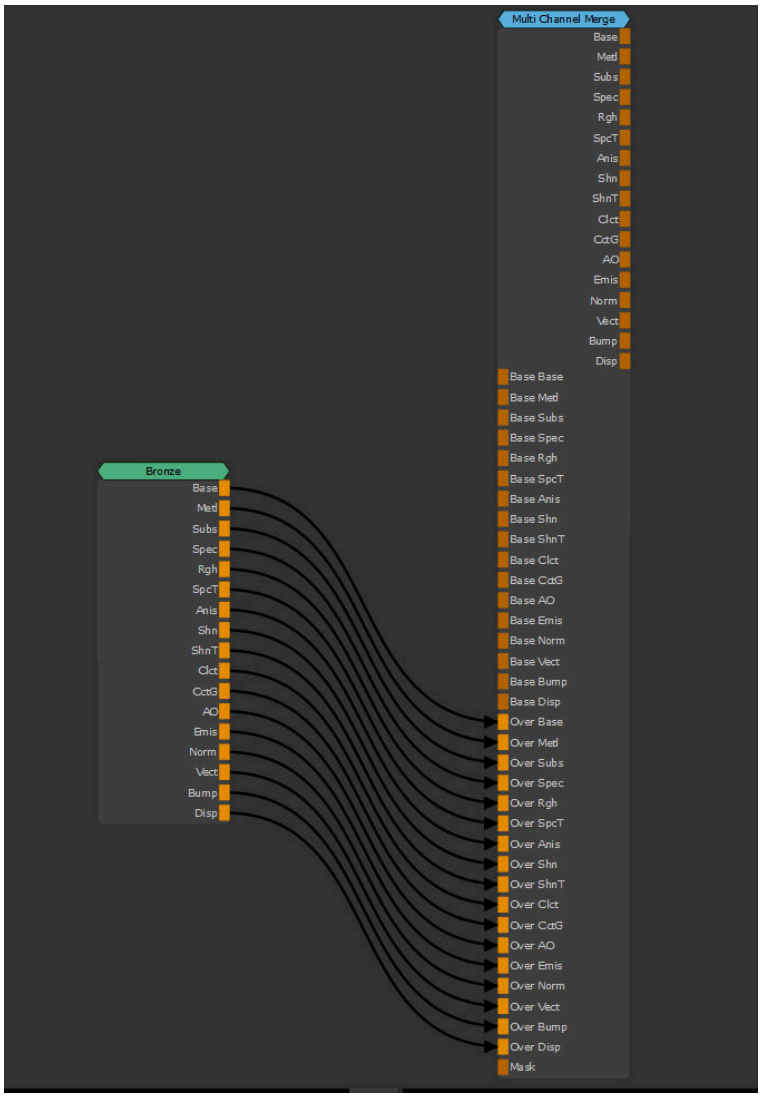
Manual Connection Method

To apply a material to an asset using Mari's manual connection method:

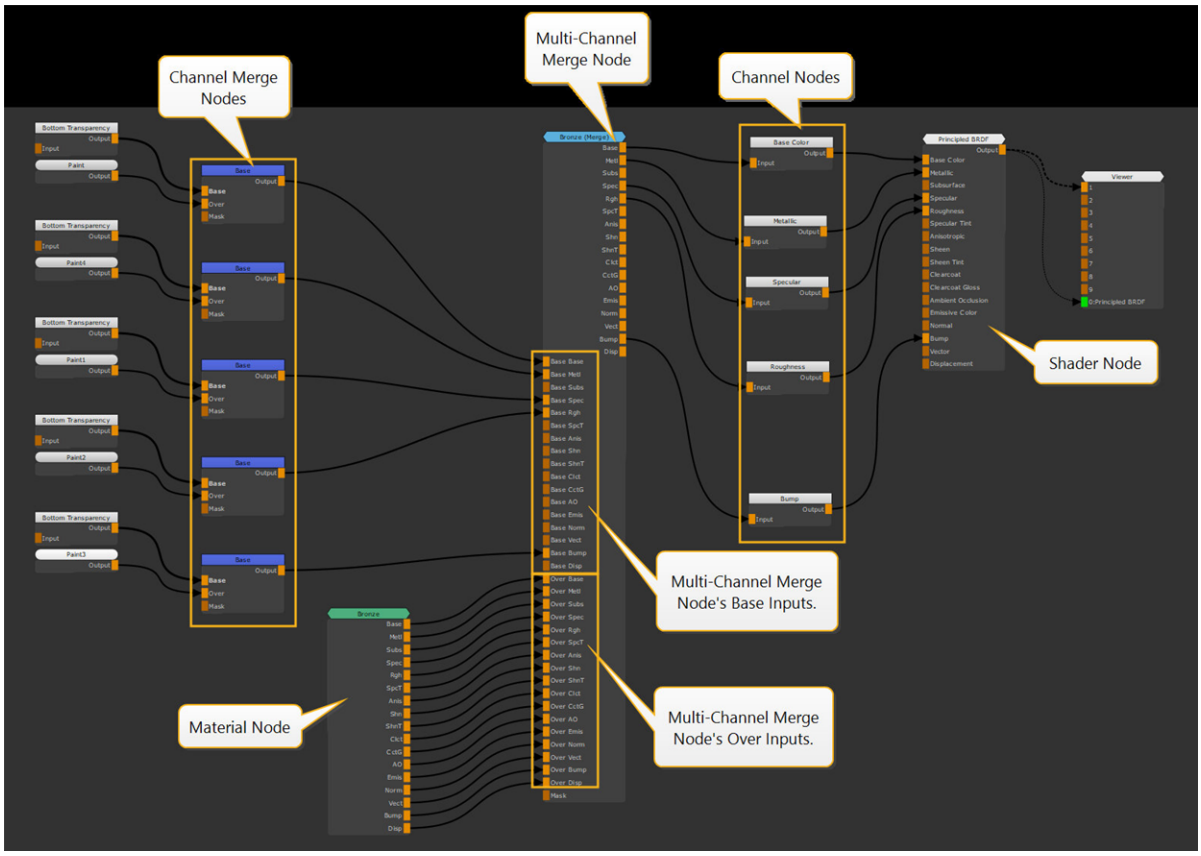
1. Ensure a Material node is placed in the Node Graph.
2. Press the **Tab** key, start typing 'Multi', and select **Multi-Channel Merge**.
   Alternatively, right-click in the Node Graph and from the menu that opens, navigate to **Nodes** > **Layer** > **Multi-Channel Merge**.
   This opens the **Create Multi-Channel Merge Node** dialog.



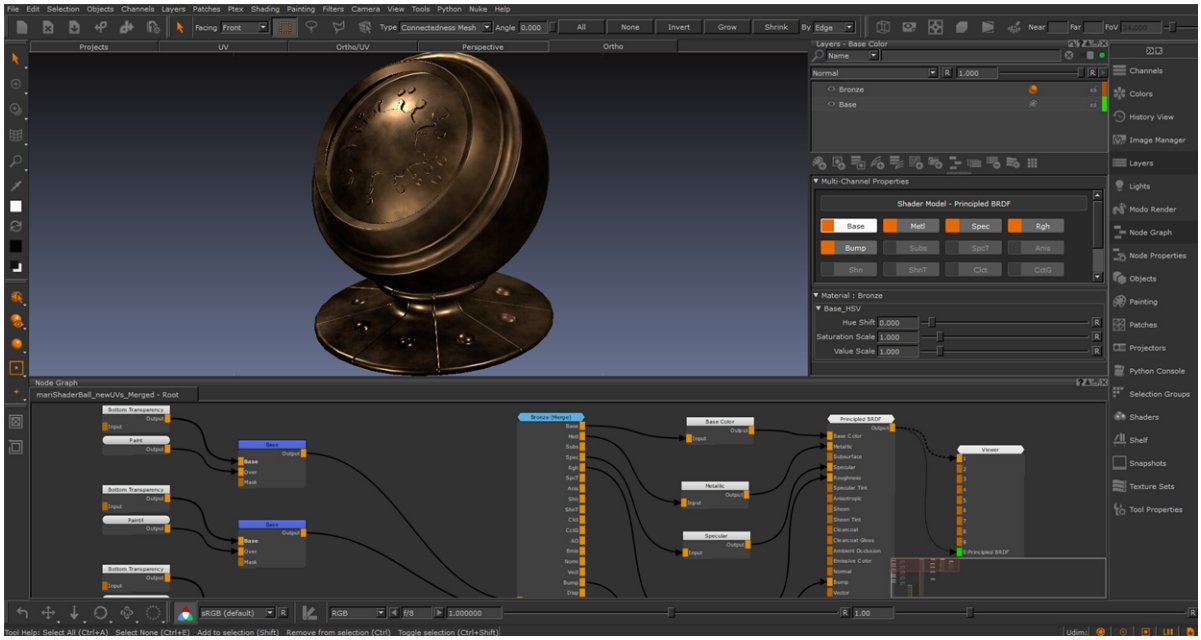
3. Select the shader that matches the shader model of your project, and click **OK**.
   This creates a Multi Channel Merge node.
4. Connect the first Material node's output to the Multi Channel Merge node's first **Over** Input.
   This automatically batch connects the Material node's outputs to the Multi Channel Merge node's **Over** inputs.



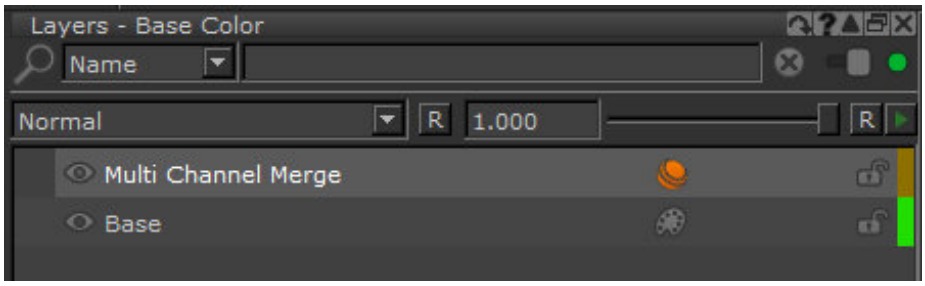
5. Apply the following connections:
   - Connect the Channel Merge node outputs to the relevant Multi Channel Merge node **Base** Inputs.
   - Connect the Multi Channel Merge node outputs to the required Channel nodes' inputs. For example: connect the Multi Channel Merge node **Base** output to the **Base Color** Channel node.
   - Connect the Channel nodes outputs to the Shader node inputs.



This applies the material to your asset.



Notice that a Material layer is added in the **Layers** palette.



Note: A material node is not a paintable node but you can use masks to choose how to apply the material to your asset.

Merging Multi-Channel Material Nodes in the Node Graph

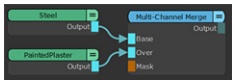
You can merge two multi-channel material nodes in the Node Graph by pressing **M** on the keyboard. This creates a new **Multi-Channel Merge** node.

Note: For more information on Multi-Channel Merge nodes, see Multi-Channel Merge Node.

To merge two multi-channel material nodes:

1. Select two multi-channel material nodes of the same shader model in the Node Graph.

2. Press **M** on the keyboard.



A new **Multi-Channel Merge** node of the same shader model is created.

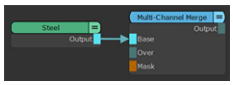
The first node you selected is connected to the **Base** connections, and the second selected multi-channel material node is connected to the **Over** connections.

Note: If the two multi-channel material nodes are not of the same shader model, a new **Multi-Channel Merge** node of the same shader model of the first selected multi-channel material node is created. The first selected multi-channel material node is connected to the **Base** connections, and the second selected multi-channel material remains disconnected.

A Multi-Channel Merge node can also be created using the **M** shortcut when just one multi-channel material node is selected.

1. Select one multi-channel material node in the Node Graph.

2. Press **M** on the keyboard.



A new **Multi-Channel Merge** node of the same shader model is created.

The node is connected to the **Base** connections.

Authoring Material variations using the Switch node

You can use the Switch Node to switch between Material variations. To do this, you'll need to set up multiple inputs in your node graph.

If you promote the **activeInput** knob to the top level of a group, material, or Smart Mask, the Active Input dropdown also appears within that node for easy access.

To learn about the Switch node, see Variations with the Switch node

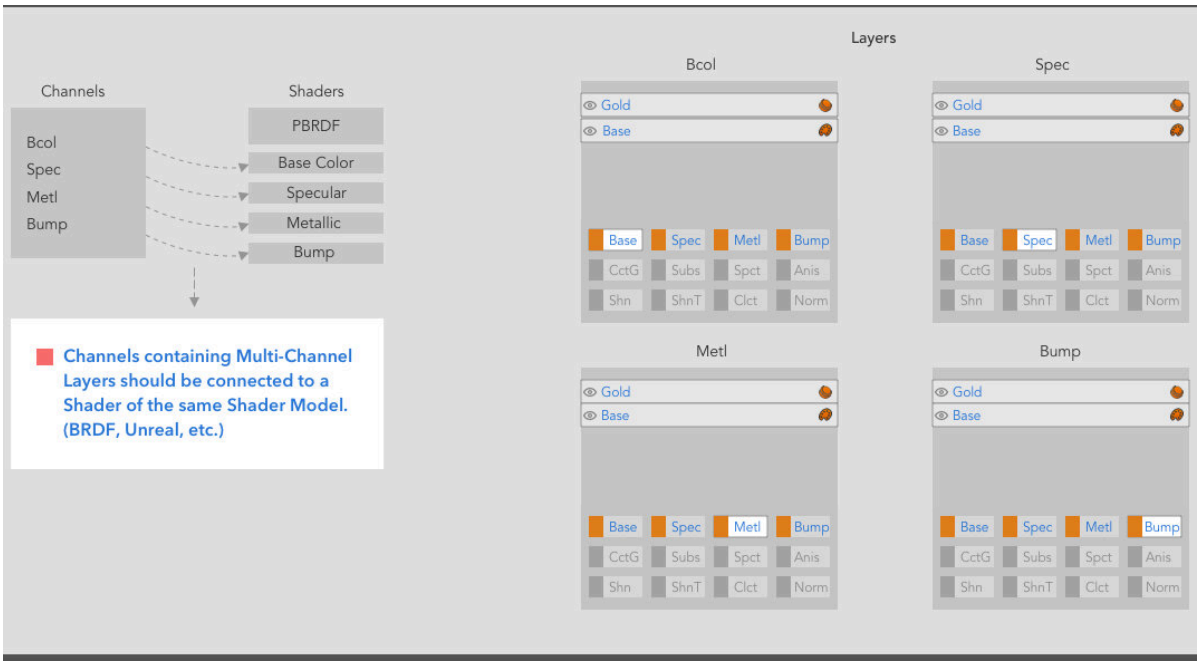
To learn about promoting knobs, see Exposing Material Properties Using the Promote Button.

Multi-Channel Layer Workflow Rules

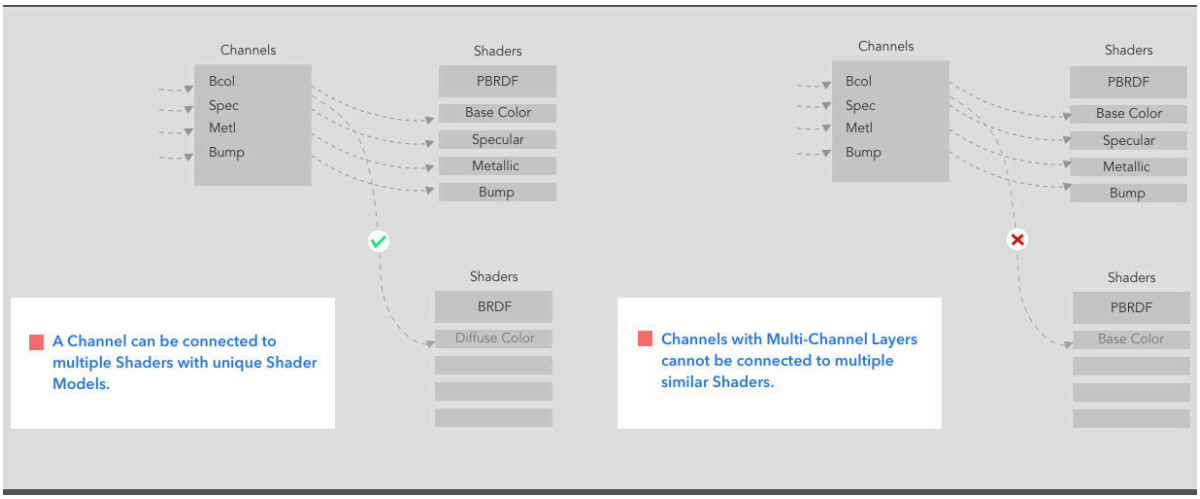
Note: These rules are only for layers and not applicable for Node Graph workflows.

Shader Models

Channels containing multi-channel layers should be connected to a shader of the same shader model.

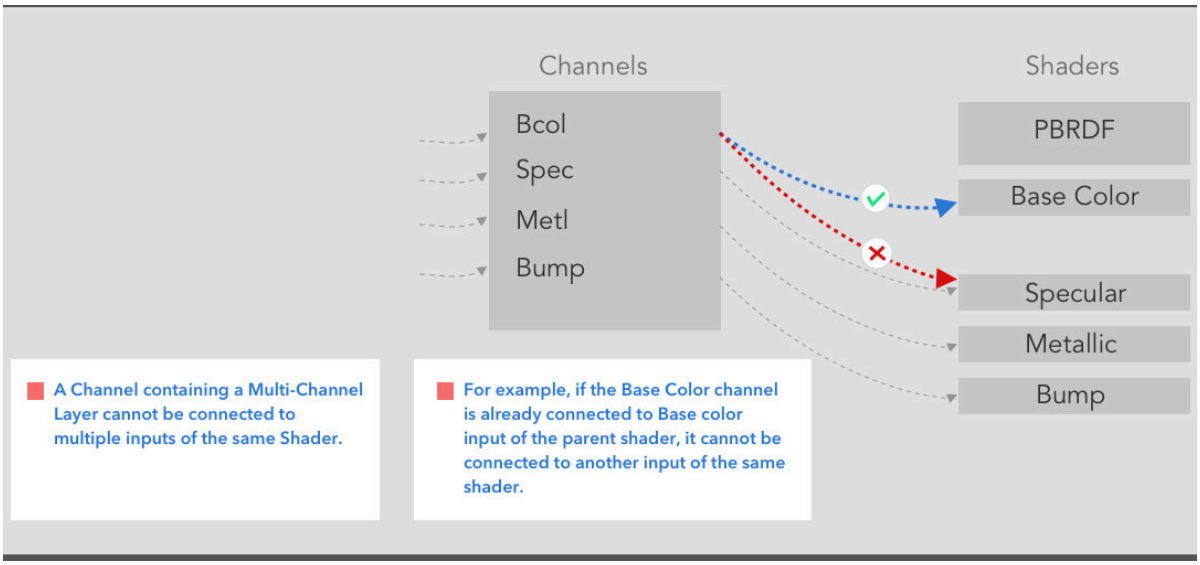


Channels containing multi-channel layers can be connected to multiple shaders with unique shader models but cannot be connected to multiple similar shaders.



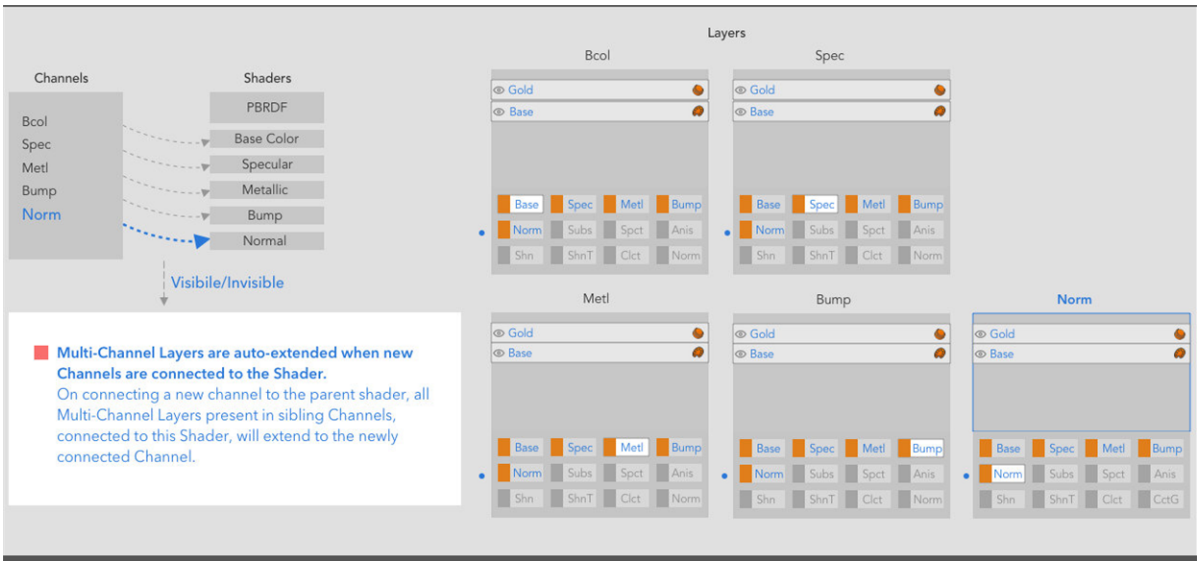
Shader Inputs

A channel containing a multi-channel layer cannot be connected to multiple inputs of the same shader.



Connecting New Channels to Shaders

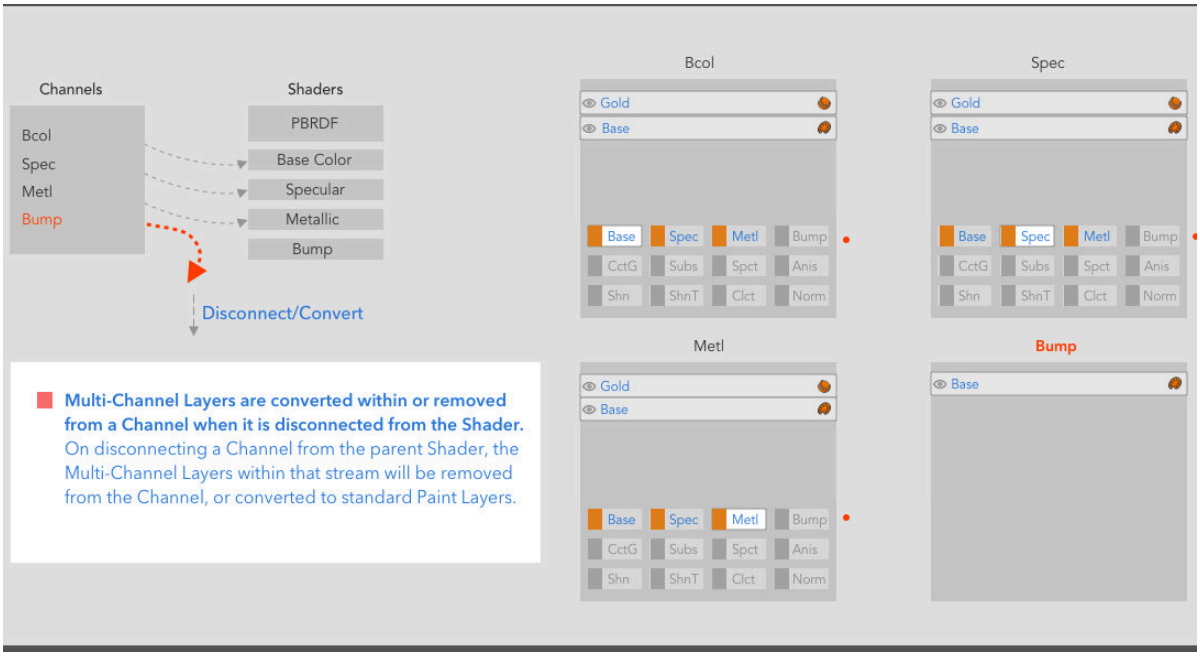
Multi-channel layers are auto-extended when new channels are connected to the shader.



Article: Read Q100488: Auto Extension and Auto Retraction of materials to learn more about auto-extension behaviour in Mari.

Disconnecting From a Shader

Multi-channel layers are converted within or removed from a channel when it is disconnected from the shader.



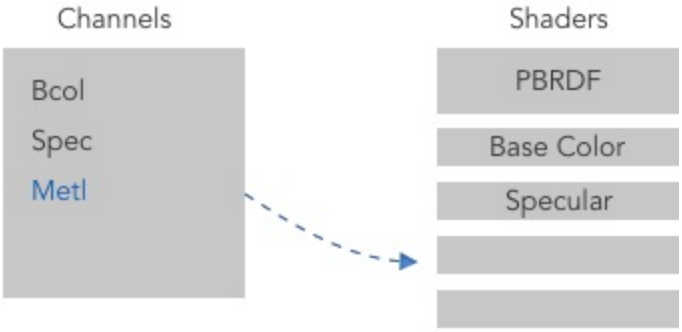
Use Cases

Auto-Extension Use Case

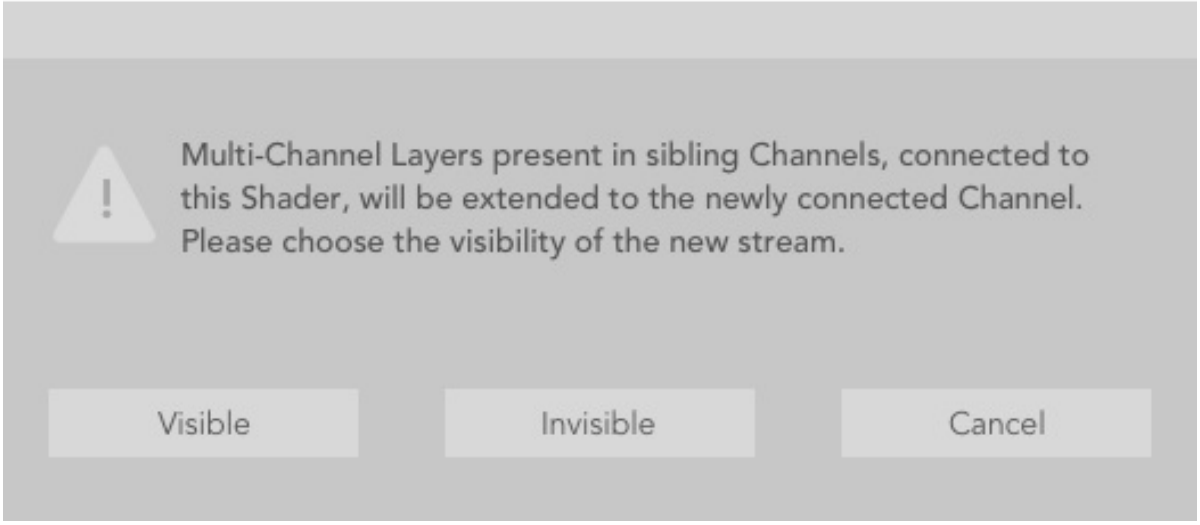
Multi-channel layers are auto-extended when new channels are connected to the shader.

On connecting the new channel to the downstream shader, it should recognize the existing materials in layers and auto-extend to the new stream (connected channel) in the **Multi-Channel Properties**. The new stream, should be **Visible** or **Invisible** depending on the option selected in the dialog.

3) Create a new Channel - Metallic
and connect it to the PBRDF Shader.



4) While connecting the new channel to the shader, the following dialog appears.



Note: When a new channel is connected to a shader with no upstream multi-channel layers, the dialog will not appear as there is nothing to extend.

a) **Visible** - The new stream introduced to the existing Multi-Channel Layers will be enabled and Visible to the Shader.

Layers - Metl

b) **Invisible** - The new stream introduced to the existing Multi-Channel Layers will be disabled and Invisible to the Shader.

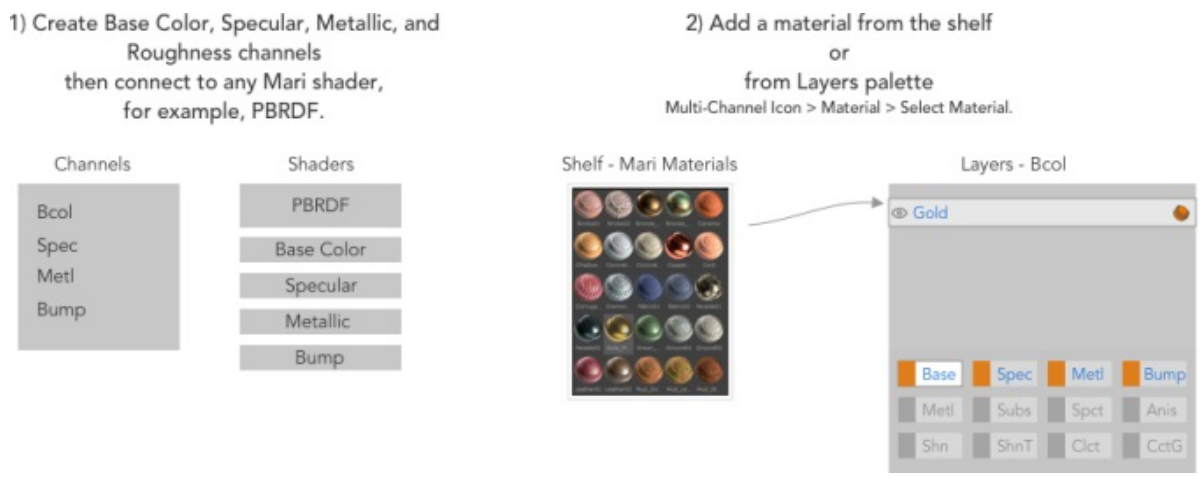
Layers - Metl

Multi-channel layers are converted within or removed from a channel when it is disconnected from the shader.

On disconnecting a channel from the downstream shader, its multi-channel layers should be disconnected, converted into standard layers, or removed depending on the option selected in the dialog.

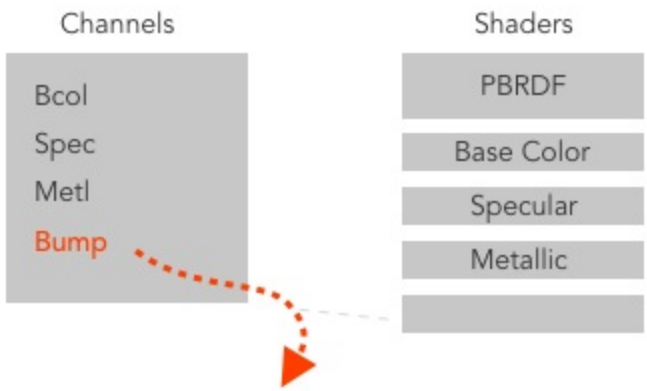
Retraction Use Cases

Case 1 - Material Layer

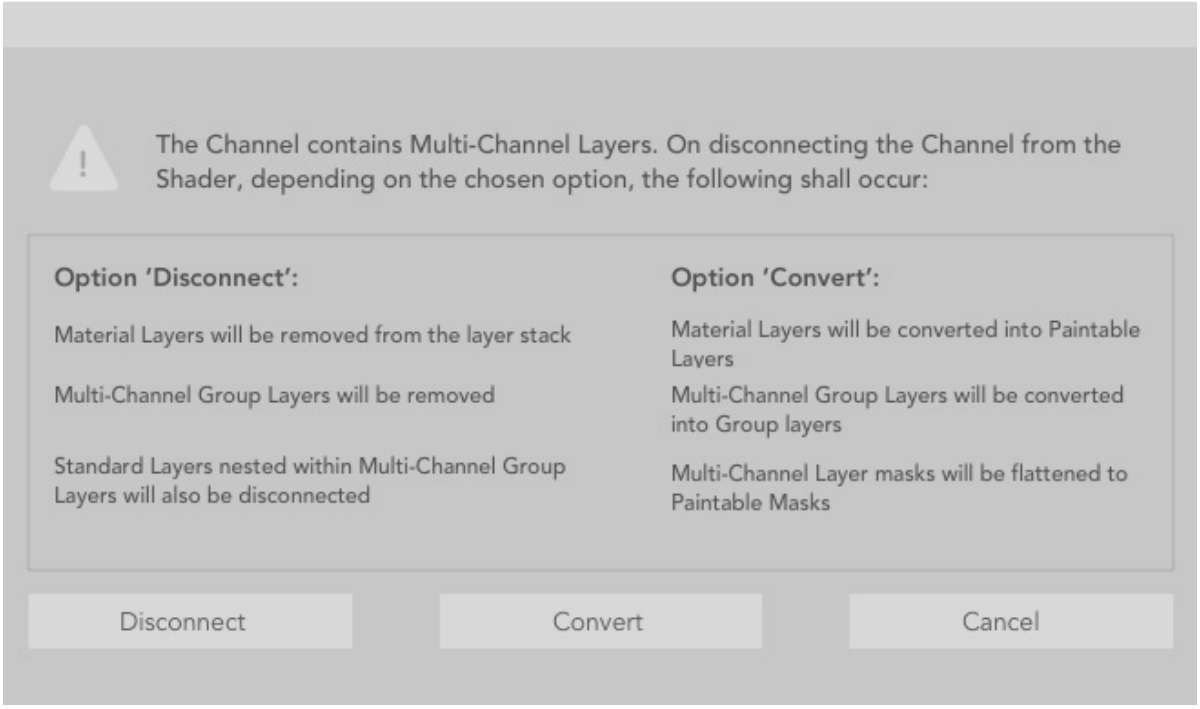


3) In the Shaders palette, disconnect a Channel from the PBRDF Shader.

(For example, remove the Bump Channel from the Shader.)



When disconnecting a Channel from the Shader the following dialog appears.



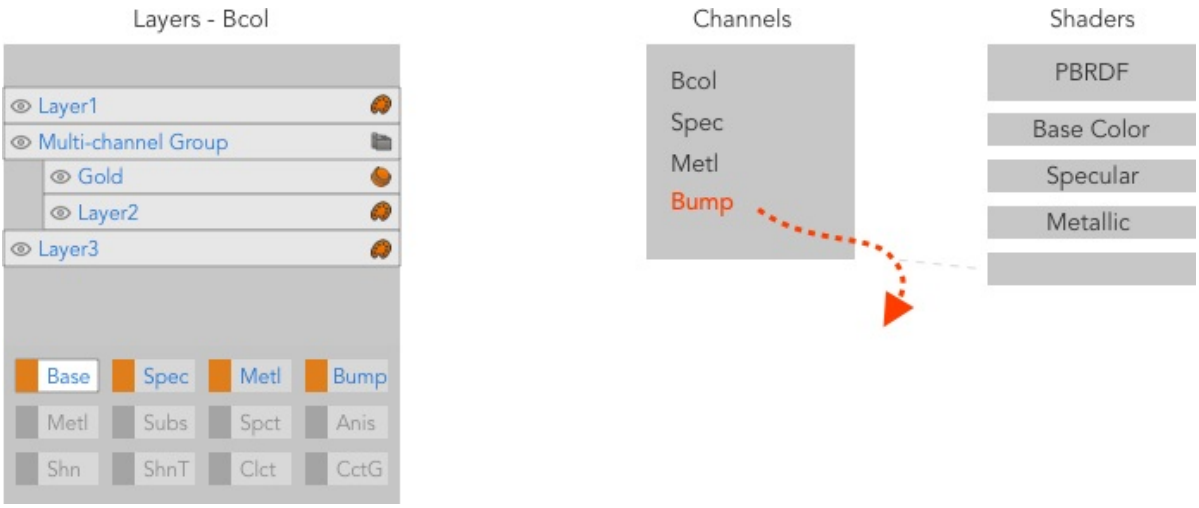
a) **Disconnect** - Material Layers will be removed from the Bump Channel.

Layers - Bump

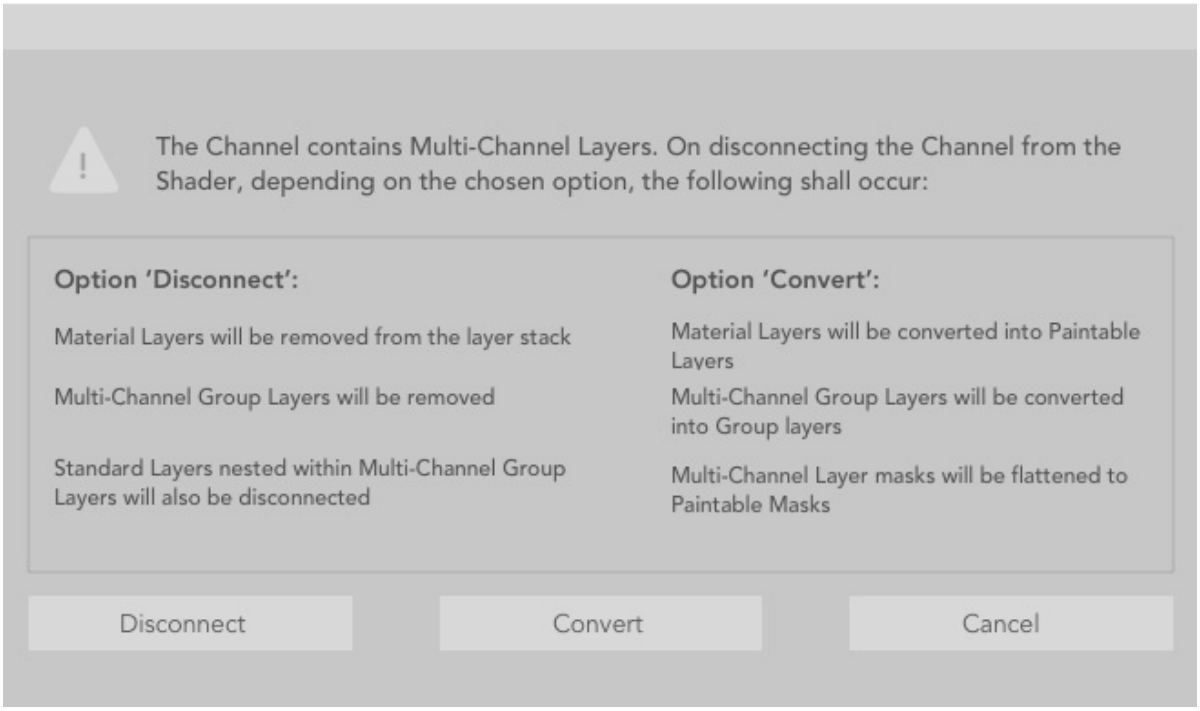
b) **Convert** - Material Layers will be converted to standard Paint Layers.

Layers - Bump

Case 2 - Material Layer with Multi-Channel Group



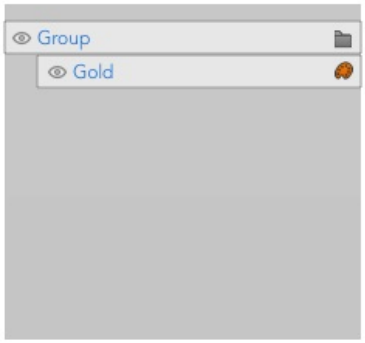
When disconnecting a Channel from the Shader the following dialog appears.



a) **Disconnect** - Both Multi-Channel and Material Layers will be removed from the Bump Channel.

Layers - Bump

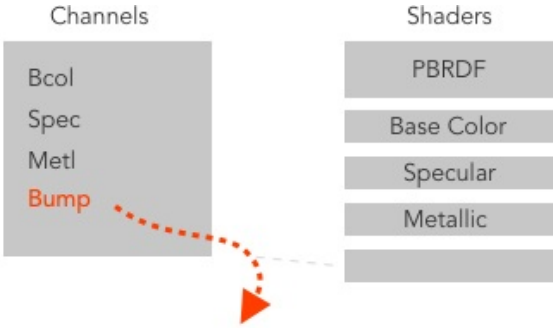
b) **Convert** - The Multi-Channel Group Layer will be converted to a standard Group Layer. The Material Layer will be converted to a standard Paintable Layer.



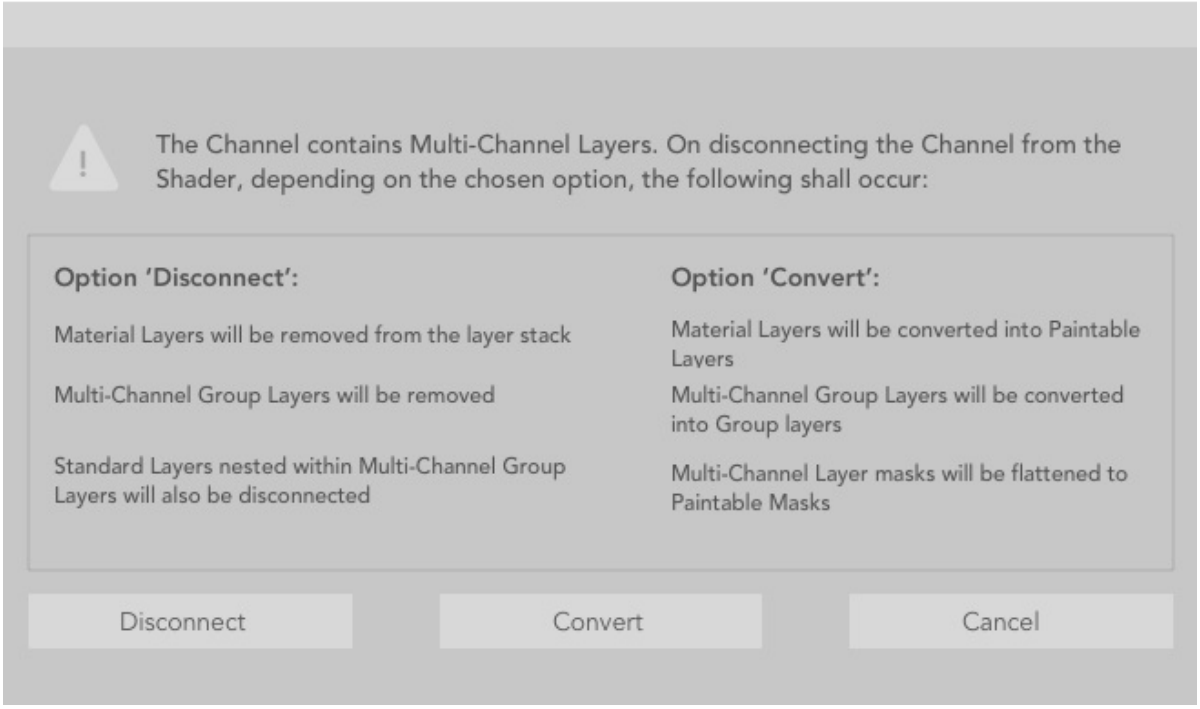
Layers - Bump

Case 3 - Multi-Channel Group with Standard Paintable Layers

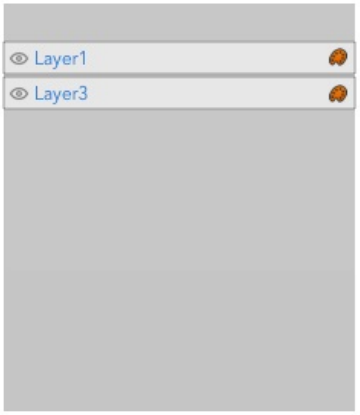
Layers - Bcol



When disconnecting a Channel from the Shader the following dialog appears.



a) **Disconnect** - The Material and standard Paint Layers within the Multi-Channel Group will be removed.



Layers - Bump

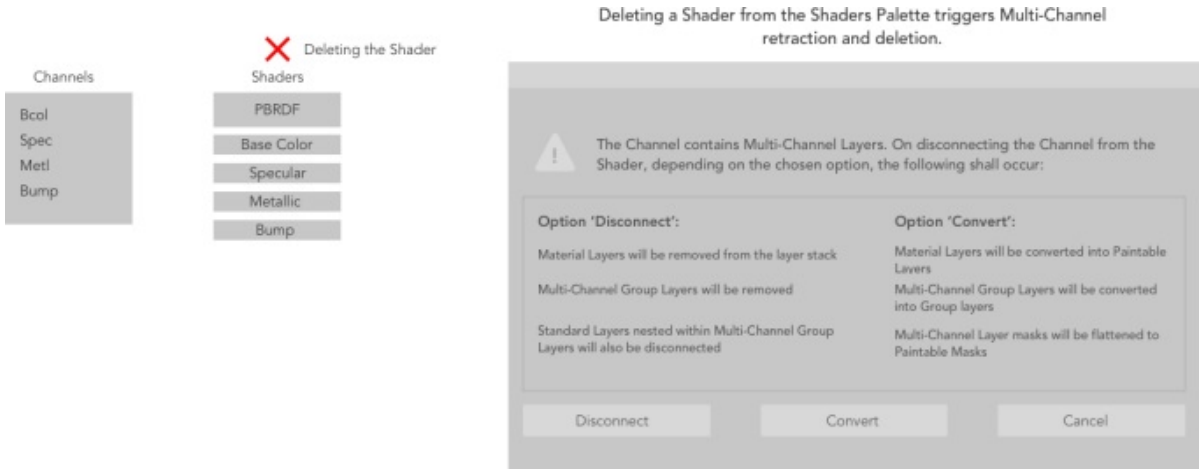
b) **Convert** - The Material and Multi-Channel Layers will be converted to standard Paint and Group Layers respectively.

Layers - Bump

Deleting Shaders

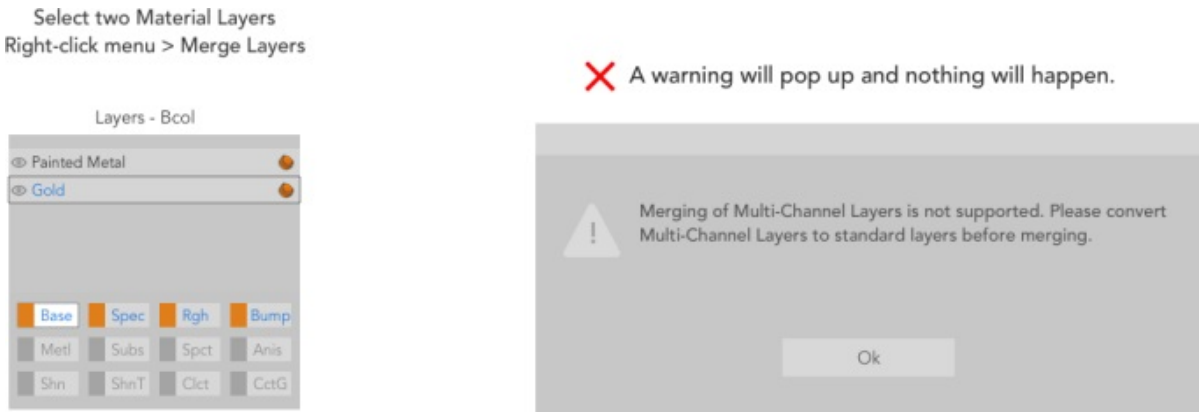
Deleting a shader will invoke retraction of multi-channel layers.

Deleting a shader from the **Shaders** palette triggers multi-channel retraction, as described above. The Retraction dialog sets the behavior for all Channels orphaned on shader deletion.



Merging Multi-Channel Layers

Merging multi-channel layers is not supported.



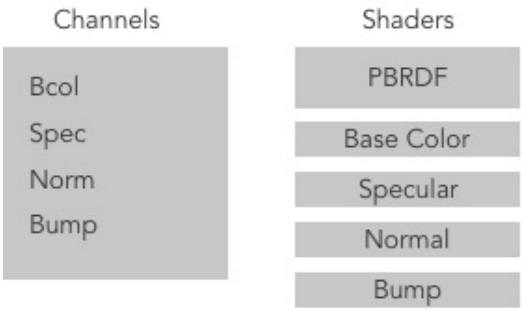
Grouping Materials

Materials cannot be grouped within a standard group layer.

3) Material Layers can be grouped with the new Multi-Channel Group Layer.

4) Material Layers cannot be grouped with a standard Group Layer.

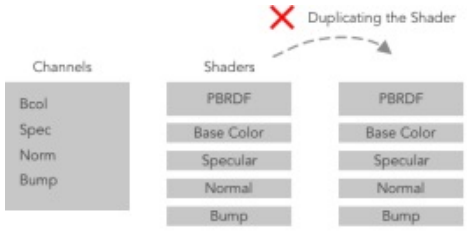
1) Create necessary channels and connect them to any Mari shader, for example, PBRDF.

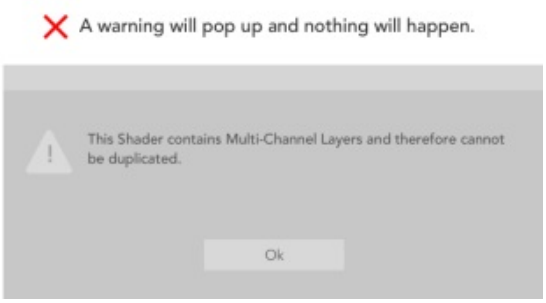


2) Add a Material from the Shelf or Layers palettes.

Duplicating Shaders

Shaders with upstream multi-channel layers cannot be duplicated.





Filtering Images and Paint

Mari includes a set of filters you can use to transform the paint on your model. You can apply filters to:

- paint in the current channel,
- only the current unbaked paint (the contents of the paint buffer), or
- an image within the **Image Manager**.

Unlike adjustment filters, the **Filters** accessed from the menubar apply a transformation to one of the following:

- The paint or image on the paint buffer
- The current paint target
- Selected patches on the current paint target (in patch selection mode).
- Selected faces on the current paint target (in face selection mode).

Though many of the filters available are the same or similar, those in the **Filters** menu do not affect the layer stack beyond what you choose to apply the filter to. In addition, you can preview your changes on the canvas before applying them.

There are many filters to choose from, and a complete list is provided in Available Filters, but a brief example of the filters Mari can apply are:

- Blurs to the painting,
- Inverted colors,
- Changes to hue, saturation, or value,
- Removal of specific color channels,
- Application of OCIO colorspace, and
- Added noise.

Every filter has the option to use a mask to control what part of the painting it affects. Depending on what you are applying the filter to, you can mask specific channels or apply an image from the **Image Manager** to create a mask.

Some filters do not require any configuration. For example, the **Edge Detect** filter has no user-configurable settings. Other filters provide a wide range of configuration options, including the ability to create, manage and load preset filters. For example, the **Color Curves** filter gives you very fine control over the contents of the RGB color components.

As well as using Mari's pre-defined filters to edit the paint on a the paint target or in the paint buffer, you can use them as part of a custom color profile. These filters only affect how the colors display on-screen (rather than actually changing the painting itself). See Managing Colors for details on setting up custom color profiles.

To make it easier, Mari provides an interactive preview of each filter. As you change the settings for the filter, the main canvas updates to show a preview of how the filter looks.

Tip: Filter previews are not available for scalar channels.

Mari also offers filters in 3D space, which do not experience artifacts due to boundaries such as patch edges. These filters behave similarly to traditional filters but are object-based. These filters can be accessed from the **Objects** > **Generate** menu.

Video: To learn how to use Filters, watch Understanding Mari Filters.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Selecting and Applying Your Filter

1. Start Mari and open your project.
2. For best results from the filter preview, set the lighting to flat, so the effects from the filter aren't lost in the effects of the lighting.
3. If you plan to apply a filter to the contents of a specific channel, make sure it is selected in the **Channels** palette, and that you select the **Current Paint Target** option.

4. If you plan to apply a filter to an image, open it from the **Image Manager** by double-clicking on the image before selecting the filter from the **Filters** menu.

> **Note:** A preview window for the image opens when you double-click it in the **Image Manager**.

5. If you plan to mask an area of the **Current Paint Target** using the **Marquee Select** tool, draw the area you want to mask.

> **Note:** Using a **Marquee Selection** as a mask is only possible when applying a filter to the **Current Paint Target**.

6. Select the filter you want from the **Filters** menu (see the table of Available Filters).

   The dialog box for your selected filter displays.

   Each filter has a different dialog box, with applicable **Options**.

7. Select **Options** for the filter.

   If you've already created a preset, select it from the dropdown menu to load it. If you want to save your current options as a preset, click to enter a preset name. You can always delete this by selecting .

8. Select whether the filter should be applied to:
   - everything on the **Current Paint Target** (not including any unbaked paint in the buffer),
   - only the **Current Painting** (that is, the unbaked paint in the buffer), or
   - the **Selected Images** (in the **Image Manager**).
9. Select whether to use a **Mask**:
   - If you're applying the filter to the **Current Paint Target**, you can use the any channel in the project as a mask.
   - If you're applying the filter to the **Current Painting**, you can use the **Marquee Selection** tool to draw an area that you want to mask. This area should be drawn before selecting the filter.

     Alternatively, you can use an image from the **Image Manager**.
   - If you're applying the filter to **Selected Images**, you can use any image in the **Image Manager** as a mask.

Note: If you choose to use a channel or an image from the **Image Manager** as a mask, an **Invert** checkbox and a **Mask Curve** editor displays. For more information on using these, see the next section on
Available Filters.

10. Toggle the **Preview** switch to see the filter's effects before applying them.

Tip: Click on the control to toggle the filter on/off while you watch the effects. The preview always displays on the whole model - even if you have the filter set to apply to the **Current Painting** or **Selected Images**.

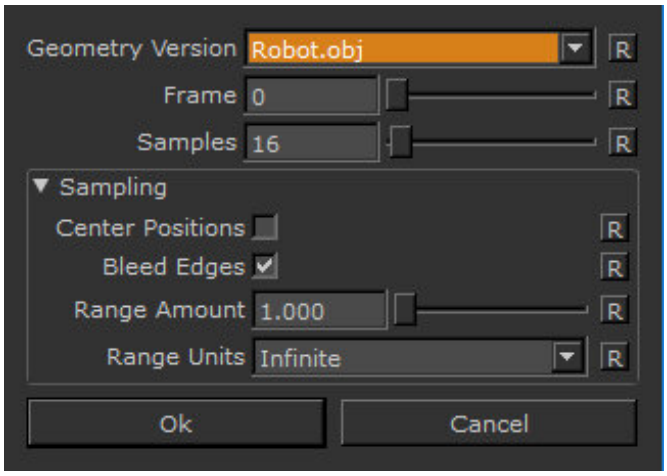
11. Click **Apply**.

Note: If you want a filter to apply only to a specific layer or layers of a channel, consider using the adjustment filters in the Layers palette. For more information on adjustment filters and how these work on individual layers, see Layer Types and Examples.

Experiment: The best way to get a feel for what the individual filters do is to play around with the options with the filter preview on. Take a few minutes to review details of the various Available Filters. Try applying some of them to a channel. You don't actually have to save the filtered effect - turn the **Preview** option on, see how it looks, and then **Cancel** to reject the changes.

Selecting and Applying Object-based Filters

1. Start Mari and open your project.
2. Select the filter you want from the **Objects** > **Generate** menu (see the table of Available Filters).
   The dialog box displays for your selected filter.



Each filter has a different dialog box, with applicable options.

3. Once you're happy with the filter settings, click **Ok**.

   The filter is applied to the entire object, not just specific patches.

Tip: For more information about the individual filter settings, refer to the *Mari Reference Guide*.

Available Filters

We mentioned previously that an **Invert** checkbox and a **Mask Curve** editor displays in certain situations. These options are universal to all filter operations that use a channel or an image from the **Image Manager** as a mask.

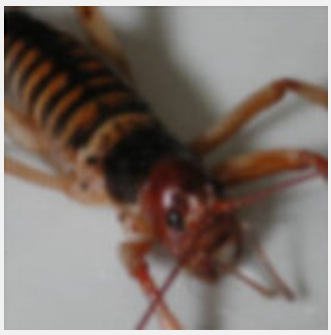
The **Invert** checkbox decides whether Mari inverts the input from the mask. If this is selected, the mask data is black on white rather than white on black.

The **Mask Curve** editor sets how the channel masking works. The x axis is the color of the source pixel (black to the left and white to the right). The y axis is how paintable the target pixel is. By default, the curve runs from the bottom left up to the top right. This means that black pixels are totally masked, white pixels are totally paintable, and there's a constant curve between the two. Use this if you want to change what pixel values are masked.

The following table describes and shows examples of each of the filters that come with Mari, applied to the following image:

Filters Menu

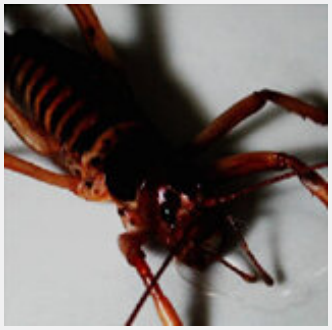
Filter	Description	Options	Example
Invert	Inverts the colors in the selection. Replaces colors with their "opposite" in the color chart. For example, replaces a color with a float value of 0.3 with a color with float value 0.7.	You can set whether to invert the Alpha channel as well as the colors.	
Luminosity	Outputs just the luminosity value of each pixel in the selection. That is, it outputs a grayscale image based on the brightness of each of the pixels in the original image.	None.	

Filter	Description	Options	Example
Blur	Applies a standard blur to the selection.	You can select the blur radius - the number of pixels Mari uses when calculating the blur value for each pixel. The higher the radius, the more blurred the results. If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.	
Soften Blur	Applies a subtle softening blur. This is a quick, predefined blur filter. If you need more control over the degree or type of blurring, use either the standard **Blur** or **Gaussian Blur** filters.	None.	
Gaussian Blur 1D	Applies a Gaussian blur. Compared to the standard **Blur** filter, this gives you much finer control over the degree of blurring, and the option of using much higher blur values.	You can set both the **Radius** and the **Amount**. The higher the radius, the more blurred the results. Amount defines how much to mix the blurred result with the original. If you use a larger blur radius, you may get slight	

Filter	Description	Options	Example
		lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.	
Gaussian Blur 2D	Applies a Gaussian blur. In contrast to the **Gaussian Blur 1D** filter, **Gaussian Blur 2D** allows you to blur in two dimensions, with the ability to seamlessly blur tiling images.	You can set both the **Radius** and whether or not to **Warp** the edges. The higher the radius, the more blurred the results. Enable **Warp** to allow the edges to blur seamlessly - perfect for images that will be tiled. You also have **Masking** options available to affect which areas or which channels of the image data are affected.	
Gamma	Changes the gamma levels in the selection.	**Do Inverse** reverses the gamma conversion. That is, a higher setting for **Gamma** results in a more washed-out looking image. Selecting **Use default gamma** disables the other controls and uses your default setting for gamma (as set in the **Painting** tab of the **Mari Preferences** dialog box, accessible from the **Edit** menu). Otherwise, select a particular gamma level.	

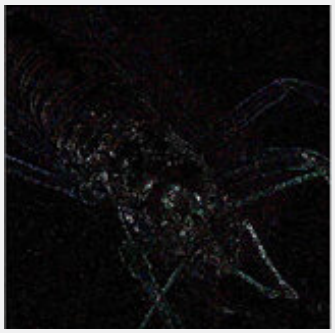
Filter	Description	Options	Example
sRGB To Linear	Applies an sRGB to linear colorspace conversion.	Checking **Invert** applies a linear to sRGB colorspace conversion.	
Hue	Changes the hue, saturation, and value of the colors on the model.	To change the hue of the colors, set the value for **Hue**. This moves the colors around the color wheel. The rotation value is the degree around the color wheel that each color shifts, between 0 and 360 (which both correspond to the original colors). The options to change the **Saturation** are: • **Saturation** - how large the saturation is. Select a multiplier from 0 to 2, where 1.00 is the original saturation value. • **Offset** - an offset value to add to the initial saturation value, before applying the scale. You can also edit the **Saturation** curve. This controls how the filter translates current saturation values to new values. For	

Filter	Description	Options	Example
		each input value along the bottom of the graph, the output value is set from the position of the line above that point. Edit the curve by: • pulling the anchor points • clicking between anchor points to add a new point • right-clicking to select from a set of preset values. You can edit the **Scale**, **Offset** and **Value** curve for the value of the colors in the same way.	
Brightness	Changes the brightness of the paint on the model, either overall or for specific channels. You can also change the contrast at the same time.	You can either change the brightness of all color channels at once, or one-by-one. Changing the main **Brightness** control changes (overwrites) any changes you make to brightness values for individual color channels.	
Contrast	Changes the contrast of the paint.	You can select the amount of contrast using the slider.	

Filter	Description	Options	Example
Clamp	Clamps color values to lie within the specified upper and lower values.	You can set the **Upper Value** and **Lower Value** by adjusting the sliders or entering values in the text fieldes. Select the individual **Color Component** to clamp in the dropdown menu.	
Levels	Changes the color levels in the paint. You can change the white, midtone and black points for each color channel or all together.	As you change the settings, the display at the top of the dialog box updates to show the current white point, midtone, and black point. You can set the: • **Color Component** - whether this is affecting the overall **Value** of the color, or the specific color channel (**Red**, **Green**, or **Blue**). • **White Point** - the upper limit of displayed color intensity. Areas with a value higher than this are mapped to 1 (white). • **Midtone** - the middle point between white and black. Mari remaps the values so that this is the middle of the range between the white point and black point. Moving this pushes the values towards that end of the	

Filter	Description	Options	Example
		spectrum. • **Black Point** - the lower limit of displayed color intensity. Areas with a value lower than this are mapped to 0 (black). • **White Output Level** - how strongly to output the white in the filtered image. Higher values show the entire white output, lower values show decreasing amounts. • **Black Output Level** - how strongly to output the black in the filtered image. Higher values show the entire black output, lower values show decreasing amounts.	
Color Curves	Changes the color curve for each color channel (RGB) in the painting.	You can also edit each color curve, controlling how each color channel appears on screen. This controls how the filter translates current color values to new values. For each input value along the bottom of the graph, the output value is set from the position of the line above that point. Edit the curve by: • pulling the anchor points • clicking between anchor points to add a new anchor point	

Filter	Description	Options	Example
		• right-clicking to select from a set of preset values.	
Color Switches	Turns individual color channels on or off.	Use the checkboxes to select which color channels display. When Mari applies the filter, it removes any unchecked color channels from the painting.	
Tone Mapping	Varies the "exposure" of the painting.	You can set the: • **White Point** - the upper limit of displayed color intensity; any colors of greater intensity map to white. • **Exposure** - lower exposures give darker images; higher exposures give lighter ones.	
Copy Channel	Copies the value from one RGB color channel to the other two. The result is a grayscale image with the intensity values from the selected channel.	Select the source channel from the list. When you apply the filter, Mari copies the selected channel over the other two channels.	

Filter	Description	Options	Example
Premultiply Alpha	Either pre- or post-multiplies the alpha in the selected image. If you are painting using an image without premultiplied alpha onto one with it, use this filter to perform the premultiplication, so the images match and you avoid lines around the outside of the patch. (Postmultiply works the same, but in reverse (removes premultiplication to match images that do not have premultiplied alpha).	If the **Postmultiply** box is checked, this acts in reverse (that is, removes premultiplied alpha).	
Edge Detect	Finds "edges" (transitions between colors) in the image and heightens them.	None.	

Filter	Description	Options	Example
Emboss	Applies an “emboss” effect, so that textures look “raised” on the surface.	None.	
Sharpen	Removes 'fuzziness' and clarifies detail in the image.	You can set the amount of sharpening. Higher values mean more sharpening, while a value of 0.00 is the picture without the filter applied.	
High Pass	Removes low frequency information from the image while maintaining higher frequency detail. This can be useful for quickly creating specular maps where only the higher frequency image highlights are maintained.	You can set the kernel size (the amount of filtering) by adjusting the **Radius**. If you use a large radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles patch edges. This filter is not supported with Ptex channels.	

Filter	Description	Options	Example
OCIO Colorspace	Applies a colorspace correction to the current channel or painting.	Set the colorspace of the current channel or painting in the **Input Colorspace** dropdown. Set the colorspace correction to apply to the current channel or painting in the **Output Colorspace** dropdown.	
Add Noise	Applies noise to the current channel or painting.	Use the **Amount** slider to adjust how much noise is applied and select the component you want to affect from the **Color Component** dropdown. For additive grayscale noise, that modifies the intensity while retaining the original color, check the **Grayscale** checkbox. You can regenerate the noise using different random samples by dragging on the **Seed** slider. Adjust the size of the noise applied by dragging on the **Size** slider.	

Objects Menu

Filter	Description	Options	Example
Gaussian Blur	Applies a Gaussian blur. Compared to the **Filters** menu **Gaussian Blur**, this allows you to apply a blur effect over whole objects in 3D space.	Set the **Geometry Version** so it knows what object to which the blur should be applied. For animated objects, **Frame** specifies on which frame the blur should be applied to the object.	
Displacement	Allows you to generate a displacement map, using a high polygon source model and a low polygon destination model to calculate the displacement.	Select the source **Object**, **Version**, **Size**, and the **Frame** (if animated) from the options provided for the high polygon model, and specify what destination version and frame to which the displacement should correlate on the low polygon model. Adjust the sampling options, such as **Interpolation** and **Sample Count** to specify how the displacement map is generated.	
Height	Allows you to generate a height map, using a high polygon source model and a low polygon destination model to calculate the height.	Select the source **Object**, **Version**, **Size**, and the **Frame** (if animated) from the options provided for the high polygon model, and specify what destination version and frame to which the height should correlate on the low polygon model. Adjust the sampling options,	

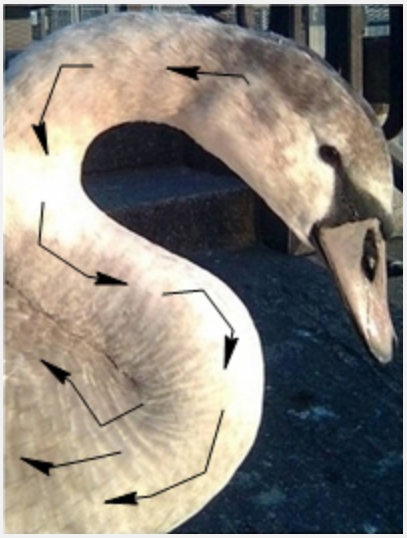
Filter	Description	Options	Example
		such as **Interpolation** and **Sample Count** to specify how the height map is generated.	

Vector Brush

The **Vector Paint** brush is useful for editing vector maps and painting directional information onto your models. With the vector-specific paint blends, paint is converted to tangent space. The vectors are then represented as diffuse RGB values. This results in specifically-colored paint strokes that represent color-encoded vectors.

In addition to using the **Vector Paint** brush, the basic **Paint** brush can be used together with vector-specific paint blending modes to achieve different effects.

Viewing vectors is especially important for the creation of flow maps and normal maps. Flow maps can assist in
creating realistic textures or materials that simulate direction-specific patterns, such as feathers, scales, fur, hair, and water flow (to name a few). These have a pattern that tends toward a particular direction, or lie against another object in a particular manner, your textures should reflect this. Once you have identified how you want the effect to appear on your model, you can create flow maps that provide this information.



Arrows indicate the direction of the feathers on the swan's body, which vectors should mimic.	Arrows show the direction of painted vectors, which indicate the flow of vectors.

For example, in the image on the left, the feathers on the swan are determined to be following the curve of the neck. This impacts the feather pattern, so that the vectors in the texture curves around to display the natural flow pattern. In the image on the right, the vectors have been painted on the image to show how this flow pattern might appear.

Painting Vectors

Begin by starting Mari and opening your project.

1. Make sure you have a paintable layer selected, then select the **Vector Paint** tool.
2. In the **Shaders** palette, ensure that the channel you are using for your vector paint is selected in the shaders **Vector** shader component.
3. If you want to see the directional markers while you paint, select the **Vector Inspector** icon in the **Vector** toolbar.
4. Select a brush from the **Shelf** palette. Once you have set up the brush options in the **Tool Properties** palette, paint in a direction across the surface of the model.

   A specific color is selected depending on the direction of your brush stroke.

Think of the four main colors as being the four cardinal points on a compass. These have specific colors to represent their directions. Strokes between these points are blends of the two colors. The four main colors are:

• Left to Right: (RGB: 1, 0.5, 0.5)

• Right to Left: (RGB: 0, 0.5, 0.5)

• Top to Bottom: (RGB: 0.5, 1, 0.5)

• Bottom to Top: (RGB: 0.5, 0, 0.5)

These colors may vary depending on the lighting model you are using and any adjustments that are selected, for instance the **Tangent to Screen** adjustment. You may find it useful to paint vectors while using a **Flat** lighting model to make accurate vector painting easier.

Modifying Vectors with Paint Blending Modes

Seven unique paint blending modes allow you to specify how the vectors are baked onto a model. These paint blending modes are not for exclusive use with the **Vector Paint** tool, but can also be used with the basic **Paint** tool. These modes are available in the paint blending mode dropdown menu in both the **Paint** and **Vector Paint** toolbars and the other paint tools, such as **CloneStamp** and **Paint Through**.

The **Paint Flow Vectors** and **Paint Normal Vectors** modes are an alpha blend between the vectors. The **Nudge Flow Vectors** and **Nudge Normal Vectors** modes add the vectors together to get the blend result. The paint flow and paint normal modes convert the painted vectors into the texture's tangent space before creating the associated blend. This conversion takes into account the orientation of the texture on the model's surface.

Note: Due to the fact that surface orientation plays such an important role, vectors baked down on a model may not be exactly the same as vectors painted. For example, a paint stroke from left to right on a model with a very high curvature may produce quite large changes in color when baked.

However, the **Scale Vectors** mode works differently to the above four modes. This mode uses the intensity of the color to control the length of the vector, and works best when painting in gray scale, with black representing short vectors, white representing long vectors, and gray being between the two. Paint **Scale Vectors** with the basic **Paint** tool.

The flow and normal vector modes refer to how the vectors are allowed to act in xyz space. Flow modes keep the vectors flat against the surface of the model. Normal modes allow the vectors to protrude outwards from the model in tangent z space. The type of vectors you want to paint may influence which paint blending modes you use. For example, flow modes are most often going to be used for flow maps, while you may prefer using normal modes for creating normal maps.

The **Add Normal Maps** and **Mix Normal Maps** modes blend two RGB-encoded vector layers with either the **Add** or **Mix** inputs and the **Over** input. Before the operation is performed the raw values are converted from the 0.0 to +1.0 colorspace-encoded range to the -1.0 to +1.0 vector range. The values are then added and normalized before converting back to the 0.0 to +1.0 colorspace-encoded range.

For more information on how to use these paint blending modes, see Paint Blending Modes in the section on Controlling the Way Mari Applies Paint to Your Model.

To adjust the paint blending mode, follow these steps:

1. With unbaked paint on your model, select one of the following modes from the Painting **Mode** dropdown menu:
   - **Paint Flow Vectors**,
   - **Paint Normal Vectors**,
   - **Nudge Flow Vectors**,
   - **Nudge Normal Vectors**,
   - **Scale Vectors**,
   - **Add Normal Maps**,
   - **Mix Normal Maps**, and
   - **Mix Reoriented Normal Maps**.
2. Selecting a paint blending mode can be done in either the **Paint** or **Vector Paint** toolbars, as the dropdown menu is in both. Alternatively, select them from the Painting **Mode** dropdown in the **Painting** palette.

> **Note:** As with other paint blending modes, the vector modes apply to paint that has not been baked down yet.

To use a layered workflow, where you paint multiple layers of vectors and blend them together, the normal painting steps do not produce the results you might have expected, due to Mari's layers using premultiplied alpha. To blend a vector layer with the vector layer beneath it in the layer stack:

1. Create a new layer.
2. Fill it with a neutral vector color, thus making the layer opaque. A neutral vector color for normals is 0.5, 0.5, 1.0, and for flow it is 0.5, 0.5, 0.5, as described in more detail in Painting Vectors.
3. Paint on the new layer.

   To blend with the layer below, add a layer mask and paint into it in order to show or hide the layer below.
4. Select a blend mode.

   A layer's alpha values are no longer affected by premultiplied alpha and the vector blending results for your layers should now be correct.

Inspecting Your Vector Work

The **Vector Inspector** tool allows you to see directional markers representing the vectors you are brushing onto the model. These markers ensure that you can see the effects of the vector brush or paint blending modes to achieve the effect you want. Vector lines can be turned on or off at any time, as well as adjusted to

change the properties, such as spacing. Like the paint buffer, the **Vector Inspector** can be moved around the canvas or rotated using
keyboard shortcuts.

To inspect your vector brush work, follow these steps:

1. Select the **Vector Inspector** tool .

   The directional markers appear.

> **Note:** The **Vector Inspector** displays the channel that is selected in the **Vector** shader component of your current shader. If there is nothing selected in the **Vector** shader component, the shader displays the diffuse channel.

2. You can move the buffer around the canvas by simply dragging it around. You can also use the **Tool Help**
   shortcuts to modify it.
3. The **Vector Inspector** toolbar includes the following options:
   - **Enable** - selecting this checkbox enables or disables the directional markers of the **Vector Inspector**. Note that while in the **Vector Inspector** tool, this is not the case. The markers remain on until you change to a

different tool. If you don't have the tool active and want to enable markers quickly, click the icon on the **Vector Painting** toolbar.

- **Spacing** - adjusts the space between the directional markers.
- **Length -** adjusts the length of the directional markers.
- **Start Color** - changes the color at the base of the directional marker.
- **End Color** - changes the color at the tip of the directional marker.
- **Vector Space** - sets whether the vector space is converted to either **Screen Space**, **Tangent Space** (no change), or **World Space.** Screen space is the most useful option for painting flow and normal maps.

4. In addition to the options on the toolbar, the following options are listed on the **Tool Properties** palette when the **Vector Inspector** tool is active:
   - **Rotation** - lets you adjust the rotation of the inspector buffer by either entering a value in the text field or adjusting the slider.
   - **Translation** - adjusts the translation (position) of the inspector buffer.

> **Note:** The **Rotation** and **Translation** options are both available in the **Tool Help** shortcuts whenever the **Vector Inspector** tool is active.

Adding Vector Shaders

There is a method for quickly setting up default vector shaders and channels, with vector layers in the **Layers** palette. This method is accessible from two locations in the UI and is described fully in the Adding Vector Shaders section.

This shader setup creates two default shaders in the **Shaders** palette for creating flow maps and normal maps. The **Vector (Screen Space)** shader is for viewing vector (colors) in tangent space, while the **Vector (World Space)** shader is for viewing vector (colors) in world space. These shaders come with channels already plugged into the shader component inputs by default to get you started.

To create vector brush default shaders, follow these steps:

1. Open the **Shaders** palette to manage newly-created shaders.
2. Create a default vector shader by either:
   - clicking the icon on the **Vector Painting** toolbar, or
   - navigating to **Python** | **Examples** > **Setup Vector Brush.**

   Clicking the icon or script automatically creates two shaders in the **Shaders** palette, three channels in the **Channels** palette, and the necessary layers for each channel in the **Layers** palette.

The default created shaders are **Vector (World Space)** and **Vector (Screen Space)**, and each of these have the newly-created channels set to the corresponding shader component inputs.

During shader setup, the following three channels are created in the **Channels** palette:

- **Vector (Paint)**,
- **Vector (World Space)**, and
- **Vector (Screen Space)**.

> **Note:** Running the script changes the Painting **Mode** to **Paint Flow Vectors** and switches the paint tool to **Vector Painting,** if this is not already selected.

If you prefer, you can create the shaders and channels manually. To do this:

1. Create the channels you want to use for your vector painting and add the **Vector**, **Tangent To Screen**, or **Tangent To World** adjustments and procedurals in the **Layers** palette.
2. Create your shaders and assign your vector channels to the **Diffuse Color** and **Vector** shader inputs. For example, the default vector shaders assign the **Vector (Screen Space)** channel to the **Diffuse Color** input and the **Vector (paint)** channel to the **Vector** input.
3. Once you've set up the shaders, go back to the **Vector (Paint)** channel and begin painting your normal or flow maps in the **Vector** layer.

Adding Vector Adjustments and Procedurals

There are three adjustments and one procedural in the **Layers** palette that allow you to set up your project for vector painting. The adjustments are **Tangent To Screen**, **Tangent to World**, and **Flow**. These adjustments are where you can paint flow and normal maps within a set channel for your vector painting. The **Flow** procedural acts as an adjustment would, and allows you to use your vector paint as surface flow over time. Input an image, preferably a tileable one, into the **Tile Image** field to apply a texture to your model for viewing flow maps. The **Time Offset** slider represents the flow of time, while the **Speed** slider represents the sensitivity of **Time Offset.**

Adjusting the scale and then dragging on the offset slider, allows you to see the effects your vectors play on the flow map, and you can enable the **Animation** attribute to see the flow over time.

There is also the procedural, **Vector**, that allows you to specify the properties for your vector visualization. **Vector** allows you to specify another color using the sliders in the layer properties.

For more information on what these shaders are and how to use them, see Adding Vector Shaders, and Layer Types and Examples in the Layers section.

Select one of the following default channels created with the setup vector brush script described in the Adding Vector Shaders instructions:

- **Vector (Paint)**
- **Vector (World Space)**
- **Vector (Screen Space)**

Notice that the layers created by the setup vector brush script already show **Vector**, **Vector (Tangent To World Space)**, or **Vector (Tangent To Screen Space)** layers in the **Layers** palette.

If you aren't using the setup vector brush script, and want to add new adjustments or procedurals, follow the steps below.

To add vector adjustments to your project:

1. Select the channel from the **Channels** palette that you are using in the shader component inputs described in Adding Vector Shaders, and open the **Layers** palette.
2. Add an adjustment:
   - From the **Layers** menu. Navigate to **Add Adjustment Layer**.
   - By right-clicking on any layer in the **Layers** palette. Navigate to **Add Adjustment Layer**.
   - From the **Layers** palette. Click the adjustment icon and select either **Tangent To Screen**, **Tangent To World**, or **Flow** from the dropdown list.
3. Adjust the properties in the layer attributes pane of the **Layers** palette.

To add vector procedurals to your project, follow these steps:

1. Select the channel from the **Channels** palette that you are using in the shader component inputs described in Adding Vector Shaders, and open the **Layers** palette.
2. Add a procedural:
   - From the **Layers** menu. Navigate to **Add Procedural Layer**.
   - By right-clicking any layer in the **Layers** palette. Navigate to **Add Procedural Layer**.
   - From the **Layers** palette. Click the procedural icon and select **Basic** > **Vector** from the dropdown list.
3. Adjust the properties in the layer attributes pane of the **Layers** palette.

Note: For a full description of the layer properties for both adjustments and procedurals, see Layer Types and Examples in the section on Layers.

Changing the File Space

Though vectors are often measured in the range -1.0 to +1.0 to represent the vectors in xyz space, Mari operates in the range 0.0 to 1.0. In order to accommodate vector brushing, the -1.0 to +1.0 range is converted into 0.0 to 1.0, and the specific RGB values that indicate vector directions are adapted to reflect this. Thus, -1.0 becomes 0, 0.0 becomes 0.5 and +1.0 stays as it is. The colors are converted to the corresponding number on the 0 to 1 scale to keep all the vector data the same.

In the **Channels** and **Image Manager** palettes, **File Space** fields allow you to set whether the channel or image should be exported as a **Normal** file (in the way Mari stores them), as a **Vector** file (-1.0 to +1.0 range), or as a **Vector (flipped Y)** file, which is the same as the **Vector** file in terms of range, but inverts the vectors in the Y direction. This allows you to easily export inverted vectors for use in other parts of your pipeline. Channel or image files exported in the **Vector** file space format can later be re-imported and changed back to the **Normal** file space format if desired.

Note: For **Vector (flipped Y)** this only inverts the vectors in the Y direction. The vectors in the X or Z direction are not changed.

Changing the file space before exporting can be done on channels, images and paintable layers. This converts the file space range from 0 to 1 (the Mari convention), to -1.0 to +1.0 (Vector convention) or -1.0 to +1.0 with inverted Y
vectors (Vector inverted convention). To change the file space on channels, follow these steps:

1. Open the **Channels** palette and select the channel you want to change.
2. Under **Channel** > **Format** | **File Space**, click the dropdown menu and select either **Normal**, **Vector**, or **Vector (flipped Y)**.

   The file space is changed in all paintable layers in the channel except for shared layers.

To change the file space on images, follow these steps:

1. Open the **Image Manager** palette and select the image you want to change.
2. Under **Image Info** > **MriImage** > **Info** | **File Space**, click the field and select either **Normal**, **Vector**, or **Vector Y Flip** from the dropdown menu.

Creating Flow Maps on Multiple Objects

It may often be necessary to create flow maps for different objects in a project, effectively separating the vector data into more than one channel. Following the steps below prevents the vector markers from appearing on multiple objects when attempting to focus on creating flow maps on only a single object.

1. Set up a basic channel on objects that do not need flow maps, in the **Channels** palette, by either:
   - right-clicking on the palette and selecting **Add Channel** from the dropdown menu, or
   - clicking on the add channel button.

   Mari displays an **Add Channel** dialog box.

> **Note:** Ensure that you have the correct object(s) selected before you begin, to avoid creating any unnecessary channels on the wrong object.

2. Create a channel, called "gray" or similar, and set the **Color** to mid-gray (RGB: 0.5, 0.5, 0.5). Create the channel for only the selected object, for this example, called Object A.

   Using the gray **Quick Channel** can simplify the process, but if you have multiple quick channels in the **Channels** palette, make sure you rename it to avoid any confusion when setting up the shader in the instructions below.

> **Tip:** Create the channel or quick channel with a 256x256 **Size** to avoid taking up unnecessary space.

3. Set up vector default shaders by navigating to **Python** > **Examples** > **Setup Vector Brush** or by clicking the icon in the **Vector Painting** toolbar.

   The script creates two default shaders, and associated channels and layers in each of the **Shaders**, **Channels**, and **Layers** palettes. These allow you to quickly set up the shaders for vector painting and flow maps.
4. With Object A still selected, in the **Shaders** palette, select the **Vector (World Space)** shader that was created by the vector default shader script.

The **Diffuse Color** shader component should already be filled in.

5. Click on the dropdown menu for the **Vector** shader component and select the "gray" channel you created in the **Channels** palette.

   The **Vector** shader component, when set up this way, blocks out the vector inspector markers on objects you don't want to create a flow map for.

6. Lock Object A by clicking on the padlock icon in the **Objects** palette. Due to the fact that vector paint can still be applied to the object - even if vector inspector directional markers aren't shown - this ensures you don't paint over your textures.
7. Select a second object for which you want to create a flow map. For this example, the second object is called Object B. Create default vector shaders, channels, and layers using the script in step three above.

   The **Vector (Paint)** channel is already set to the **Vector** shader component in the **Vector (World Space)** and **Vector (Screen Space)** shaders. This allows you to see the flow map channel where the vector markers are present, without affecting other objects.
8. If you don't already have them activated, turn on the vector painting directional markers by clicking on the icon in the **Vector Painting** toolbar before painting your flow maps.

If you paint on the model now, paint is applied to all unlocked objects. However, only objects with the **Vector (Paint)** channel listed in the shader's **Vector** shader component field show the vector inspector markers.

Note: For more information on shaders and shader components, please see Shaders.

Painting Animated Objects

As well as static model files, Mari can import models that include animation data. This allows you to create a project based on an animated object, so you can check how your textures work as the model moves.

This support for animation lets you:

- Paint models in context - you can check how the textures look through the full range of motion.
- Paint specific textures tied into frame ranges. Imagine that in frame 37 of a sequence, a bullet ricochets off your giant robot. Mari lets you open the model, move to frame 37, and edit the relevant texture layers to show the chipped paint and the displacement from the impact. You can adjust and export texture layers frame by frame.

Keyframes

Keyframes define when the texture changes within a sequence. When you create a keyframe, Mari remembers what textures you've got on your model. Then:

- Any textures you add earlier in the sequence disappear when Mari reaches the keyframe.
- Any textures you add after the keyframe appear once Mari reaches the keyframe.
- When you export your textures, you can export a separate version of each patch for each keyframe.

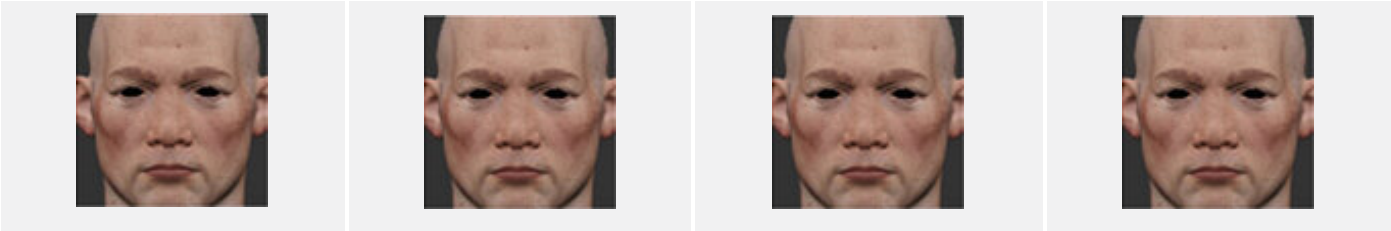
Keyframes divide the sequence into parts. You can paint in any frame within a part, and that paint appears from the previous keyframe until the next keyframe.

The first frame of animation is always treated as a keyframe.

Example

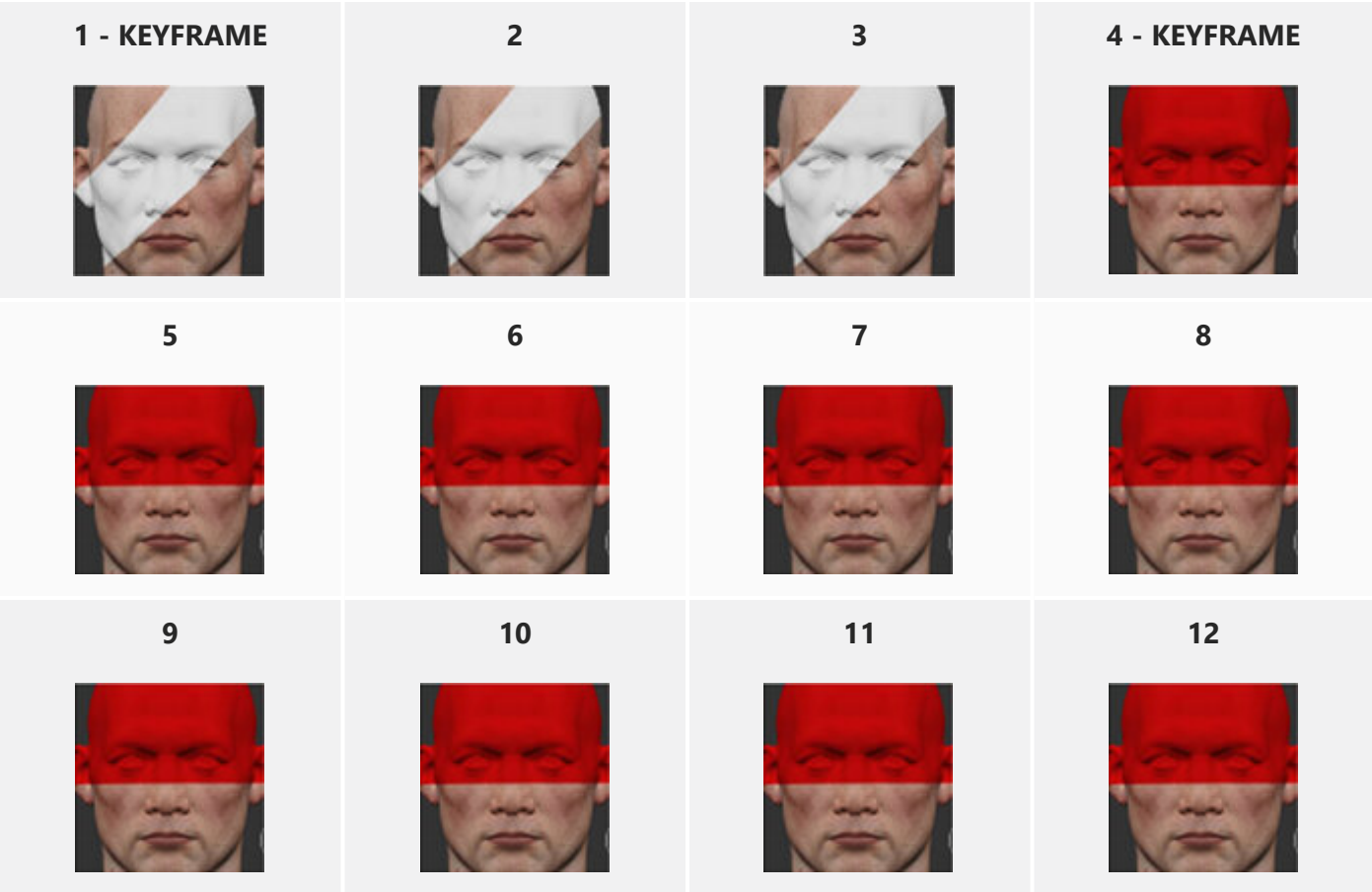
For example, we're painting an object with a 12-frame animation. We have some existing paint. Here, the existing paint shows throughout the entire animation sequence:

1 - KEYFRAME	2	3	4
5	**6**	**7**	**8**
9	**10**	**11**	**12**



Now we set a keyframe at frame 4, and paint on frame 3 and frame 6. In this case:

- The paint on frame 3 appears from frames 1-3.
- The paint on frame 6 appears from frames 4-12.



If we add another keyframe at frame 9 and paint on frame 10:

- The existing paint between frames 4 and 12 continues to display in that frame range, as it was in place before we added the new keyframe.
- The paint on frame 10 appears from frame 9 to 12:

1 - KEYFRAME	2	3	4 - KEYFRAME
5	**6**	**7**	**8**
9 - KEYFRAME	**10**	**11**	**12**

When we export the layer, each patch has three versions - one per keyframe.

1 - KEYFRAME	4 - KEYFRAME	9 - KEYFRAME

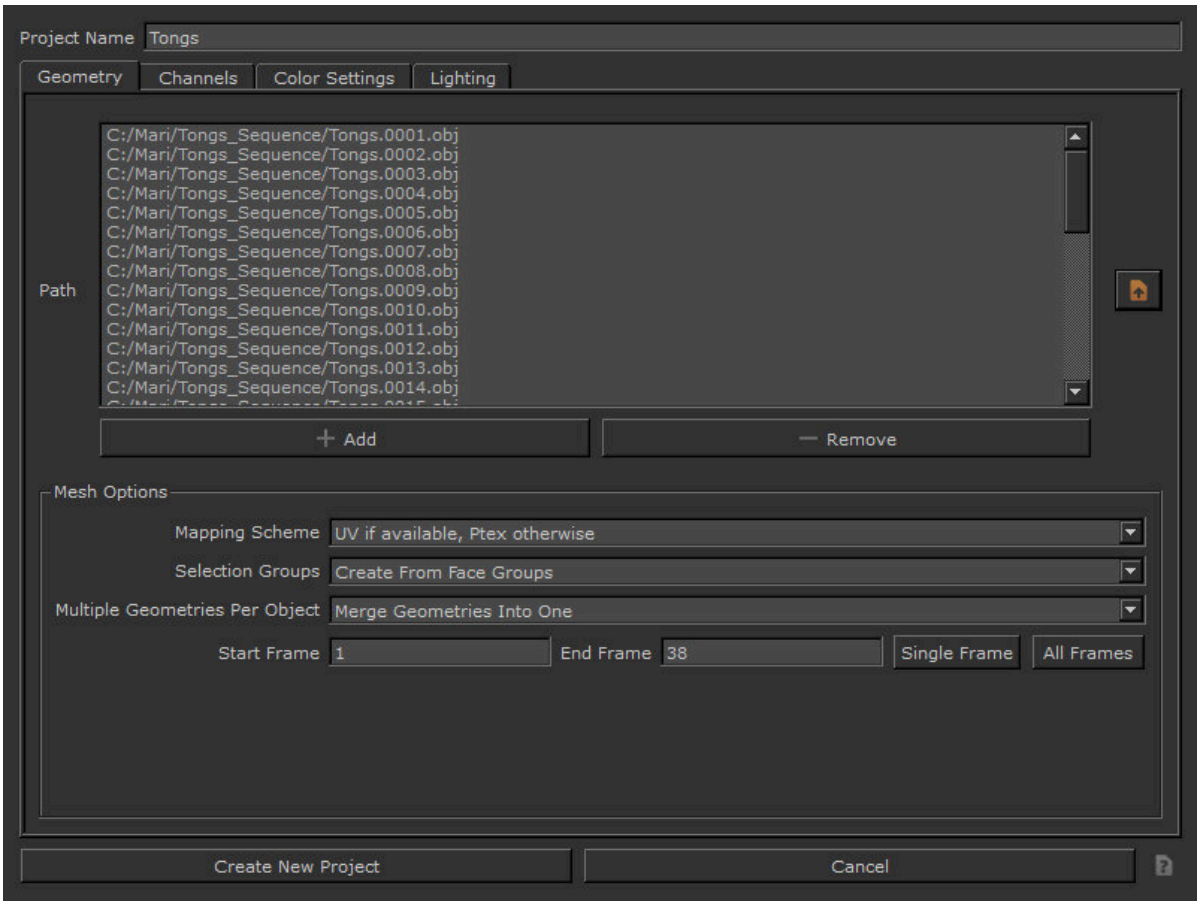
Loading Animated Objects

To load your animated object:

1. Select an animated **.obj** file sequence, and either:
   - create a new project with the file, or

- add the object to an existing project.

2. On the **Mesh Options** tab, a new section displays, letting you set which frames to import.



In addition to the three mesh options offered when opening a new project, you can select:

- **Start Frame** - the first frame of the range that you want to import.
- **End Frame** - the last frame of the range that you want to import.
- **Single Frame** - set the controls so as to only import the first frame in the animation.
- **All Frames** - set the controls to import all the frames within the animation.

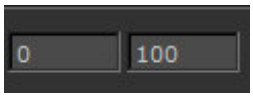
3. Finish creating the project or adding the new object as normal.

Playing Animation

To play back animation:

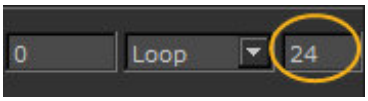
1. If it's not displayed, open the **Play Controls Toolbar** by right-clicking on any toolbars and selecting **Play Controls**. See the *Play Controls Toolbar* section toolbar in the *Mari Reference Guide*.

2. Set the desired frame range. This plays only a subset of the animation.



3. To skip to a particular frame, type the frame in the frame box.

4. Set the playback mode in the dropdown. This can be:
   - **Loop** - keeps looping through the animation (start to finish).
   - **Bounce** - keeps looping back and forth through the animation (start to finish, then backwards back to the start).
   - **Stop** - plays the animation once and stop.
5. Set the frame rate.



6. Use the play controls to play through the animation:

Button	Effect	Shortcut key
	Moves back one frame.	**Ctrl/Cmd+Left**
	Moves back to the first frame.	**Ctrl/Cmd+Down**
No button	Moves to the last frame.	**Ctrl/Cmd+Up**
	Plays forward.	None.
	Stops playing.	None.
	Moves forward one frame	**Ctrl/Cmd+Right**

You can also pull the timeline control back and forth to move through frames:

Painting Animated Textures

Before you paint the animated textures, make sure you've painted all the non-animated elements of your textures. Any paint you add after you've inserted a keyframe only appears in one part of the animation (see the example above).

Once you've got your initial texture set, you can set keyframes.

Setting Keyframes

To set a keyframe:

1. Move to the frame that you want to set as your keyframe.
2. Click or press **Ctrl**/**Cmd**+**K** to set a keyframe.

   The new keyframe displays as a square in the keyframe timeline.

3. You can remove keyframes by selecting the keyframe in the timeline and clicking .

   Removing a keyframe also removes any paint for that keyframe. Be careful when removing keyframes if you've done any painting!

Painting the Textures

To paint a texture within a section of animation:

1. Move to a frame somewhere within the section of animation you want to paint.

   It doesn't matter which frame within the section you move to - the paint displays until the next keyframe.
2. Paint your textures as normal.

   Bear in mind that:

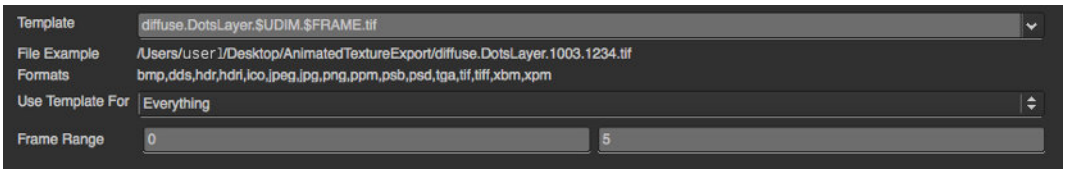
- Any textures you paint within one section disappear when you reach the next keyframe.
- You can paint, then add another keyframe, to "build up" an animated texture (see the Example in Painting Animated Objects for an explanation).

Exporting Animated Textures

When you export an animated texture, Mari exports a separate file for each patch, for each keyframe. For example, a model with 20 patches painted as an animation with 3 keyframes gives 60 files when you export it.

To export an animated texture:

1. Open the **Export** or **Export Flattened** dialog box, from the **Channels** or **Layers** menu.
2. Set the export details as described in the Exporting and Importing Textures section, except:
   - Include the $FRAME keyword in your **Template** - Mari replaces this with the frame number of the keyframe for the exported textures, padded out to five digits. For example, if you are exporting a layer called **color**, with keyframes at 9 and 6, then using an export **Template** of $FRAME_$LAYER.$UDIM.tif gives the following files for patch 0001: 00001_color.0001.tif, 00009_color.0001.tif, and 00016_color.00001.tif.
   - If you want to export only the textures for a particular frame range (which can include multiple keyframes), enter the range in the **Frame Range** field at the bottom of the **Export Layer** dialog box.



3. Click **Export All Patches** as usual to start the export.

   Mari exports the files, creating a file for each patch/keyframe combination.

Configuring the User Interface

Mari provides a fully customizable GUI. You can set it up to suit your working style and preferences.

The main GUI elements that you can customize are:

- **palettes** - areas that contain controls and information, for example to view and configure tool properties, colors, or project elements. You can select which palettes to display and where to put them.
- **toolbars** - panels that contain Mari tools and common functions, for example to paint, erase, open or close a project. Toolbars contain icons you click to quickly access common tools and functions. You can create new toolbars, with shortcut buttons to your favorite functions.
- **HUD** - the heads-up display (HUD) displays information about the current tool and other Mari and project information at the top of the canvas.
- **status bar** - the status bar displays information about current processes, as well as progress bars, error warnings, and basic tool help for the current tool.

Palettes and toolbars can be either *docked* (fixed in one place) or *undocked* ("floating" above the canvas). Palettes can be docked to:

- any side (including top or bottom) of the Mari window, or
- each other (in a separate window).

Docked palettes can also be *stacked* - in the space of a single palette, with tabs to select which palette appears on top. Similarly, toolbars can be docked on the top, left, or right of the Mari window (but not the bottom). By default, toolbars appear at the top, but can be moved. Once you have your workspace laid out properly, you can save the
layout for later use. If you decide that you don't like your custom layout, you can revert to the default layout.

Video: Watch the Customizing the Interface video to learn how to customize Mari's interface. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Choosing and Organizing Palettes

To select and position palettes:

1. Select which palettes to display:
   - from the **View** menu, select **Palettes**, and then a palette from the sub-menu, or
   - right-click an empty spot on the toolbar area, and select a palette from the dropdown menu.

   The dropdown menu lists palettes at the top, and toolbars below.

   OR

   From the **Palettes** toolbar, click the palettes' icon and/or name.

Tip: When you open a palette, it displays where it was last closed. If you open it using the **Palettes** toolbar, it differs depending on which mode you opened the palette. See Configuring Your Workspace for more information.

Video: You can also toggle between hiding and showing palettes, watch a quick demo.

2. You can do the following to position a palette:
   - **move** - click in the title bar and drag.
   - **undock** - click in the title bar and drag to the middle of the canvas, or outside the main Mari window.
   - **dock** - drag to an edge of the canvas (left, right, top, or bottom).

     The side of the canvas expands to "grab" the palette
   - **dock beside another palette** - drag onto the side of another undocked palette, between docked palettes, or between a docked palette and an edge of the canvas.

     Palette(s) "shuffle over" to make room for the new palette, if there is enough room, otherwise they stack (see below).
   - **dock on top (stack)** - drag on top of a docked palette. Palette(s) "shuffle up" slightly, to display tabs below for selecting which palette displays (on top).

Tip: If you dock two floating palettes together, they appear as a separate palette window. You can also use the standard controls at the top right of the palette to minimize, maximize, switch between floating and docked, and close the palette.

Mari also has a show/hide and expand palette function: press the **Space** bar while hovering over the canvas to toggle between showing and hiding the currently open palettes. You can also press the **Space** bar while hovering over a palette to expand it. You can't expand a palette if you selected an item in it.

Tip: Palette behavior is customizable. Experiment with the palettes to see how they can be docked, undocked, moved, and stacked.

Choosing and Organizing Toolbars

To select and position toolbars:

1. To select which default toolbars to display, right-click an empty spot on the toolbar area, and select from the dropdown menu (as for opening palettes, above).

The default toolbars are:

- **Ptex** - Ptex settings (changing face resolution, texel density, and so on).
- **Channels** - buttons to switch between the four default views for painting (Channel, Stack, Layer, and Paint).
- **Navigation** - buttons to set your navigation preferences while moving around the canvas.
- **Tool Properties** - lets you set the properties for the current tool.
- **Project** - common project functions (New project, Open, Save, and so on).
- **Lighting** - buttons to switch between the three lighting modes and to toggle shadows on or off.
- **Painting** - buttons to clear the paint buffer.
- **Tools** - all the painting tools.
- **Selection** - buttons to switch between three selection modes (objects, patches, and faces) when using the Select tool.
- **Canvas** - view options for the UV and 3D views.
- **Mirroring** - buttons to switch between four paint mirroring settings.
- **Vector Painting** - enable the vector grid and quickly set up vector shaders.
- **Colorspace** - lets you specify your colorspace and the settings associated with your choice.

2. To position a toolbar, click the drag control at its left or top edge, and:

- **move** - drag to another position.
- **undock** - drag to the middle of the canvas, or outside the main Mari window.
- **dock** - drag to an edge of the canvas (left, right, or top - not bottom); the side of the canvas expands to "grab" the toolbar.
- **resize** - drag left or right, or up or down.

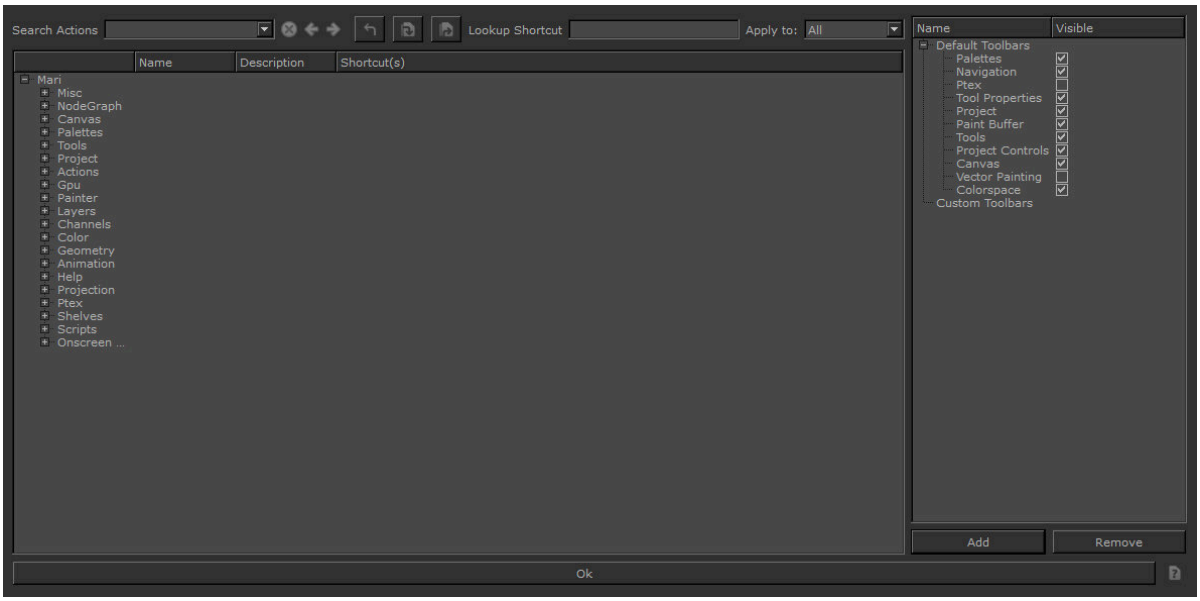
Tip: If the toolbar is too small to fit all of its items, a double chevron appears at the end (»). Click this to see all the tools in the bar, or resize and position the toolbar to see all items.

Creating Custom Toolbars

If you want quick access to functions not included on the default toolbars, you can create as many additional toolbars as you need. To create a new toolbar:

1. From the **Edit** menu, select **Toolbars**.

   The **Manage Toolbars** dialog box opens.

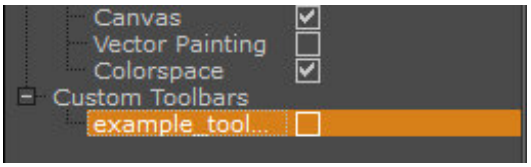


2. Click **Add** to create a new toolbar.

   Mari prompts you to name the toolbar.

3. Give your toolbar a name, and click **OK**.

   The new toolbar displays under **Custom Toolbars** in the **Manage Toolbars** dialog box, and an empty toolbar displays for it on the main screen.



Tip: If the empty toolbar does not display, make sure that the **Visible** checkbox in the **Manage Toolbars** dialog box is ticked. Also note that the new toolbar may display at far right, and you may have to drag its handle to the left to resize it, to make room for new icons.

4. Add functions to your toolbar by dragging from the list on the left in the **Manage Toolbars** dialog box, onto the empty toolbar on the main screen.

Tip: Click **+** to expand the nodes in the list to see the functions available. At the moment, items cannot be removed from custom toolbars.

5. Click **OK** to close the **Manage Toolbars** dialog box when you're happy with your new toolbar.

Note: To remove items from your custom toolbar, click on the dropdown arrow next to the item and select **Remove**.
Your project needs to be open in order to remove items.

Try this to acquaint yourself with managing toolbars:

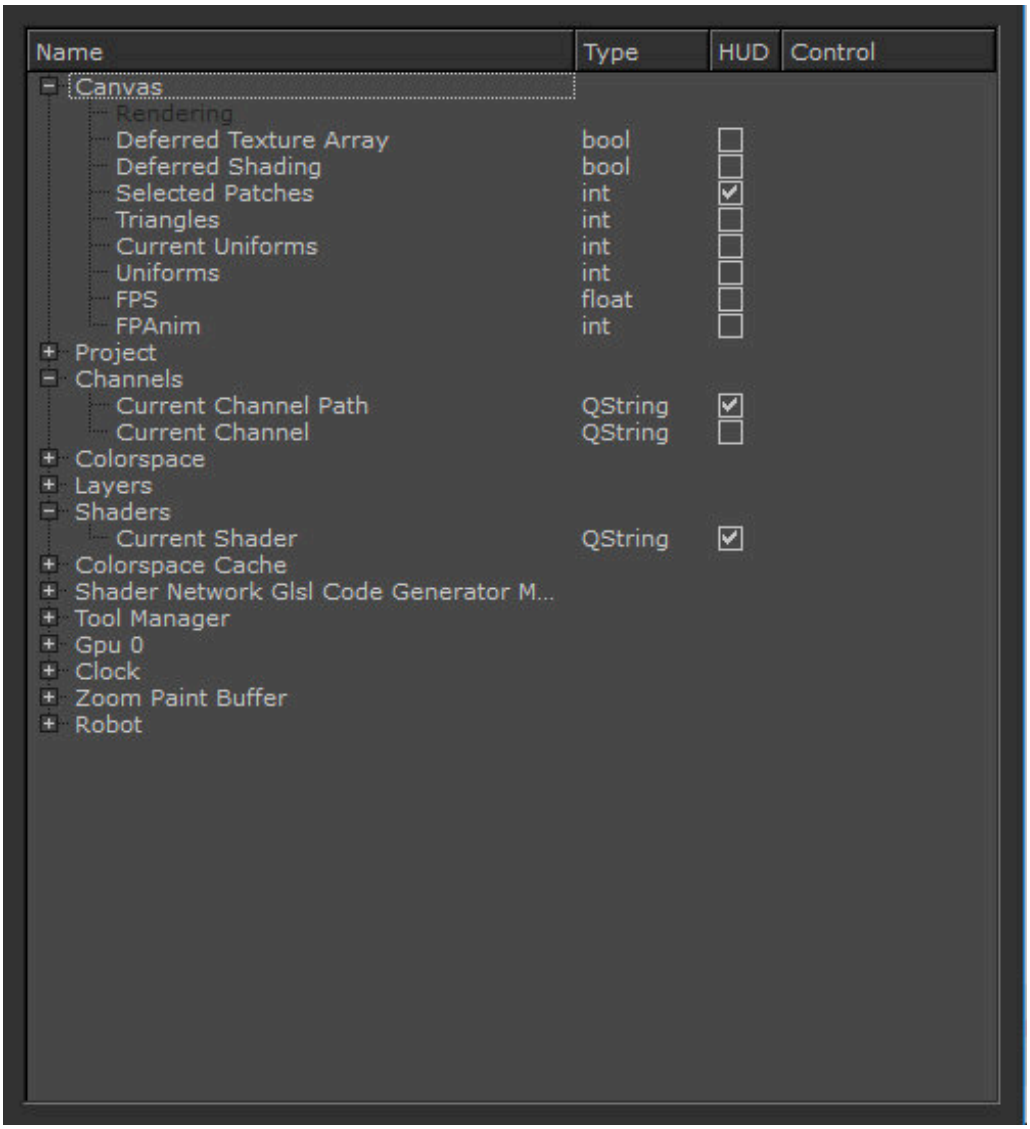
1. create a new toolbar,
2. add a few items to it,
3. resize it,
4. undock it, move it, and re-dock it somewhere else,
5. hide it from the display, then
6. remove it.

Configuring the HUD

To configure the head-up display (HUD):

1. From the **Edit** menu, select **HUD Manager**.

   The **HUD Manager** dialog box opens. The information you're most likely to want to display is under **Tool Manager** and the current project (in the illustration, **Example Project**).



Tip: Many of the options in the **HUD Manager** are for debugging purposes only. For an explanation of the different options, see the *Mari Reference Guide*.

2. Click the checkboxes for the options you want to include in the heads-up display at the top of the canvas, then close the **HUD Manager** dialog box.

   Mari displays information in the HUD as you select it in the **HUD Manager**.

Saving Your Layout

A layout is a record of how your Mari workspace is set up. Once you have your workspace set up, you can save the layout so you can easily recall it later. Alternatively, you might want different layouts for painting different types of objects.

1. From the **View** menu, select **Save Layout**.

   A standard **Save** dialog box opens.
2. Enter a filename and select a location for your saved layout.
3. To reload the saved layout at any time: from the **View** menu, select **Open Layout**.

Tip: To revert to the default layout: from the **View** menu, select **Default Layout**.

Tip: Try moving some palettes and toolbars around, then saving the layout. Revert to the default layout, then re-load your saved layout.

Understanding the Status Bar

The status bar is a fixed panel of icons that appears in the bottom-right corner of the Mari window. It displays information about the project as icons, including progress bars for current processes, error warnings, features that can be toggled, and basic tool help for the current tool.

All icons display tooltips when the cursor hovers over the icon. Some can be clicked to toggle the displayed functionality. For more information, see the Status Bar.

Status Bar Icons

Icon	Description	Notes
Udim:	Displays the UDIM number of the patch you have selected. If nothing is selected, then there is nothing listed after **Udim**.	
	Notifies you that there is unbaked paint in the paint buffer. Clicking on this icon bakes the current painting.	This icon only displays in the status bar if there is paint in the paint buffer.
	Notifies you that your painting is hidden.	
	Notifies you that background jobs are currently running. Clicking on this icon displays a dialog for the progress bar.	
	Notifies you that the project has changed and allows you to click on the icon to save changes.	
	Notifies you that you are currently projecting on the front of objects. Clicking on this icon toggles to Project Through Objects. Clicking on this icon toggles between projecting through objects and projecting on the front of objects. The icon takes the default icon's place when switching to project through. More information for projecting on or through objects can be found in the Projectors section.	
	Notifies you that you currently have edge masking enabled. More information on edge masking and the edge masking icon can be found under Edge Masks.	This is not a unique icon. It also displays to indicate that fractal noise masking is enabled. If both masks are being used, the icons appear separately on the status bar.
	Notifies you that you currently have channel masking enabled. More information on channel masking and this icon can be found under Channel Masks.	

Icon	Description	Notes
	Notifies you that you currently have ambient occlusion masking enabled. More information on ambient occlusion masking and this icon can be found under Ambient Occlusion Masks.	
	Notifies you that you currently have depth masking enabled. More information on depth masking and this icon can be found under Depth Masks.	
	Notifies you that you currently have backface masking enabled. More information on backface masking and this icon can be found under Backface Masks.	
	Notifies you that you currently have fractal noise masking enabled. More information on fractal noise masking and this icon can be found under Fractal Noise Masks.	This is not a unique icon. It also displays to indicate that edge masking is enabled.
	Notifies you that you currently have the global mask preview enabled. Clicking this icon disables the global mask preview. More information on the mask preview and this icon can be found under Masking What You Can Paint On.	
	Notifies you whether you are in LDR (Low Dynamic Range) or HDR (High Dynamic Range) mode.	Clicking on this icon toggles between LDR and HDR. When in HDR mode, the icon appears as .
	Warns you that a low memory status has been detected. Low memory affects Mari's performance and may make it unstable. Try freeing memory or closing and restarting Mari to clear this issue.	This is not a unique icon. It also displays to warn you that the cache disk is critically full.
	Notifies you that there are warning messages present.	Clicking on this icon displays these warnings.

Icon	Description	Notes
	Warns you that the cache disk is critically full. If you don't free space, data loss may occur. To free space, try: • closing and reopening the project. • freeing up space on the disk. • placing your project files on a file system with more free space.	This is not a unique icon. It also displays to warn you that a low memory status has been detected.
	Hovering the cursor over this icon displays the tooltip **No space left for this operation.** There are a few reasons that this might display, including: • The virtual texture size is set too low in the **Preferences.** • There are too many channels in the **Channels** palette. • The patch resolution is set too high on some patches.	This is not a unique icon. It also displays to indicate a higher bit-depth than what is configured. Attempting to click on the icon for the Overload warning displays nothing further.
	Hovering the cursor over this icon displays the tooltip **Channels are higher bit-depth than what the virtual texture is configured for.** Clicking on the icon increases the bit-depth of the virtual texture to match that of the highest bit-depth and removes the status bar warning.	This is not a unique icon. It also displays to indicate virtual texture overload. Hovering over the icon for a tooltip indicates which warning it is.
	Notifies you that Mari is currently loading data from the disk.	
	Notifies you that Mari is currently loading data from the RAM to GPU memory.	
	Notifies you that Mari is listening for commands on [COMPUTER NAME] port [PORT NUMBER].	Clicking on this icon shows you what computer name and port number Mari is using to listen for commands.
	Notifies you that there was an error while Mari was listening for commands.	

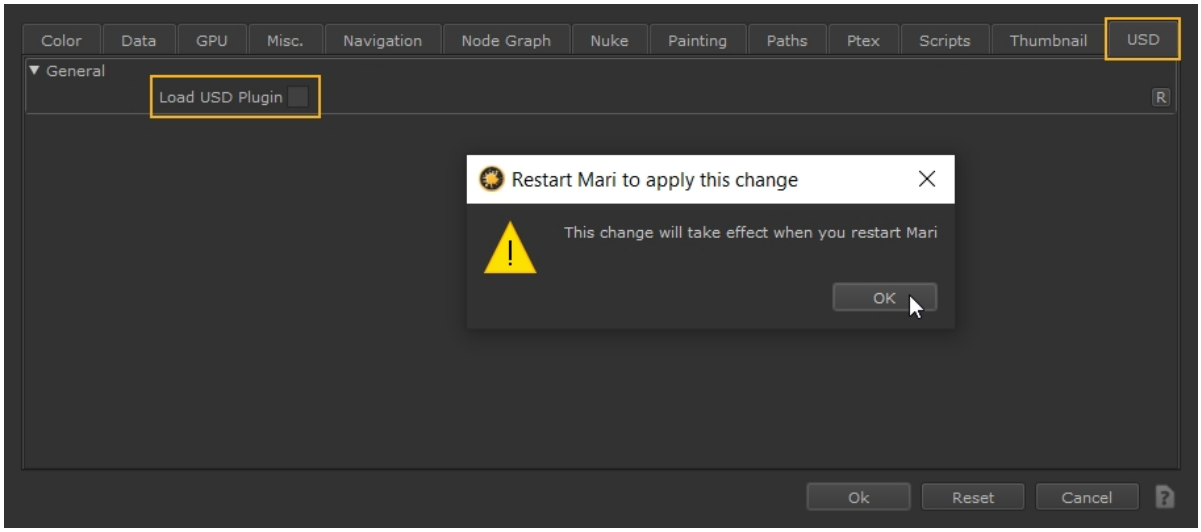
Configuring USD Plug-ins

As part of Mari's support for the USD format, it ships with a native USD plug-in platform. This plug-in is enabled and loaded by default.

Note: The native USD platform plug-in for Mari operates on USD version 25.08.

If you want to use your own plug-in instead of the native support platform, you can unload native support in the **Preferences**:

1. Under the **Edit** menu, click **Preferences**.
2. Navigate to the **USD** tab and uncheck the **Load USD Plugin** box.
3. Mari must be restarted to disable or enable the native USD plug-ins, so click **OK** and restart the application to apply the change.



Items and Objects

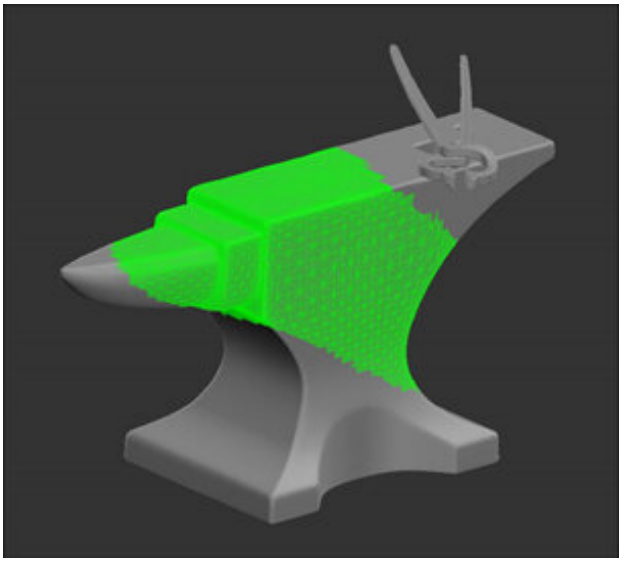
Discover Mari's selection tools, how to work with patches and objects, and working with Ptex and Alembic.

Selecting Items

Mari has two selection tools:

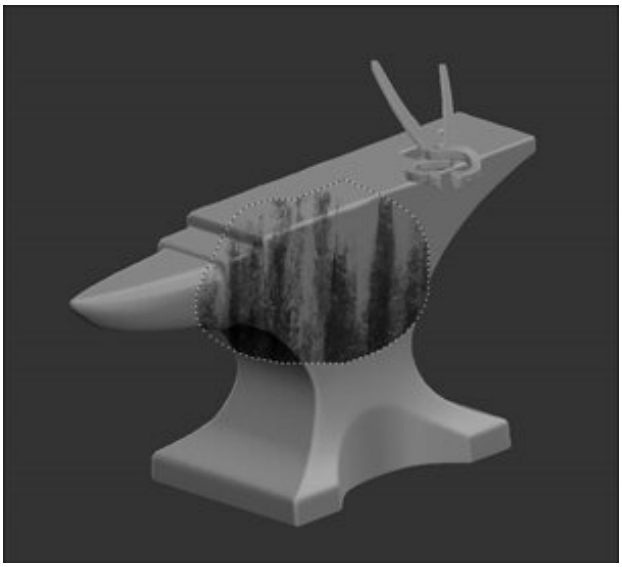
- **Select** - this lets you select areas on the model(s) in your project. You can select whole objects, patches, or areas on the surface of the objects.
- **Marquee Select** - this lets you select areas on the screen.

The **Select** tool lets you select areas on the objects in your project. Once you have made a selection, you can choose to show or hide the selected areas, or to lock them for editing.



Area selected on the model

The **Marquee Select** tool lets you select an area onscreen, to control where you can apply paint. When you've made a selection with the **Marquee Select** tool, paint only bakes down within the selected areas.



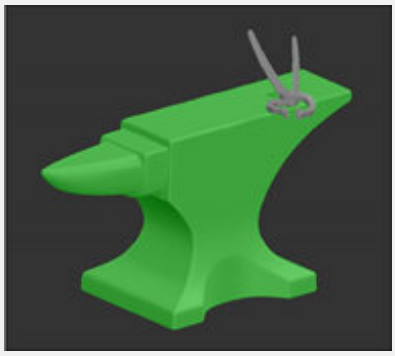
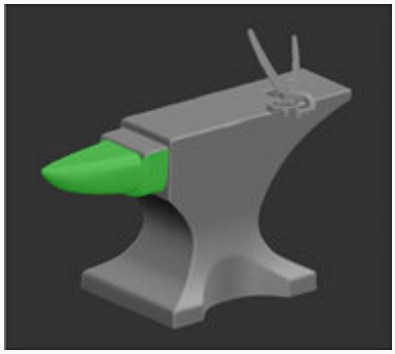
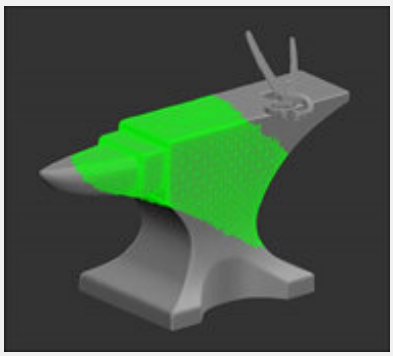
Marquee Select - paint only shows within the selection area.

Tip: This works like the **Select** tool when **ProjectOn** is set to **Selected**, except that you're selecting areas on the screen rather than on your model.
You can't use the **MarqueeSelect** tool to select areas to show/hide or to lock.

Video: To learn how to select geometry and control the visibility of it, watch the Paint Buffer Basics video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Selecting Objects, Patches, or Faces

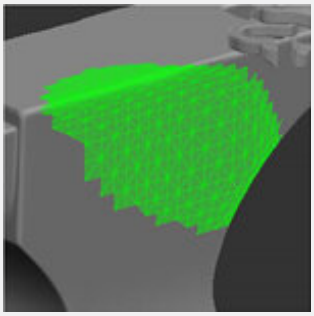
Mari has three selection modes. These control how you select things in your project, from whole objects to sections of surface.

Mode	Description	Example
Object	Lets you select whole objects at a time. Clicking on an object selects the whole of that object - every patch, every point on it.	
Patch	Lets you select individual patches on an object. Clicking on a patch selects only that patch (and all points within the patch).	
Face	Lets you select object faces. You can use standard marquee and loop select tools to select parts of your objects.	

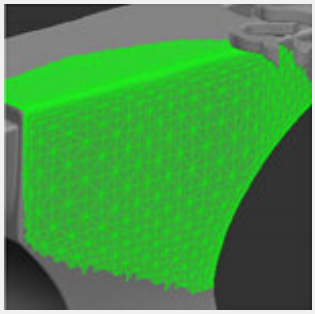
Video: Watch a quick demo on selection modes.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

In all three modes, you can add to, or remove from, the current selection. You also have the option of selecting all items or none, or invert the current selection.

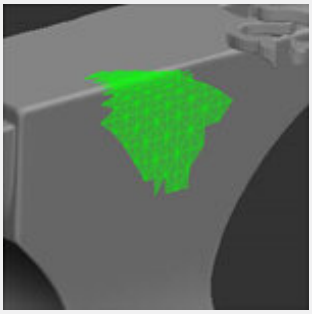
In **Face** selection mode, you can grow or shrink the current selection.



Initial selection



Selection grown



Selection shrunk

The smart selection mode of the **Select** tool (when in **Face** selection mode) lets you create selections based on the surface of the model. Similar to how Photoshop's magic wand tool lets you select areas of similar color, this lets you select areas that face in the same direction.

You can base your smart selection on:

- Connectedness Mesh
- Connectedness UV
- Edge Angles
- Orientation
- Selection Group

Tip: You can also make a smart selection by double-clicking a geometry while any selection tool shape is selected (square, lasso, or polygonal lasso). This way you can combine standard selection and smart selection together.

Video: Watch a quick demo on how to use the **Face** selection mode.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

You can use the selection to control where paint can bake onto the surface of your model. If **Project On** (on the **Painting** palette) is set to **Selected**, then Mari only bakes paint onto selected areas.

To use the **Select** tool:

1. Start Mari and open your project.
2. From the **Tools** toolbar, choose the **Select** tool .
3. You can use the **Selection** tool on the **Project Controls** toolbar to switch selection modes.

If you can't see the **Project Controls** toolbar, right-click on the toolbar area and choose **Project Controls** from the dropdown menu.

4. Set the selection mode.

In this mode	You can...
Object	click on an object to select it.
Patch	click on a patch to select it, or click and drag a selection to select all the patches under that selection.
Face	click and drag to make the selection.

5. On the **Tool Properties** toolbar, set the selection details to **Facing** - selecting the (visible) front of the model, or to select "through" the model, including the back side - or the selection tool shape.

   The tool shape can be either:

   - a square selection box,
   - a lasso selection (allowing you to draw any shape of selection),
   - a polygonal lasso selection (allowing you to draw any shape in the form of a polygon), or
   - smart selection (letting you select areas based on the shape of the model).

Note: Smart selection only works in **Face** selection mode.

6. Click and drag to make your selection.
7. You can use modifier keys to add to, remove from, or toggle the selection:

Hold down...	To...	Effect
Shift	**Add** to the selection	Areas that were unselected become selected, areas that were already selected stay selected.
Ctrl/Cmd	**Remove** from the selection	Areas that were selected become unselected, areas that were already unselected stay unselected.

Hold down...	To...	Effect
Ctrl+**Shift**+click	**Toggle** the selection	At the cursor's location, areas that were selected become unselected and areas that were unselected become selected. **Note:** This only works with **Patch** and **Face** selection modes.

8. You can select all, none, or invert the current selection. Use the buttons on the **Tool Properties** toolbar, the options in the **Selection** menu, or the right-click menu on the canvas.
9. In the **Face** selection mode, use the **Grow** and **Shrink** buttons to change the size of the current selection.

   You can use the Grow/Shrink **By** field to set how these work. **Edges** grows along the edge of the current selection, **Vertices** grows using the vertices as a guide.

Tip: You can only grow the selection up to the edge of the patch(es) that your face selections cover.

The **Smart Selection** mode works slightly differently from the other modes. To use it:

1. In **Face Selection** mode , click .
2. In the tool properties toolbar, set the **Type**. This sets how to create the selection area:
   - **Connectedness Mesh** - selects the areas on the model connected in 3D view.
   - **Connectedness UV**- selects the areas on the model connected to the selected face. When you look at the area in the UV view, this selects the whole connected area within the current UV patch. If the UV patch holds several different areas, only those areas you click on are selected.
   - **Edge Angle** - selects based on the degree of change between neighboring faces. The selection goes until it hits an angle higher than the **Smart Angle**.
   - **Orientation** - selects the area that faces in the same direction as your initial selection.
   - **Selection Group** - selects the whole selection group that the selected face belongs to. If the selected face does not belong to a selection group, nothing happens. If a face belongs to more than one selection group, clicking repeatedly cycles through the other selection groups to which it belongs.
3. Set the **Angle** - this defines the maximum angle for the smart selection.
   - For **Edge Angles**, this sets the maximum angle between areas on the model surface. For example, if the angle is set to 30, this selects areas up to a 30 degree change of facing.

- For **Orientation**, this sets how far from the facing of your original selection the selection goes to. For example, if the angle is set to 30, this selects areas up to 30 degrees away from the facing of your original selection.

4. Click and drag to make the selection.

   Mari creates a selection around the area you drag.

   Alternatively, if you set **Selection Group** for the **Type**, click a face on the canvas repeatedly to cycle through the selection groups to which it belongs. The selections groups are selected in the **Selection Groups** palette and the selection group is highlighted on the model.

Alternatively, you can use the buttons on the **Tool Properties** toolbar to quickly select patches. To do this:

1. In **Patch Selection** mode, click any of the selection modes in the **Tool Properties** toolbar (Rectangle, Lasso, Smart).
2. To quickly select all patches, click the All button on the **Tool Properties** toolbar.

   All patches on your current object are selected.
3. To quickly clear your selection so that none of your patches are highlighted, click on the None button on the **Tool Properties** toolbar.

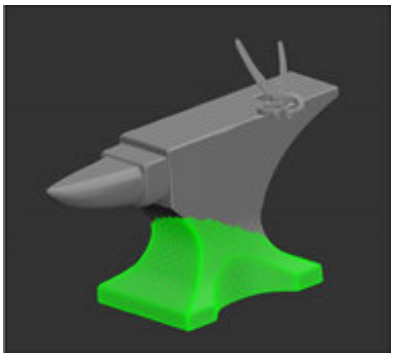
   Any patches you have selected are unselected.

Hiding and Locking

You can show or hide objects, patches, and faces based on whether they are selected, and you can also invert the visibility of objects, patches, and faces. For example, you can select a patch or area on an item, and then hide everything not selected. Similarly, you can lock objects and areas based on whether they are selected or not.

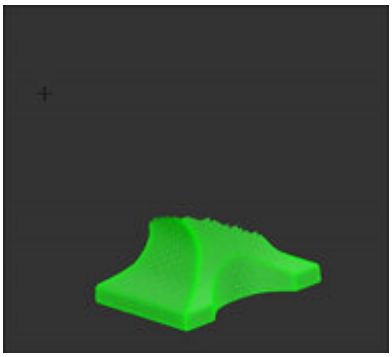
Hiding Objects, Patches, and Faces

1. In the **Object** , **Patch** , or **Face** selection modes, select an area or an object on your model.



2. From the **Selection** menu, select **Hide Unselected**.

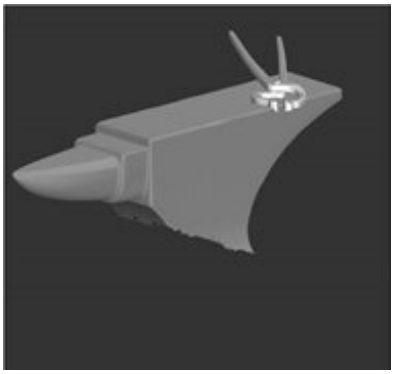
   Mari hides everything except your selection.



   You can clear the selection, and the other areas or objects remain hidden.

3. Conversely, from the **Selection** menu, select **Hide Selected.**

   Mari hides only the selected areas or objects.



4. To show the hidden objects, patches, or faces again, from the **Selection** menu, select **Show Selected**, **Show All**, or **Show Entire Object,** to restore visibility of objects, patches, or faces.
5. You can invert the visibility on all of the objects in the scene based on the current selection mode. From the **Selection** menu, select **Invert Visibility**.

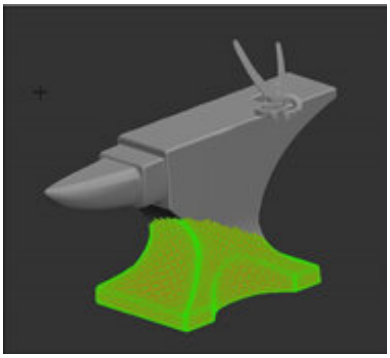
Selection Mode	Invert Visibility Option
Object	Any objects currently visible are hidden. Any hidden objects are made visible.
Patch	For each object in your scene, any patches currently visible are hidden. Any hidden patches are made visible.
Face	For each object in your scene, any faces currently visible are hidden. Any hidden faces are made visible.

Note: The **Invert Visibility** option works per selection mode.
For instance, if you have hidden some, or all the faces or patches in an object, then using the **Invert Visibility** option in a different selection mode on the object, has no effect on the hidden areas of the object. The invisible patches, faces, or objects remain hidden until you make them visible in the mode they were hidden.

Locking Objects, Patches, and Faces

1. In the **Object**, **Patch**, or **Face** selection modes, select an area or an object on your model.
2. To lock the selection so that you can't paint on the locked areas, from the **Selection** menu, select **Lock Selected**.

   Mari locks the selection. Try painting a few strokes over the model, the paint does not “stick” to the locked areas.



3. As with hiding, you can also lock the inverse of your selection by selecting **Lock Unselected** from the **Selection** menu, or if you want to lock the whole model in order to work on something else, you can **Lock All**.
4. To unlock the selection, from the **Selection** menu, select **Unlock Selected**, or unlock the whole geometry by selecting **Unlock All.**

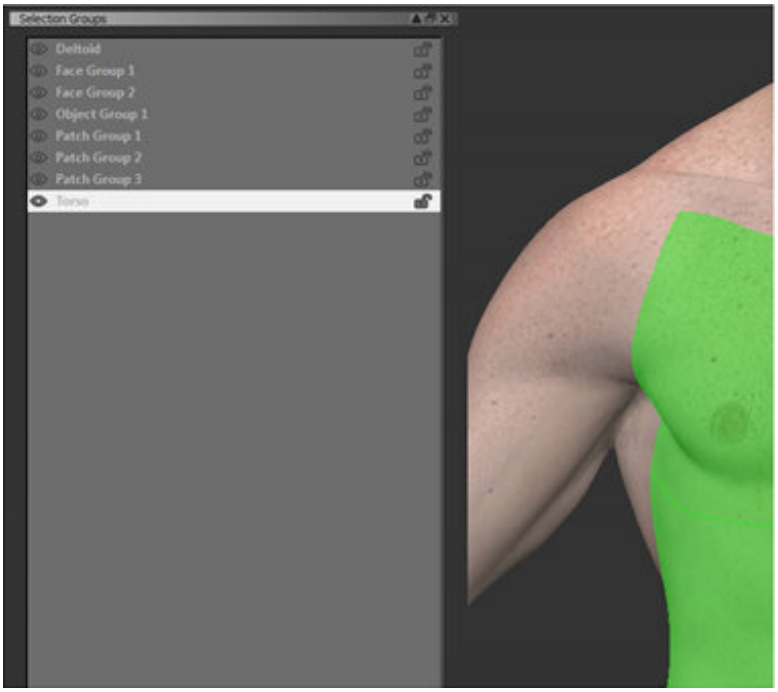
5. If you've locked multiple patches, faces, and even objects and want to quickly unlock everything in your project, select **Unlock Entire Object** from the **Selection** menu.

Tip: You can also show/hide and lock selections from the right-click menu on the canvas.

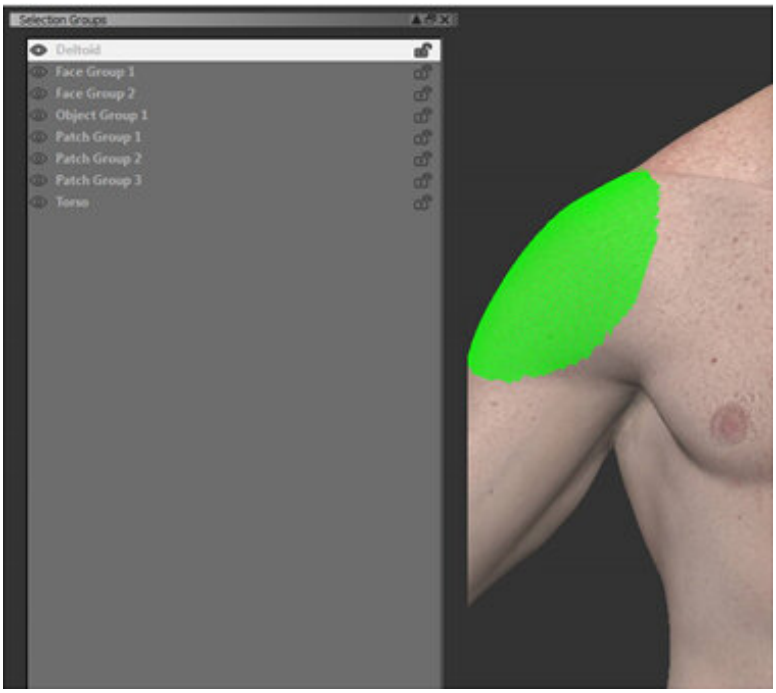
Saving Selections on the Model

Mari lets you save selections and come back to them later. Once you've made a selection (in any mode - objects, patches, or faces), you can create a selection group. Clicking on a selection group selects the items in that group.

Video: Watch a quick demo on saving selections.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.



Example of a selection group containing patches



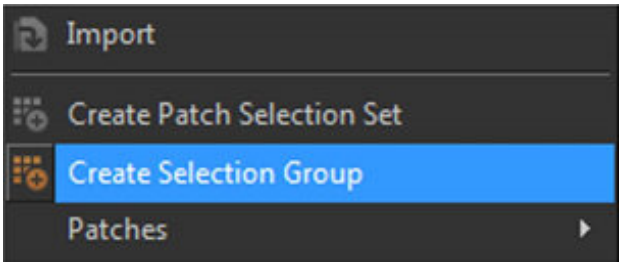
Example of a selection group containing faces

You can use this to set up face selections of specific areas on your model, and then to reuse these selections. This lets you define your own selection areas for relevant features that aren't necessarily reflected in how the patches on the model are put together. For example, you could use this to create a selection for the blacksmith's deltoid muscle, or his collarbones.

1. With the **Select** tool, create your selection.

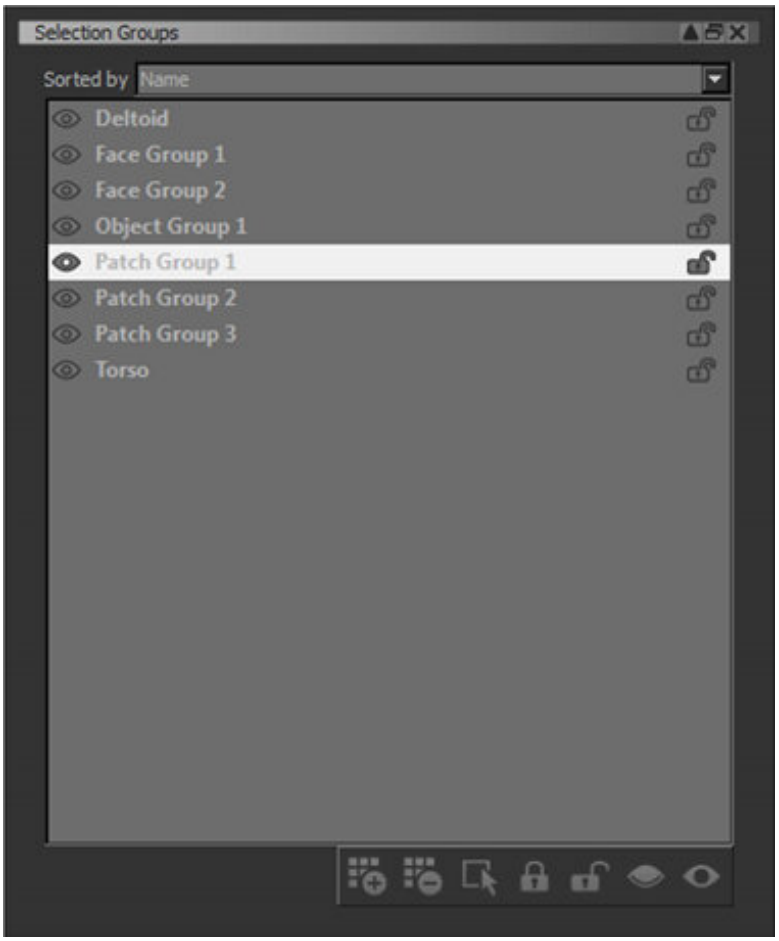
   You can select in any mode - object, patch, or face.

2. Right-click on the selection. From the dropdown menu, pick **Create Selection Group**.



3. Open the **Selection Groups** palette.

   This displays all the selection groups for your project.



4. Click to select a selection group.

   When you click on the selection group, Mari changes to show that selection. You can switch between selection groups by clicking them in the list, or by selecting a face in the canvas, when the **Type** is set to **Selection Group**, and clicking on it repeatedly to cycle through the selection groups to which it belongs.

   Object groups can be added to selection groups. You can create up to 200 of these groups.

> **Note:** If you have **Type** set to **Selection Group**, and have clicked on a face but nothing happened, the face may not belong to any selection groups.

If you have a selection group highlighted in the list, but have used the **Select** tool to make another selection, you can reload the selection group by clicking on the button. You can also lock and unlock selection groups, as well as showing and hiding them.

To...	Click on...
lock the selection group	
unlock the selection group	
hide the selection group	
show the selection group	

Tip: You can also create patch selection sets by right-clicking on your selection and selecting **Create Patch Selection Set**. This can also be done in the **Patches** palette by right-clicking on your selection and selecting **Create Set**.

You can add selections to or subtract selections from a selection group by using modifier keys to adjust what selections are visible on the canvas. This allows you to view which faces or patches are selected, or create a new selection group from existing groups.

1. To add selection groups together, select a selection group from the **Selection Groups** palette then **Shift**+click on another selection group.

   The canvas updates to display the two selection groups as they would appear if they were one group, by adding the first selection in the palette to the second.
2. To subtract a selection group from another, select a selection group from the **Selection Groups** palette then **Ctrl**/**Cmd**+click another selection group.

   The canvas updates to display the first selection group without any of the selections from the second group.

Selecting Areas Onscreen

There are also quick menu items for selecting large areas on your model at once. In the **Selection** menu, you can opt to:

- **Select All** - to select all of an object, patch or face.
- **Select None** - to de-select patches and faces.
- **Select Invert** - to select the inverse of the selected area.

• **Select Visible** - to select the visible area of a patch, face, or object.

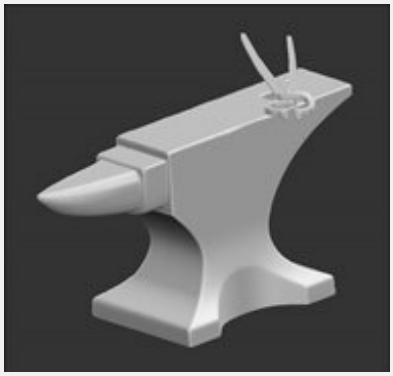
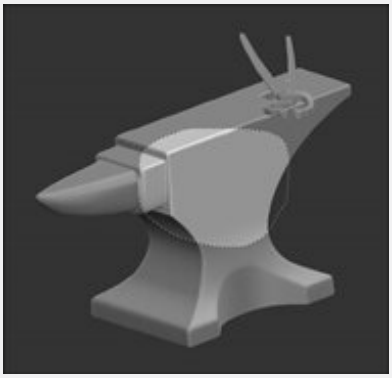
Alternatively, use the **Marquee Select** tool to select an area onscreen, rather than on the model. The **Marquee Select** tool lets you build up your selection (adding and removing to the selection as required), then transform it – moving it around, rotating it, fill it, invert it, or resizing it.

1. Paint a few strokes onto your model, so you can see the effects when you start drawing marquee selections.
2. Select the **Marquee Select** tool .
3. In the **Tool Properties**, select the tool type. You can select a rectangle, lasso, polygonal, ellipse, or wand.

Tool	Lets you
	Draw a rectangular selection.
	Draw the outline of your selected area. You can draw any shape you want.
	Draw the outline of the selected area by setting points for the polygon sides and double-clicking or clicking the original point to close. You can draw any shape you want.
	Draw an elliptical selection.
	Select by color. When you click, Mari creates a selection around the point you clicked. The selection is based on color – areas with a similar color to the original point are selected. You can change the color threshold for the selection.

Tip: For the **Polygonal** mode, you can also close the selection by pressing **Enter** or double-clicking on the canvas.

4. Click and drag to create your initial selection.

	
Initial buffer, with unbaked paint.	Marquee select - only the paint within the selection is visible.

5. To edit the shape of the selection, change the **Mode** in the **Tool Properties**. You can set this to:

Mode	Does	Notes
Replace	the new selection replaces the existing selection.	In **Replace** mode (the default), you can: • hold down **Shift**, click and drag to add to the current selection, or • hold down **Ctrl/Cmd**, click and drag to remove from the current selection.
Transform	lets you move your selection area around (move, rotate, or scale the selection)	In **Transform** mode, you can: • hold down **Shift**, click and drag to move the current selection, • hold down **Ctrl/Cmd**, click and drag to rotate the current selection, or • hold down **Shift**+**Ctrl/Cmd**, click and drag to scale the current selection.

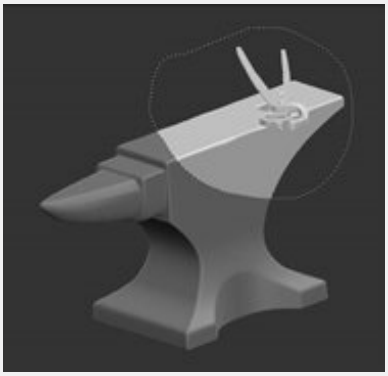
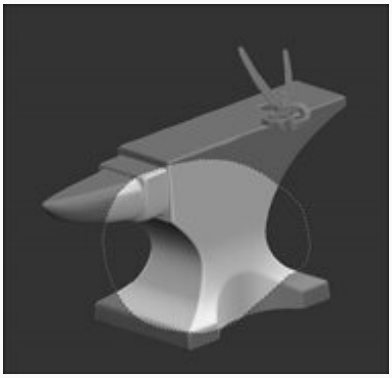
6. You can fill the selected paint buffer area with the current foreground color using the Fill option, or by dragging-and-dropping the foreground or background color swatches.

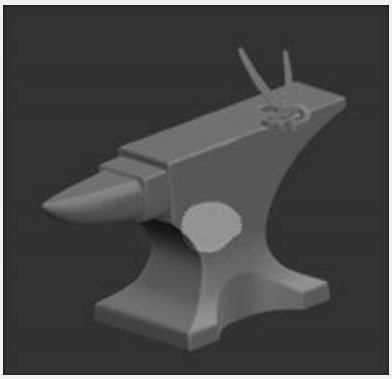
7. If you want to soften the edges of the selection, turn up the Feathering . Higher values give the selection a softer, fuzzier edge. At 0, the selection has a hard edge.

8. In Wand mode, you can change the **Color Threshold** (how close the colors need to be to be selected when you click). At 0, only areas of the exact shade as the pixel you click on are be selected. As the value increases, more
colors are selected when you click.

Moving and Distorting Marquee Selections

To move or distort the selection, set the **Mode** to **Transform**.

To	Do this:	Example
move the selection	hold down **Shift**, click and drag	
rotate the selection	hold down **Ctrl**/**Cmd**, click on your desired point of rotation, and drag	

To	Do this:	Example
scale the selection up or down	old down **Shift**+**Ctrl**/**Cmd**, click and drag	

Working With Patches

Models that you paint in Mari are divided into patches. You can change the properties of the patches, such as the patch resolution. You can also edit patches as a whole - by copying and pasting between patches, filling patches with color, or flipping the paint on them.

Video: Have a quick look at the Working with Patches in Mari video to learn more.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Patch Resolution

Each patch on a channel is stored at a particular resolution. Resolution is a determination of how many pixels are used for an item. For example, a patch may have a resolution of 512 x 512. This patch is stored as an output file 512 pixels wide by 512 pixels high. Higher resolutions take more memory and require more storage, but provide more detail.

Resolutions available in Mari range from 256 x 256 to 32k x 32k (32768 x 32768). Resolutions in Mari are always “square”, with the width equal to the height.

You set the resolution for a channel when you create it. You can however change the resolution later, for example downsizing if you find the file's getting really big and you don't need so much detail, or upsizing if the shot focuses on something that you find needs more detail. You can either:

• change the resolution of the entire channel at once (that is, all patches in the channel), or

• resize specific patches within the channel.

Changing the resolution of a patch in one channel does not affect the resolution of the same patch in another
channel. For example, a particular patch could be at 512 x 512 in the bump channel and 2k x 2k in the diffuse channel. All layers in a channel or patch are resized when the channel or patch is resized, but you cannot resize individual layers.

Tip: For more details on working with channels, see Channels.

Patch Limit

Mari has a technical limit of how many patches it can accurately handle, which is primarily based on the graphics card you are using. For NVIDIA and AMD cards, the limit is 4096 patches, while for Intel the limit is 1024 patches. The patch limit primarily affects the visibility and locking ability of patches, as well as the ability to select through faces. You may be able to work with more patches than the stated limit, if you are not using this functionality but it isn't recommended and correct behavior cannot be guaranteed.

Changing Patches as a Whole

Mari lets you change patches as a whole. You can:

• copy textures from one patch to another

• fill patches with a single color

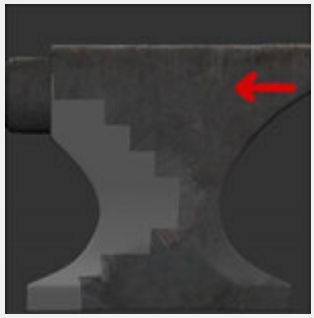
• rotate or flip the paint on a patch

• mirror paint from one side of a patch to the other

• link patches together.

Here's an example.

The original patch.



Flipped horizontally.

Rotated 90 degrees counter-clockwise.

Tip: Edits to patches only apply to baked paint. Unbaked paint in the paint buffer is not affected.

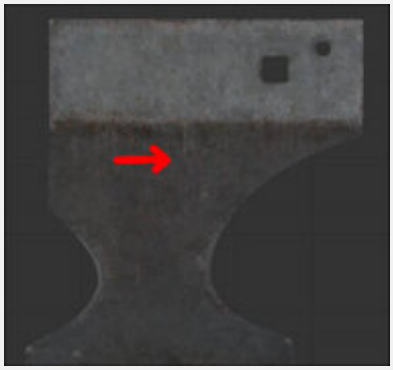
When you are copying patches onto other patches, unless your patches are identically shaped, you can see:

- blank areas at the edges, where the texture from the source patch runs out due to the change of shape, and
- traces of the "overpaint" around the edges of the patch.

You can also see these when transforming paint on patches unless your patches are square.

The overpaint is used to ensure that the patches fit together without visible gaps. You can view the overpaint for patches in the UV view. Turn the UV paint display on in the **Display Properties** dialog box (right-click on the canvas, select **Display Properties** from the dropdown menu, and turn on **Render UV Image**).

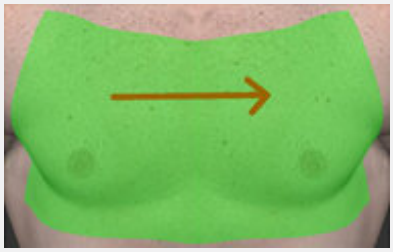
Note the blank patches and "bleed" from the overpaint areas.



The original patch.

Rotated 90 degrees.

Paint transforms only apply within each patch. If you have more than one patch selected, each patch transforms independently. For example, if you flip two side-by-side patches horizontally, each patch flips separately.

	
Two patches selected.	Flipped horizontally.

Copying and Pasting Patches

See Copying and Pasting Paint for details on how to copy and paste individual patches.

Limiting Patches in a Bake Point Node

You can reduce bake times in a Bake Point node by limiting the amount of Patches/UDIMs being baked by that node to only Patches that require a bake. See *Limiting Patches in a Bake Point Node* in the *Bake Point Node* in the *Mari Reference Guide*.

Viewing and Resizing Patches

1. Start Mari and open your project.
2. To open the UV view, click either the **UV** or **Ortho/UV** tab at the top of the canvas.

   The UV view displays all individual patches on the model, and labels the patches with the size format in an abbreviated mode, such as 4K instead of 4096x4096. The label also lists all active patch sizes in a comma-separated list.

3. Zoom in to view the details of a specific patch more easily.

   Once you zoom in enough, Mari displays the patch details (such as resolution and color depth).

4. If you select one or more patches and you want to switch the view to display all of the selected patches, press **A**.

   Mari focuses the view on the patches that are selected.

   If you don't have any patches selected, the view focuses on the model as a whole (changing to show the whole model).

Tip: If you find that the view starts cutting through the front clipping plane, try pressing **A**.

Resizing Patches

1. Select the patch(es) to resize.

Tip: You can select the patches in any view - you don't have to be in the UV view to do this.

2. From the **Patches** menu or the right-click dropdown menu on the canvas, select **Resize Selected** and then a resize option. You can select a specific resolution from the list (up to the maximum resolution for the channel), or choose to halve or double the resolution.

   If you try to double the size of a selection already at the channel's maximum resolution or halve the size of one at 256 x 256, nothing happens.

Note: If attempting to resize patches in a channel that contains layers shared from other sources, the shared layers are not resized; all other layers in the channel are resized.

Note: If you don't select a patch or a layer when resizing, Mari asks if you want to resize all patches or layers before continuing.

Using UDIM Values

Each patch represents one square unit in UV space; for instance, a square with dimensions 1x1. UDIM values are a way of representing the integer coordinates of that square, from the coordinates of its bottom-left corner in UV space.

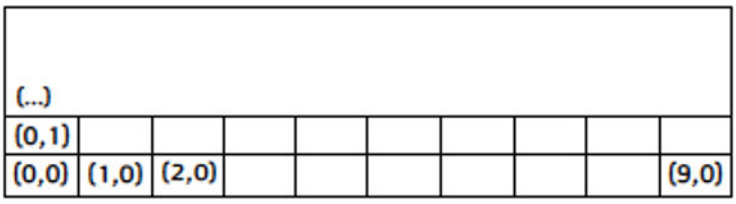
Channels can contain up to ten patches across, and any number of patches upwards. This means that patches can have U indices between 0 and 9 inclusive, and V indices can be zero or any positive integer. With these values, the UDIM for a patch can be calculated using the formula:

udim = 1001 + u + (10 * v)

Note: The U index limit of ten is currently hardwired.

The UDIM value of the bottom-left patch, which represents the UV space region (0,0) to (1,1), is 1001. The next patch to the right has a UDIM value of 1002, and the patch directly above the bottom-left is 1011. For example, the patch representing the UV space region (2,5) to (3,6) has a U index of 2 and a V of 5, so replacing the values in the formula, we get:

udim = 1001 + 2 + (10 * 5) = 1053



Extracting Patches to the Image Manager

To extract one or more patches to the **Image Manager**:

1. Select the patches to extract.
2. From the **Patches** menu, select **Extract Selected**, or right-click on the canvas and select **Patches** > **Extract Selected** from the dropdown menu.

   Mari copies the patches to the **Image Manager**. See Managing Images.

Tip: You can only copy patches up to 8K resolution.

Filling Patches with Color

To fill one or more patches with a color:

1. Set either your foreground or background color to the color you want to fill.

   See Managing Colors for details on selecting colors.

Video: Watch a demo on how to select the foreground and background colors.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

2. Select the patch(es) to fill.

3. From the **Selection** > **Fill** menu or from the right-click context menu on the main canvas, select either:
   - **Black** - to fill the patches with a constant black color.
   - **Gray** - to fill the patches with a constant gray color.
   - **White**- to fill the patches with a constant white color.
   - **Foreground** - to fill the patches with the foreground color.
   - **Background** - to fill the patches with the background color.
   - **Transparent** - to make the patches transparent.
   - **UV Mask**- to make patches a UV mask.
   - **Wireframe** - to fill patches with a wireframe.

   Mari fills the patches with the selected color, mask, or wireframe.

Flipping and Rotating Painting

1. Select one or more patches with baked paint.

   See Selecting Items for details on selecting patches.
2. From the **Patches** > **Transform** menu, select the transformation:
   - **Flip Vertical** - to flip the selected patch(es) vertically.
   - **Flip Horizontal** - to flip the selected patch(es) horizontally.
   - **Rotate 90 CCW** - to rotate the selected patch(es) 90 degrees counterclockwise.
   - **Rotate 90 CW** - to rotate the selected patch(es) 90 degrees clockwise.
   - **Rotate 180** - to rotate the selected patch(es) 180 degrees.

Note: The transform functions in the **Patches** menu are applied to selected patches for the current and selected layers. If you don't select a patch or a layer, Mari asks if you want to apply the transform to all patches or layers before continuing. To apply the transform functions to a single Paint node, double-click a Paint node and in the **Node Properties** palette, set the **General** > **Management** option to **Self**.

Mirroring Painting

1. Select a patch with baked paint.

See Selecting Items for details on selecting patches.

2. From the **Patches** > **Mirror** menu, select the mirror operation:
   - Mirror Left to Right - to mirror the selected patch from left to right.
   - Mirror Right to Left - to mirror the selected patch from right to left.
   - Mirror Top to Bottom - to mirror the selected patch from top to bottom.
   - Mirror Bottom to Top - to mirror the selected patch from bottom to top.

Note: The mirroring functions in the **Patches** menu are applied to selected patches for the current and selected layers. If you don't select a patch or a layer, Mari asks if you want to mirror to all patches or layers before continuing. To apply the transform functions to a single Paint node, double-click a Paint node and in the **Node Properties** palette, set the **General** > **Management** option to **Self**.

Tip: Experiment with flipping, rotating, and mirroring patches. Apply paint first to see what happens to the paint when the patch is changed.

Saving Images of the UV Masks

Mari has a function that allows you to save an image of the outline of a UV mask. You can then save these images for use as a mask in another program, if required.

Patches selected in UV view.

B/W UV masks in the **Image Manager.**

1. Select one or more patches.

   See Selecting Items for details on selecting patches.
2. From the **Patches** menu, select **UV mask to Image Manager**.

Mari takes a snapshot of the UV outlines on the selected patches and puts it into the **Image Manager**. It creates a separate image for each patch selected.

You can now use the images or save them to disk. See Managing Images for details on using the **Image Manager** palette.

Bleeding Across Patch Edges

By default, Mari creates an overspill area at the edge of UV patches to prevent seaming. In some cases, the overspill may not be large enough to prevent seaming artifacts, or you may get render artifacts if your renderer expects a
larger bleed area.

For example, from some angles, Mari's blacksmith tutorial asset shows a seaming artifact in the hair just above the left ear when viewed front on. This is because the bleed for that edge of the UV patch is slightly too small.

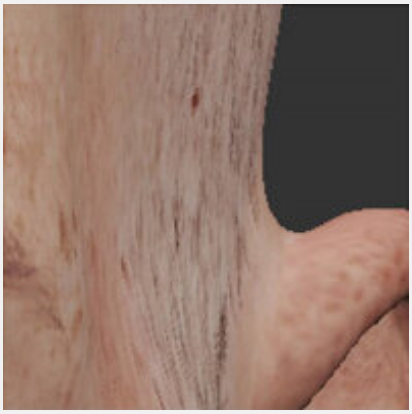
From front on, you can see a slight line at the patch edge.

This is caused by a too small overspill edge.

Tip: You can view the overspill bleed by turning on **Show UV Patches** in the Display Properties dialog box.

If you are experiencing problems with seaming artifacts (either within Mari or after rendering) try conducting a full patch bleed. This extends the bleed area across the entire UV patch.

UV patch after full bleed.



And the artifact is eliminated.

Note: Mari does not do a full bleed by default because it takes much longer to calculate.

For performance reasons, Mari divides large patches into smaller, more manageable 256x256 tiles. When baking paint into a channel, Mari always performs an individual edge bleed for every tile affected by the paint. The bleed operation fills the tile but does not affect surrounding tiles.

When bleeding is done per tile, some exceptional conditions may occur depending on how the tile lines up with the geometry shell:

- As bleed width is extended from the shell to the tile boundary, it may vary depending how close the shell boundary is to the tile boundary.
- Resizing a patch changes the tile grid causing it to line up differently with the geometry shell, which can affect bleed results.
- Tile corners that just touch the outer shell of the geometry can result in unexpected bleed behavior.

Edge bleeding per patch is different because it renders the entire patch first and then performs a full bleed on it. This avoids the conditions mentioned previously but requires much more memory and processor time, especially when bleeding large patches.

The per tile method is the default behavior and ensures that normal paint-bake operations are performed quickly. Full patch bleeding should only be used in situations where exported tile-bled patches are causing problems in third party applications.

If you see seaming artifacts at the edge of your patches, try doing a full bleed across the affected patches.

1. Select the patches to bleed.

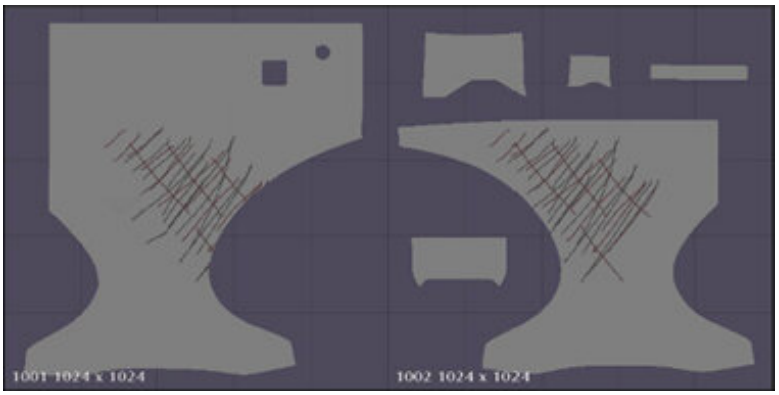
   See Selecting Items for details on selecting patches.
2. From the **Patches** menu, select **Bleed Patch Edges**.

Mari calculates a full bleed for the selected patches, taking the existing textures next to the patch on the model into account. This may take a few seconds.

Linking or Unlinking Patches

Mari has the ability to link patches together to provide a way of sharing the same underlying image data between multiple patches quickly and easily. A patch is given the same paint as the patch it is linked to if the linked patch is painted after they have been linked. This function can be found within the **Patches** palette. It is best to view the model in **UV** view when attempting to link or unlink patches.

The linked patches feature works best when the underlying UV maps are identical. If they are not, then the differing UV maps result in different edge bleeds, which could manifest as bleeding artifacts on linked patches. If you need to avoid edge bleeding artifacts, you can use the Whole Patch Projection option in the UV view.



Note: Please be aware that if painting on more than one linked patch, any overlap of paint can cause baking artifacts. If you have trouble painting on linked patches because of these overlaps, you can lock the patches you don't want to paint on directly.

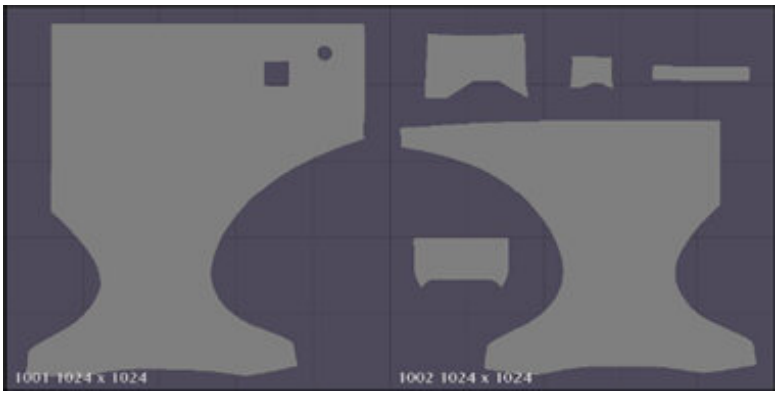
Tip: If you notice paint conflicts and you are already taking care to avoid overlapping paint, try locking neighboring linked patches to the one you are painting on to avoid conflicts.

To link together two or more patches:

1. Within the **Patches** palette, select the patches you want to link together. You can select the patches by clicking on them in either the palette or on the canvas.

   This is best done in UV view for easier selection visibility.
2. Click the **Link Selected Patch Images** icon to link the selected patches.

   In the UV view, a colored background displays behind the selected patches to denote which patches have been linked.



To unlink patches:

1. Within the **Patches** palette, select the patches that you want to unlink from the linked selections.
2. Either:
   - click the **Unlink Selected Patch Images** icon to unlink the selected patch from any other patches,
   - right-click the **Patches** palette and select **Unlink All Linked Patch Images**, or
   - right-click on the canvas and select **Patches** > **Link | Unlink All Linked Patch Images.**

   In UV view, the colored background no longer shows behind the patches.
3. If you want to specify that a particular patch be used as a "master" patch, from which the other linked patches can be initially copied, you can do this by first selecting the desired "master" patch on its own and linking it using the **Link Selected Patch Images** button. You can then select the master patch and the rest of the patches to re-link them using the **Link Selected Patch Images** button again.

Note: When linking patches, if any patches have been previously linked, Mari determines which set of linked patches has the most patches in the selection set and retains those links while re-linking the other selected patches.

Tip: You can find all of the options discussed in this section in the right-click dropdown menu on the canvas.

Working with Objects

Mari allows you to load multiple objects and child objects into a scene, and paint them at the same time. This lets you paint objects in context - for example, you can paint all the elements of a character's costume by loading them up onto the character. You can show and hide objects, so you can easily switch between working only viewing a single object, and seeing how it fits with the other objects.

You can load multiple copies of a single object into the project. This allows you to, for example, build up a road scene with twenty sections of pavement, two lamp posts and a fire hydrant from three basic models (pavement section, lamp post, and hydrant).

When you create a project, you must load in an object. Once you've created the project, you can add more objects to it.

Initial project, with only the blacksmith torso model.

With other objects loaded.

If a project has two or more objects, you can remove objects from it. All projects must have at least one object. Removing an object from a project has no effect on the source file for the object. The objects within a project are listed in the **Objects** palette.

Video: For more on working with objects, have a look at the Managing Geometry in a Project video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Channels and Shaders on Multiple Objects

Each object in the project has its own set of channels. When you add channels, you have the option to apply the action to all objects in your project.

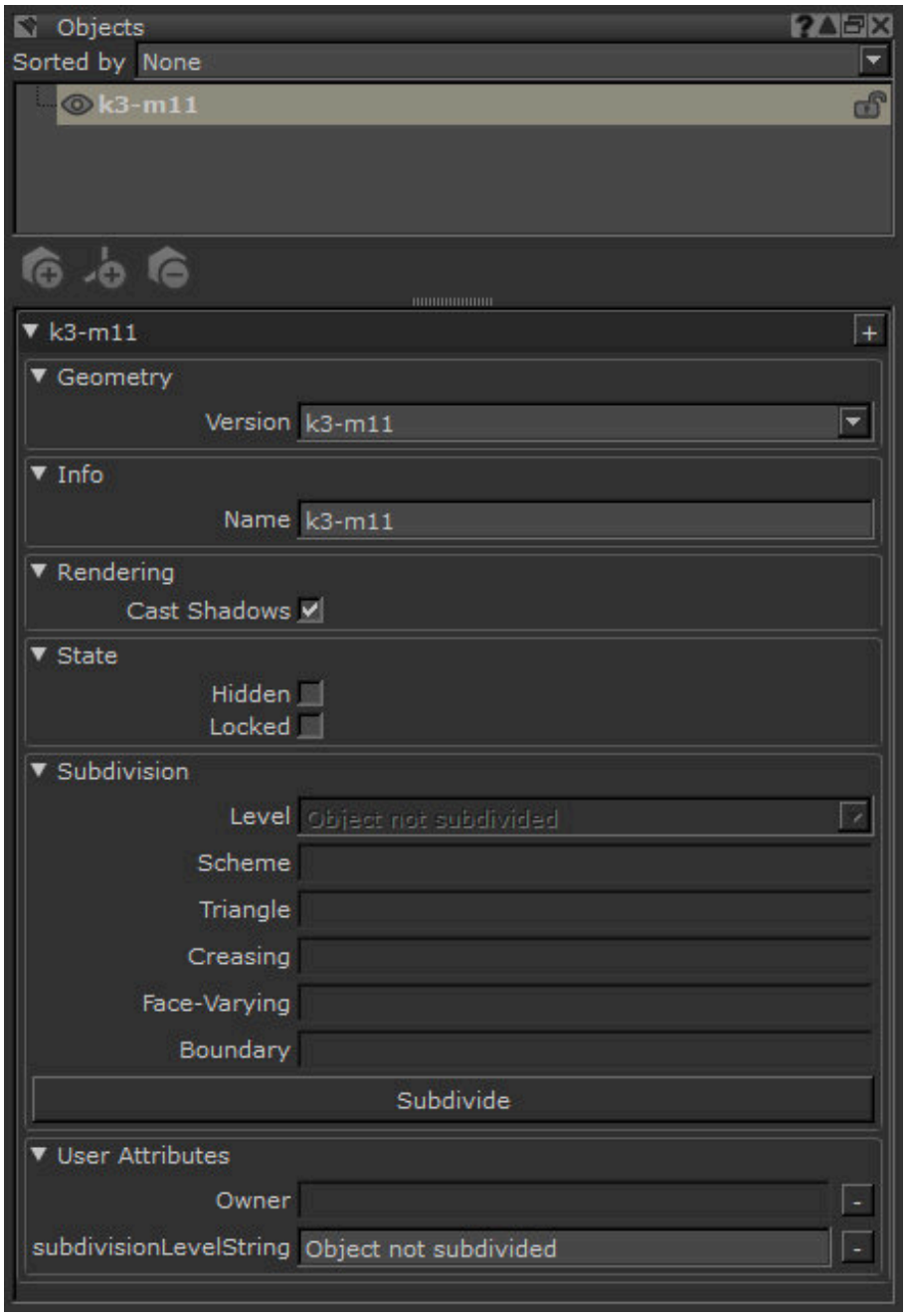
When you add a new object, Mari creates a single channel, **diffuse**, to hold the diffuse color. You can add more channels as you would normally.

Similarly, each object has its own shaders. As with channels, there are default shaders but you must set up your custom shaders separately for each object. If you want to copy shaders from the **Shaders** palette of another object, use the **Ctrl**/**Cmd**+**C** and **Ctrl**/**Cmd**+**V** shortcuts to copy the shader and paste it into the **Shaders** palette for the current object.

Adding Objects

1. Start Mari and open your project.
2. Open the **Objects** palette.

   This displays all the objects in your project.

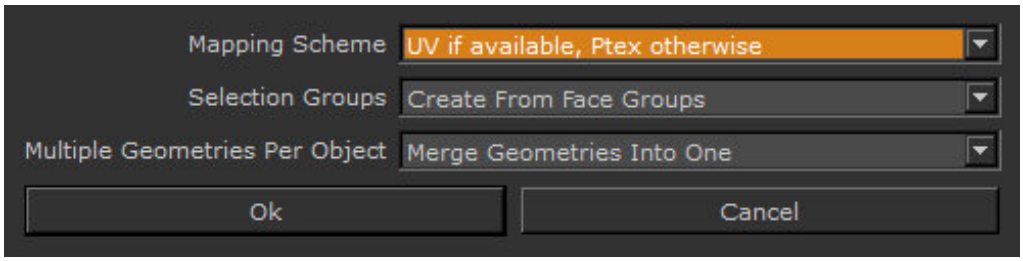


3. Add a new object by:
   - clicking **Objects** > **Add Object** from the menubar,
   - clicking in the **Objects** palette,
   - right-clicking on the empty space in the **Objects** palette, and selecting **Add Object** from the dropdown menu.

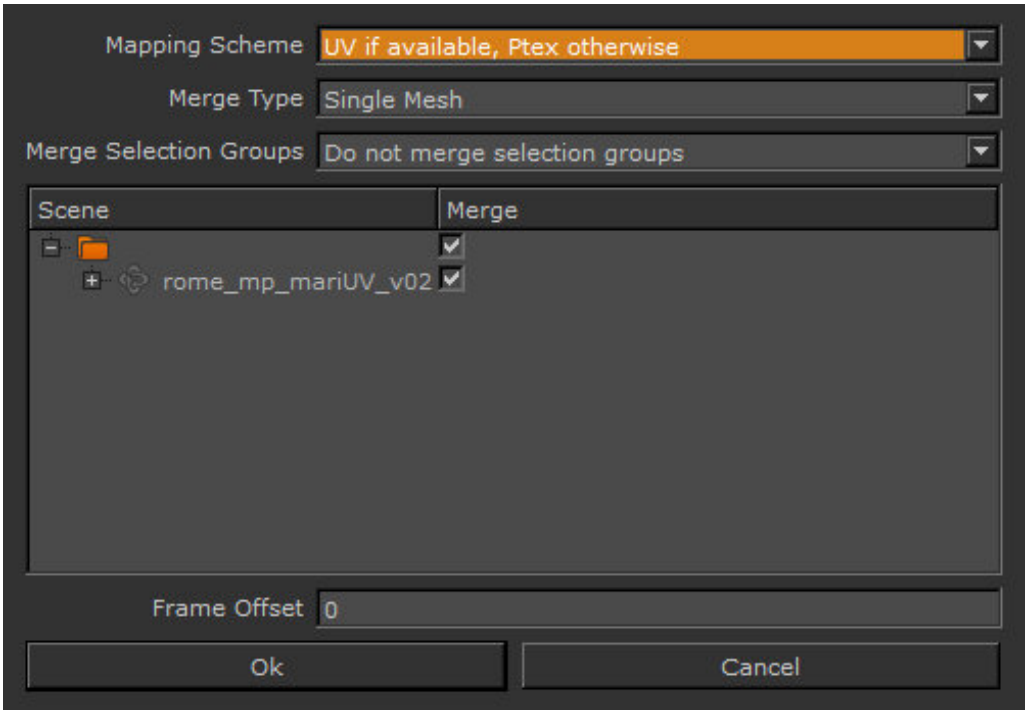
   The **Add Objects** dialog box displays, allowing you to select the new object file to add.
4. Select the file to add and click **OK**.

   The **Add Object Settings** dialog displays.

- If the object you are adding is an **.obj** file, set the **Mapping Scheme**, **Selection Groups**, and **Multiple Geometries Per Object** fields to specify how you want the object to be treated on creation in the project.



- If the object you are adding is an **.fbx** or **.abc** file, set the **Mapping Scheme**, **Merge Type**, and select the objects to import from the object hierarchy. You also need to select which frames you want to load and, in the case of Alembic files, the **Frame Offset**.



- If the object you are adding is a **.ptx** file, set the **Merge Type** field to specify how multiple geometries in the object are handled.

Mari adds the new object to your project. Your new object has a single channel, **Diffuse** (to hold the diffuse color). You probably want to add additional channels (see Creating and Deleting Individual Channels).

Tip: You can add a file that already exists in your project. In this case, the new object is added as a second instance of the same file, and is initially on top of the first instance.

Note: For more information on how Alembic and FBX objects are handled in Mari, refer to the Alembic and FBX section. For more information on how Ptex objects are handled in Mari, refer to the Ptex

Adding Child Objects

1. Start Mari and open your project.
2. Open the **Objects** palette.

   This displays all the objects in your project.
3. Add a new child object by right-clicking an object in the **Objects** palette, and selecting **Add Child Object** from the dropdown menu.

   The **Add Objects** dialog box displays, allowing you to select the new object file to add.
4. Follow the rest of the steps from 4 onwards in the 'Adding Objects' steplist above.

   The object is added as a child to the selected object.

Filling Objects with Color

To fill one or more objects with a color:

1. Set either your foreground or background color to the color you want to fill.

   See Managing Colors for details on selecting colors.

Video: Watch a demo on how to select the foreground and background colors.

2. Select the object(s) to fill.

3. From the **Selection** > **Fill** menu or from the right-click context menu on the main canvas, select either:
   - **Black** - to fill the patches with a constant black color.
   - **Gray** - to fill the patches with a constant gray color.
   - **White** - to fill the patches with a constant white color.
   - **Foreground** - to fill the patches with the foreground color.
   - **Background** - to fill the patches with the background color.
   - **Transparent** - to make the patches transparent.

   Mari fills the objects with the selected color.

> **Tip:** You can also drag and drop a color onto an object directly, simply click on a color swatch and drag and drop it onto the object.

Adding and Removing Object Versions

Objects within your project are versioned. If the source file for the model changes, you can load a new version of the file – Mari updates the model in your project, and keeps all the textures you've painted on it.

When you add a new version of a model, Mari assumes that the patch numbering hasn't changed, and projects the existing textures over the updated geometry. Whenever you add a new version of an object, check any textures you've already painted, especially if the patch numbering has changed.

All objects start with an initial version. You can add and remove versions, as long as at least one version of the object remains. Mari does not enforce any other restrictions - for example, if an object has three versions, you can delete any of the three versions.

> **Tip:** Mari does not restrict what you can add as a new version (as we have no way of knowing what kinds of changes to the models you may need to make as you develop the asset!). As a result, you can add **any** object file as a new version of an existing file – so be careful that you don't add the wrong file by mistake.

To add a new version to an existing object:

1. Right-click on the object in the **Objects** palette, and select **Add Version** from the dropdown menu.

The **Add Version Settings** dialog box displays.

2. Select the file for the updated object and click **Open**.

   Mari adds the new version to the object's history and updates the display to show the new version.

> **Tip:** Check any existing paint on the object, as it may have been distorted by the move to the new mesh.

To remove an object version, right-click on the object in the **Objects** palette and select **Remove Version** from the dropdown menu.

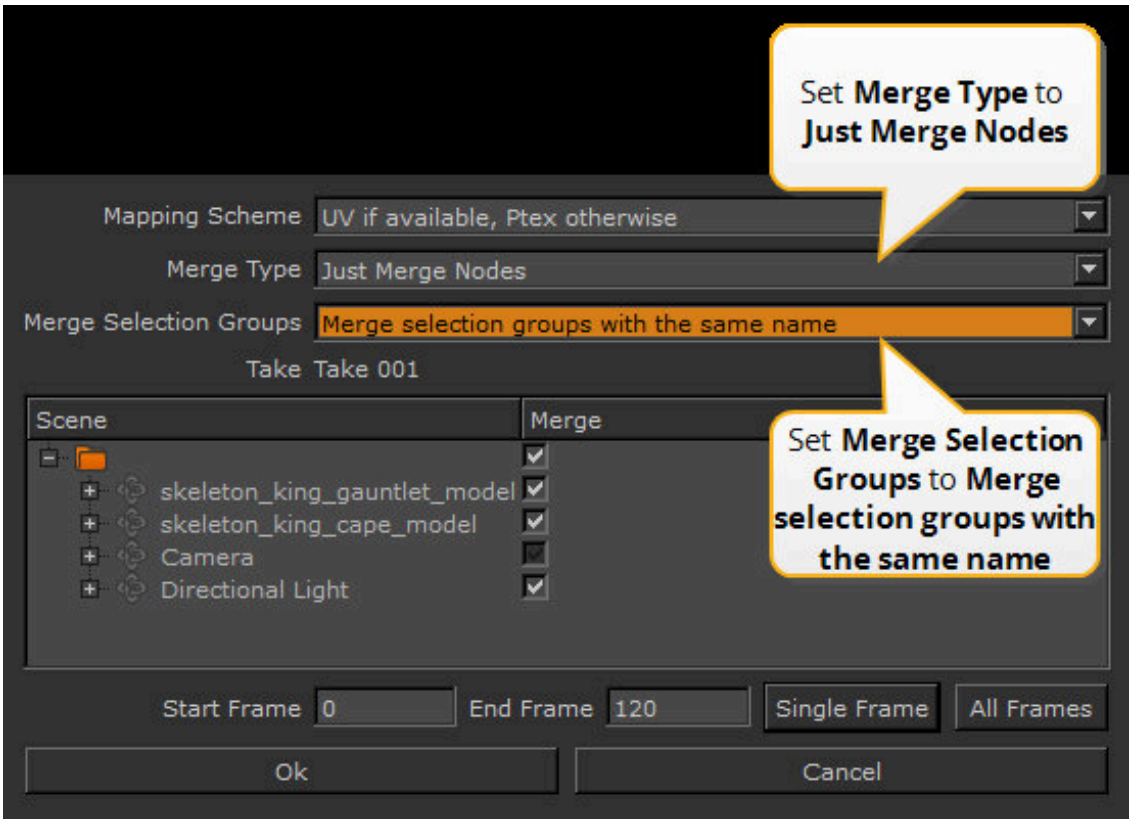
Saving Selection Groups Between Object Versions

Mari preserves selection groups between object versions. If you have several versions of an object containing different selection groups, here is how you can import them in Mari:

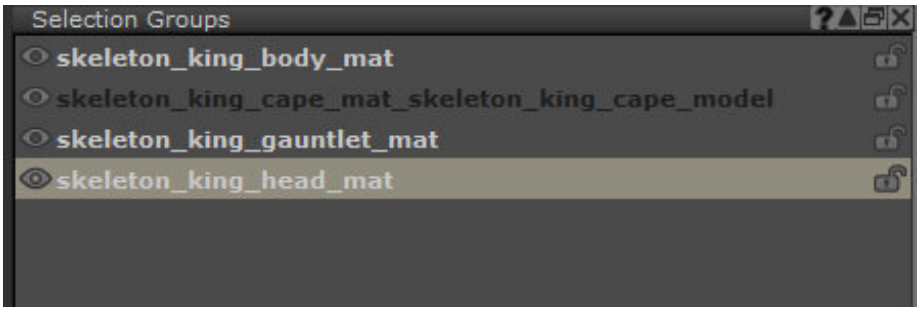
1. Launch Mari and open your project.
2. In the **Objects** palette, right-click on the required object and select **Add Version** from the dropdown menu.

   The **Add Version** dialog opens.
3. Select the file of your object version that contains selection groups and click **Open**.

   The **Add Version Settings** dialog opens.
4. In the **Merge Type** field, select **Just Merge Nodes**.
5. In the **Merge Selection Groups** field, select **Merge selection groups with the same name**.

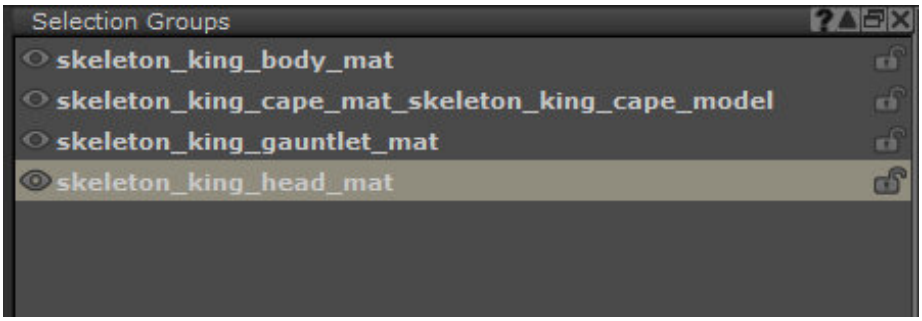


The file of your object version is imported and the selection group names of this latest version are displayed in black font in the **Selection Groups** palette. The black font indicates selection groups cannot display on the model as the required object version is not selected in the **Objects** palette.



6. In the **Objects** palette, in the **Geometry** section, select the version of the object you just imported, in this case the most up-to-date version.

   The selection group names are displayed in gray font in the **Selection Groups** palette. The gray font indicates you can display the selection groups on the model. The selection groups contained in the selected object version display on the model.



7. In the **Selection Groups** palette, click a selection group name or **Shift** + select several selection group names.

   Enable the **Select** tool on the **Tool** toolbar to see your selection reflected on the canvas, highlighted in green.

Adding Locators for Objects

You can load locators and child locators in order to transform an object by rotating, scaling, or moving it. This is especially helpful with spherical projects, where you may want a procedural to have a 3D world or camera space position that it can use in its calculations.

Tip: Move locators in the viewport using the Transform Selected Object tool, which uses the locator's **Local** space by default.

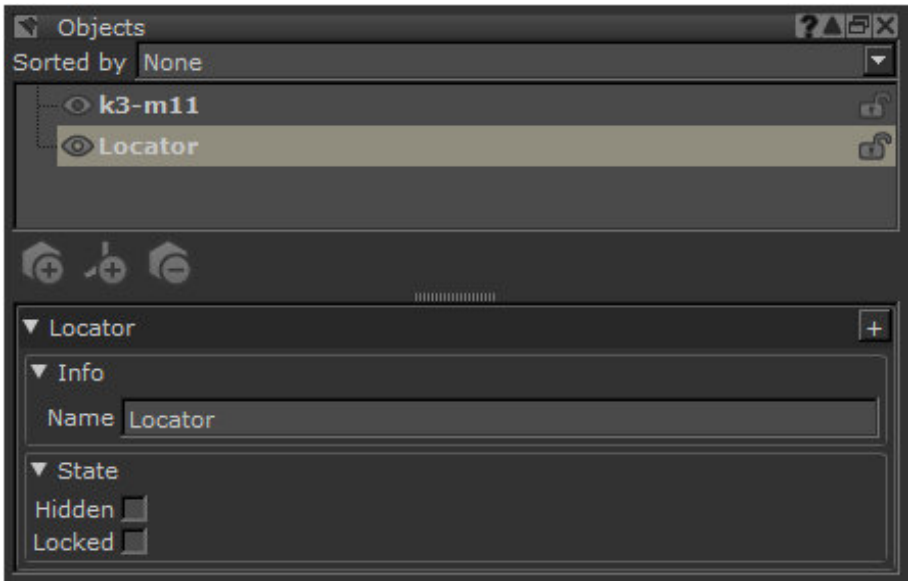
Note: Currently, linking a procedural to a locator is something that can only be done through the **Python Console**. There is no GUI option for this functionality at present.

Adding Locators

1. Start Mari and open your project.
2. Open the **Objects** palette.

   This displays all the objects in your project.
3. To add a new locator, right-click in the **Objects** palette and select **Add Locator** from the dropdown menu, or click on the **Objects** palette.

The locator is added to the bottom of the objects list.

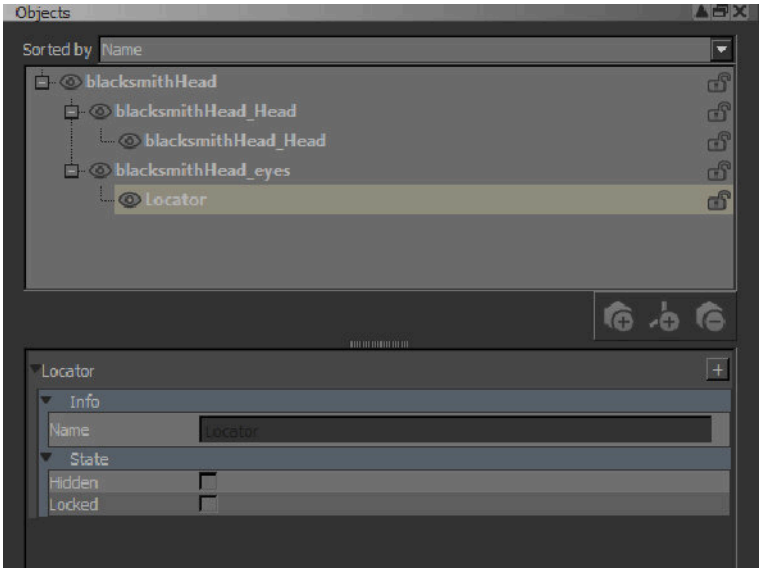


Adding Child Locators

1. Start Mari and open your project.
2. Open the **Objects** palette.

   This displays all the objects in your project.
3. To add a new child locator, right-click the object in the **Objects** palette and select **Add Child Locator** from the dropdown menu.

   The child locator is added beneath the selected object.



Adding Geo-Channels to Objects

You can add Geo-Channels to your objects to define arbitrarily named geometry specific data such as ambient occlusion, and curvature in your material creation workflow. Since geo-channels are geometry-agnostic, you can use the same material across a variety of objects while ensuring the material looks realistic for the specific object. Geo-Channels must be added to your object in order for Geo-Channel nodes and layers to function. See Geo-Channel Node for more information.

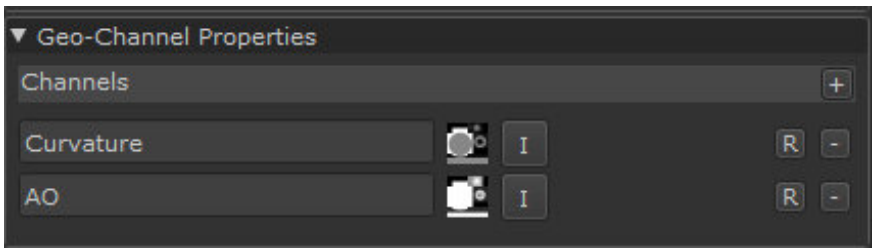
Tip: You can also export baked data from Bake Point nodes directly into Geo-Channels, to act as an intermediate caching point for complex Node Graph setups. See the Bake Point Node and Working with Nodes for more information.

Adding Geo-Channels to your Object

1. Open the **Objects** palette.
2. Select your object from the list of objects in the Objects palette.
3. Scroll down to **Geo-Channel Properties**.
4. Click the + button next to **Channels** to add a new geo-channel.

   A new geo-channel appears in the list of geo-channels.
5. Click a geo-channel to enter a new name.
6. Click the I next to the specified geo-channel to import an image set for that geo-channel.

Note: The imageset bit-depth for geo-channels automatically converts to the bit depth of the imported texture.



The **Geo-Channel Properties** inside the **Objects** palette.

Importing Modo Render Bake Results as a Geo-Channel

If you don't have any rendered maps to use as Geo-Channels, to use Mari's built-in Modo renderer. The built-in Modo renderer allows you to bake textures for your objects within Mari.

To import the baked results back into Mari as a geo-channel:

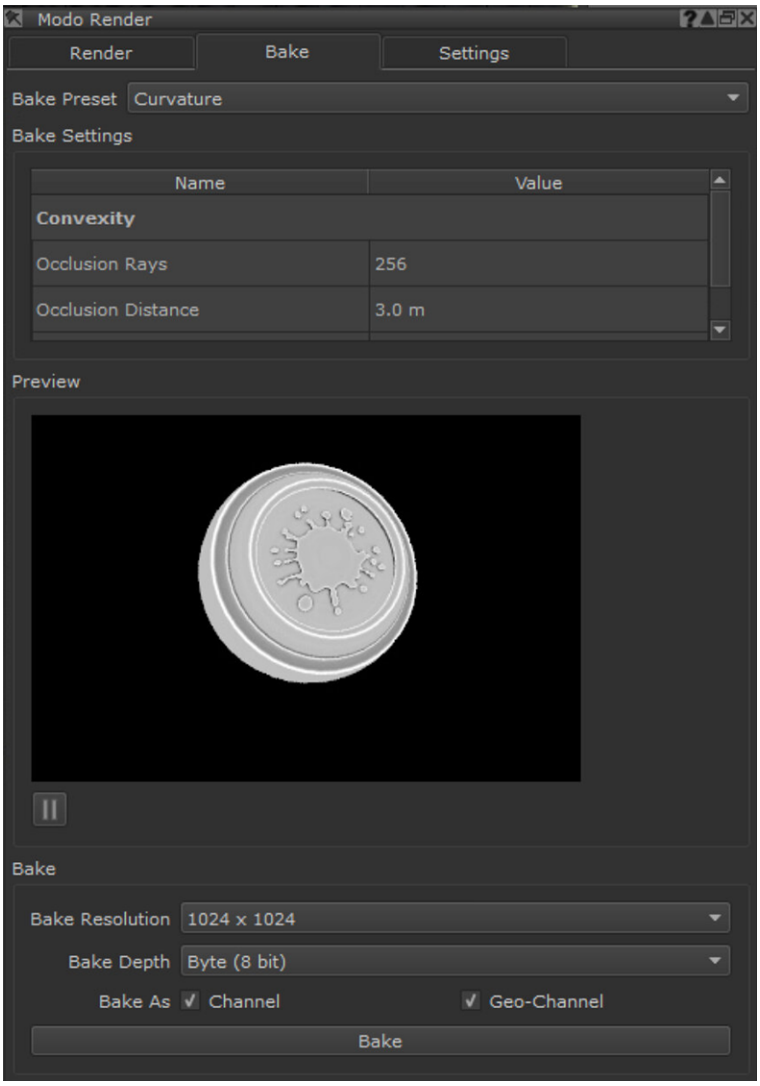
1. Open the **Modo Render** palette.
2. Select the **Bake** tab.
3. Select a preset from **Bake Preset**.

   A preview of the bake is generated.
4. Enable **Bake As** > **Geo-Channel** option.

   This imports the baked result back into Mari as a Geo-Channel.

> **Warning:** This destructively overwrites the contents of a Geo-Channel with the same name, or automatically creates a new Geo-Channel if one of the same name doesn't exist.

5. Click **Bake**.



The **Bake** tab inside the **Modo Render** palette, with the Curvature **Bake Preset** selected.

Duplicating Objects

You can duplicate child objects but not parent objects, locators, or child locators. Duplicating an object creates a copy of it and adds it to the bottom of the object list in the **Objects** palette. When duplicating, you can choose to copy just the object or both the object and its associated shader network.

1. Open the **Objects** palette and select the object you want to duplicate.
2. Either:
   - select **Objects** > **Duplicate** > **Object Only** or **Object And Shader Network**, or
   - right-click on the object and select **Duplicate** > **Object Only** or **Object And Shader Network** from the dropdown menu.

3. If you chose:
   - **Object Only** - Mari duplicates only the selected object without any of the shader, channel, or layer data.
   - **Object And Shader Network** - Mari duplicates the selected object, including all associated shaders, channels, and layers that belong to the object and the data they contain.

   The duplicated object is added to the bottom of the object list.

Removing Objects and Locators

1. To remove an existing object, child object, locator, or child locator, select it in the **Objects** palette and:
   - either click **Objects** > **Remove Object** from the menubar,
   - right-click the object and select **Remove Object** from the dropdown menu, or
   - click on the **Objects** palette.

   Mari asks you to confirm the deletion.
2. Click **Yes** to remove the object.

   You can remove any object from a project, as long as there is at least one object remaining afterwards.

Moving, Viewing, and Locking Objects

Mari allows you to move objects freely across the canvas. For example, you might need to move an object which appears on top of an existing object.

> **Note:** Where objects appear on the canvas depends on the configuration of the model file.

To move an object:

1. Select the Transform Selected Objects tool.

   The transform handles (gimbal) appear at the position of the object's pivot point that was saved in the original **.obj**, **.fbx**, or **.abc** file.

2. Drag the object, or use the transform handles to move the object along one of the handle directions in the object's local space.

Initial state.	Apron moved to the left.

Clicking on an object selects it. You can then configure and edit it, for example, to change to paint on a different channel.

3. The **Objects** palette lets you show/hide and lock/unlock the object:

- To show or hide an object, click .
- To lock or unlock an object, click .

4. If you move the object accidentally or want to reset the object back to its original position, **Shift**+click to reset the object, press the apostrophe key (’) on the keyboard to reset all objects, or select **Objects** > **Reset Object Transform** to return the selected entity in the **Objects** palette back to its original coordinates.

> **Tip:** You can also use the **Transform Selected Objects** tool to move lights around (see Configuring the Lighting).

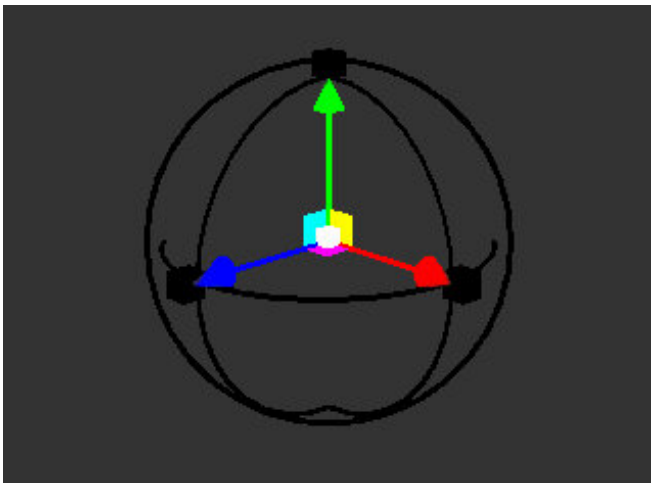
Using the Gimbal

When you want to move an object using the **Transform Selected Objects** tool along either **Global** or **Local** space, a gimbal appears at the selected object's pivot point. A gimbal allows you to drag on transform handles to move, rotate, or scale the selected object. The colors of the handles correspond to the axes available in 3D space: red transforms the x-axis, green transforms the y-axis, and blue transforms the z-axis. Mouse over parts of the gimbal to highlight the control for each axis.

Moving an Object with the Transform Handles

Drag the arrow to move an object on the given axis:

- X translate (red arrow),
- Y translate (green arrow),
- Z translate (blue arrow), or
- Free translate (center of the gimbal widget)



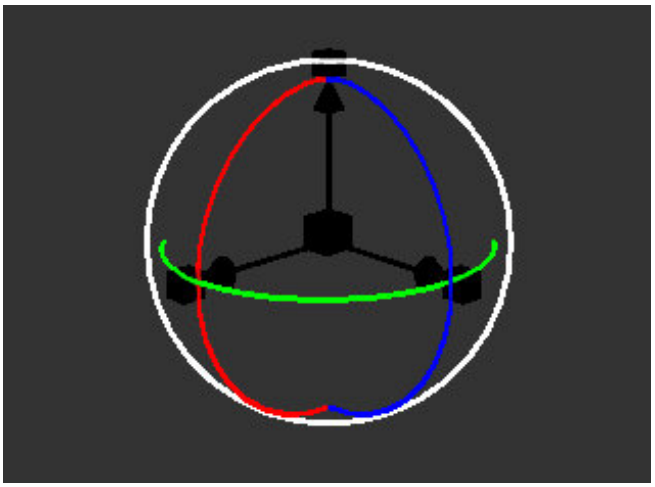
Note: Free translation is a 2D (x, y) transformation in the angle of projection of the camera, and can also be achieved by clicking anywhere on the model.

Tip: By default, the gimbal moves the selected object along its local space. You can change this to **Global** (world) space using the dropdown in the **Transform Selected Objects** Tool Properties Toolbar.

Rotating an Object with the Transform Handles

Drag a specific axis ring to rotate either:

- X rotation (red circle),
- Y rotation (green circle),
- Z rotation (blue circle), or
- from the camera angle (white circle).

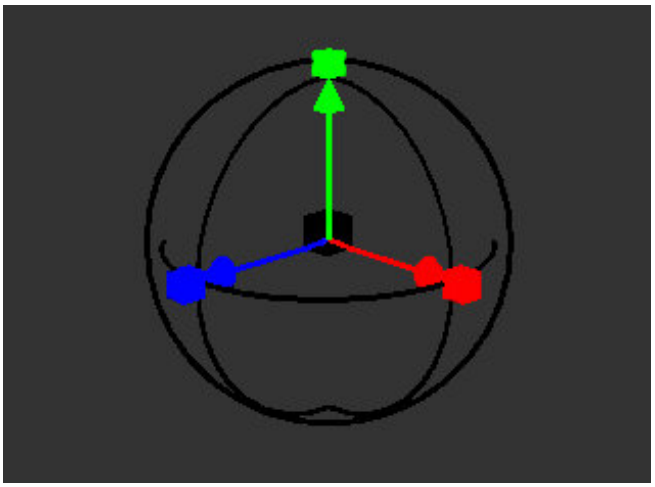


Tip: Hold **CTRL** while rotating with the **Transform Selected Objects** tool to snap the rotation of the object or locator. Adjust the stepping angle in **Edit** > **Preferences** > **Navigation** > **Stepping Angle**.

Scaling an Object with the Transform Handles

Click and drag a single axis square to scale on that axis or **Ctrl**/**Cmd**+**Shift** and drag any axis square to scale uniformly on all axes:

- X scale (red cube),
- Y scale (green cube), or
- Z scale (blue cube).



Calculating Ambient Occlusion

Mari does not automatically calculate the ambient occlusion for objects. This is because ambient occlusion can take a long time to calculate for complex objects, and it is not required unless you are using ambient occlusion for masking or shading.

If you need ambient occlusion for an object, you must calculate it separately for each object.

To calculate the ambient occlusion for an object:

1. Select the object in the **Objects** palette.
2. Either:
   - select **Objects** > **Ambient Occlusion** from the menubar, or
   - right-click an object in the **Objects** palette, and select **Ambient Occlusion** from the dropdown menu.

   Mari calculates the ambient occlusion for the object in the current project. Depending on the size and complexity of your model, this may be a good time to take a break.

Subdividing Objects

Subdividing the surface of an object smooths the surface of a low polygon mesh to create a smoother surface with many more faces, and is usually divided into either quadrilaterals (quads) or triangles. Mari uses OpenSubdiv to create static subdivided surfaces, as well as to load and work with subdivided geometry. You can determine the **Level** of subdivision you want to apply from a choice of levels **1**-**3**. The levels represent the number of quads or triangles into which your mesh is divided for each level, and the exact number is dependent on the subdivision **Scheme** you choose. For example, a **Catmull-Clark** method at level **1** may produce more quads or triangles after subdivision than a **Bilinear** method at level **2**.

> **Note:** If you instruct Mari to calculate the subdivision using a background process, you need to manually select the **Level** to apply once the calculation is complete.

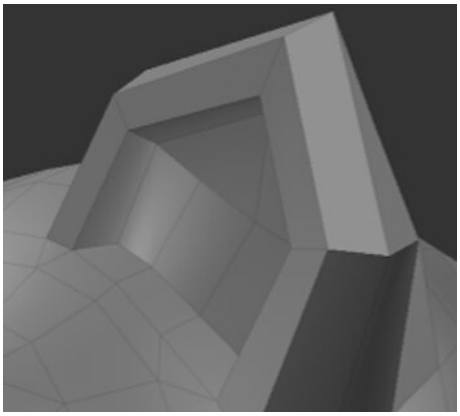
The manner in which the surface is smoothed is calculated based on a subdivision method. Mari offers three methods: **Catmull-Clark, Loop**, and **Bilinear**. What method you choose may be, in part, determined by whether your mesh is comprised of quads or triangles, as some methods, such as **Loop**, only works correctly

with meshes comprised of a certain type of polygon. In the instance of the **Loop** method, the polygon type is triangle.

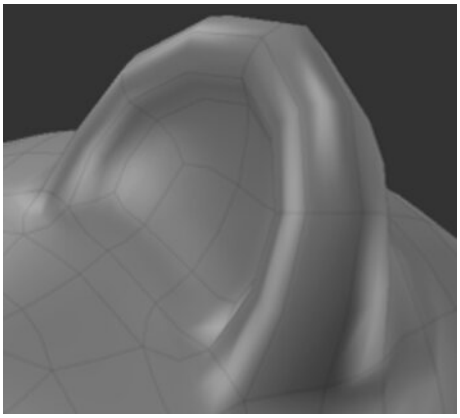
Note: For more information, see http://graphics.pixar.com/opensubdiv/docs/subdivision_surfaces.html.

Warning: Subdivision increases the number of polygons exponentially to the subdivision level, which means that if a geometry is already quite dense and you attempt to subdivide it to a high level (**2** or **3**), Mari may become slow or unresponsive.

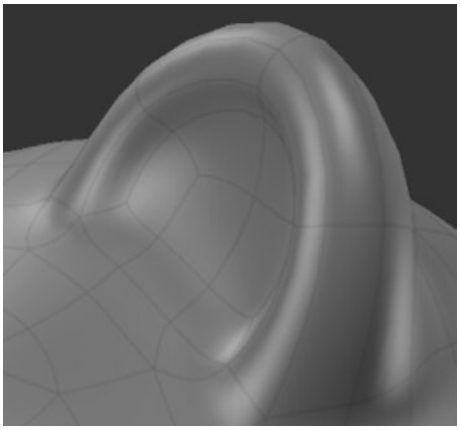
Below are examples of the subdivision levels on an un-subdivided mesh:



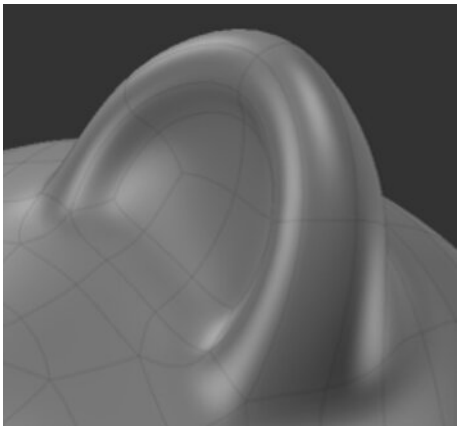
No subdivision applied (level 0)



Level 1 subdivision applied using the Catmull-Clark method



Level 2 subdivision applied using the Catmull-Clark method



Level 3 subdivision applied using the Catmull-Clark method

After subdivided geometry is calculated in Mari, the subdivided data is cached in the project. Whenever the subdivision level is changed, Mari loads the cached subdivided data and renders it. When subdivision is re-calculated, the cached data is deleted and the new data, with new subdivision settings, is cached.

Warning: If you have already created masks on an un-subdivided model and then choose any level of subdivision, the masks appear offset from where they did on the un-subdivided model. This is due to the nature of subdivision and the way it "distorts" the UV layout from that of the original model because the UV shells are also subdivided with the 3D mesh.

To subdivide an object in your project:

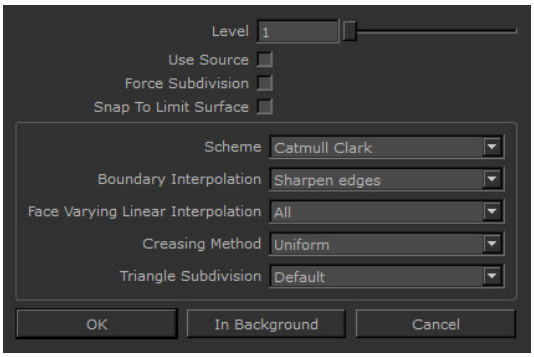
1. Select the object in the **Objects** palette.
2. In the Subdivide section, click **Subdivide**.

   Right-click and select **Subdivide** from the context menu.

   OR

   From the menubar , select **Objects** > **Subdivide**.

   The **Subdivide** dialog displays.



3. Choose the **Level** of subdivision applied to your object. You can choose from levels **1**-**3**.
4. Enable the **Snap To Limit Surface** checkbox to snap the last level of subdivision to the limit surface, if required.
5. Enable the **Use Source** checkbox if you want to use the subdivision options from the source geometry file.
6. Enable whether to use **Force Subdivision** or not.

   When enabled, Mari forces subdivision the geometry no matter how the subdivision flags are set.
7. Choose the Subdivision **Scheme** to determine how you want the object to be subdivided:
   - **Catmull-Clark** - a subdivision model that creates a new mesh that consists of only quadrilaterals.
   - **Loop** - a subdivision model that supports meshes consisting of only triangles, and divides each triangle into four smaller ones.
   - **Bilinear** - a subdivision model that creates quadrilaterals with non-planar vertices.

8. Choose how Mari handles the **Boundary Interpolation** (for boundary edges and vertices):
   - **None** - no boundary edge interpolation should occur, instead boundary faces are tagged as holes so that the boundary edge-chain continues to support adjacent interior faces but is not considered to be part of the refined surface.
   - **Sharpen edges** - all the boundary edge-chains are sharp creases, boundary vertices are not affected.
   - **Sharpen edges and corners** - all the boundary edge-chains are sharp creases and boundary vertices with exactly one incident face are sharp corners.
9. Select which **Face Varying Interpolation** to use:
   - **None** - smooths everywhere the mesh is smooth.
   - **Corners only** - sharpens (linearly interpolate) corners only.
   - **Corners plus1** - **Corners only** + sharpening of junctions of 3 or more regions.
   - **Corners plus2** - **Corners plus1** + sharpening of darts and concave corners.
   - **Boundaries** - linear interpolation along the boundary edges and corners
   - **All** - linear interpolation of everywhere (boundaries and interior).
10. Choose how Mari interpolates the sharpness of incident edges by setting the **Creasing** option:
    - **Uniform** - applies regular semi-sharp crease rules.
    - **Chaikin** - applies Chaikin semi-sharp crease rules. Chaikin's curve subdivision algorithm improves the appearance of multi-edge semi-sharp creases with varying weights.
11. Set the **Triangle Subdivision** option:
    - **Default** - default **Catmull-Clark** scheme weights.
    - **Smooth** - smooths triangle weights.
12. Once you've determined how you want Mari to subdivide the selected object, choose whether Mari calculates the subdivision in the foreground or background:
    - **OK** - a progress bar displays as Mari calculates the subdivision. Once complete, the level of subdivision specified is applied to the object immediately.
    - **In Background** - the subdivision calculation is submitted to a background process and no progress bar is displayed. The subdivision **Level** must be applied manually in the **Objects** palette when the "running man" icon disappears from the status bar.
      Go back to the **Objects** palette and, with the subdivided object selected, click **Subdivision** > **Level** in the object properties pane and choose the level of subdivision you want to apply.

> **Note:** When the level of subdivision you chose in the **Subdivide** dialog has completed, it appears in the **Level** dropdown and you can select it at any time. If you chose the highest level of subdivision for the object, you can select any lower level of subdivision from the dropdown as well.

The object is subdivided and both the object on the canvas and the object properties in the **Objects** palette are updated to reflect this.

Note: Mari detects whether to subdivide each Alembic geometry piece, based on subdivision flags set in the file.
You can also force the subdivision of Alembic geometry pieces no matter how the subdivision flags are set, refer to the *Mari Reference Guide* in the *Subdivide Dialog* section.

Shadow Settings

Each object in your scene casts shadows if depth projection is not disabled under **Preferences**. To display shadows, you must enable at least one light to cast shadows by checking the **Render Shadows** box in the **Lights** palette before adjusting the settings in the **Objects** palette. If this is not selected, objects with **Cast Shadows** enabled still don't appear to be casting shadows.

If **Shadow Maps > Allowed** have been enabled in the Mari Preferences Dialog (see the Mari Preferences Dialog section in the Mari Reference Guide), there are two related options that can be set to customize the shadow settings. **Resolution** and **Layer Count** control the texture resolution and shadow coverage.

Both these options are dependent on the graphics card. Mari automatically attempts to set this option at the resolution that works best for your setup. If there aren't sufficient graphics memory resources, Mari degrades the quality of the shadows so as to use fewer resources until shadows can be rendered.

To allow your model to cast shadows:

1. Select the object in the **Objects** palette.
2. Under **Rendering**, enable **Cast Shadows.**

   Each object in your scene has the setting **Cast Shadows**.

   Selecting this checkbox casts shadows from your models onto other objects in the scene, based on your current lighting setup.
3. From the **Project Controls** toolbar, select one of the two following lighting modes: **Basic with Shadows** or **Full with Shadows**.

Note: Using variance maps allow the shadow edges to be smoothed out, but may sometimes cause
rendering artifacts.
To prevent these artifacts from occurring, disable the **Use Variance Maths** checkbox under **Mari Preferences Dialog** > **GPU** > **Shadow Maps**. See the Mari Preferences Dialog section in the *Mari Reference Guide*.

Ptex

Ptex (Per-face Texturing) is a texture mapping system developed by Walt Disney Animation Studios as an alternative to traditional methods of texture mapping. Ptex addresses various issues associated with traditional texture mapping by eliminating the need for artists to manually create UV maps.

With Ptex, each face of a geometry has its own unique texture. This allows for the possibility of dynamically changing the resolution of textures on individual groups of faces.

Ptex files contain the per-face texture and face/edge adjacency data along with the geometry metadata, which can be useful for previewing textures and checking whether they are valid for a given geometry.

Video: Have a quick look at the Working with Ptex Projects video to learn how to create and work with Ptex projects in Mari.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Ptex Support

UV patches are 2D textures mapped onto regions of 3D geometry according to UV maps created by the artist. In addition to working with traditional UV maps, Mari also supports Ptex. With Ptex, UV patches are no longer a valid concept since there are no explicit UVs associated with the model. As a result, UV patches have been replaced with a UV atlas of Ptex faces.

When a new Mari project loads a UV-less **.obj** file, or a **.ptx** (Ptex) file containing geometry, every face of the geometry has a texture assigned to it, and in turn, every texture is assigned to an unused region of a UV atlas.

The packing of these face textures occurs in order of descending size, and because of this, adjacent faces with differing texture resolutions may not be spatially coherent in the UV atlas. Therefore, painting in the UV view does not make logical sense.

Currently, Ptex support in Mari includes:

- starting a new project with either a UV-less **.obj**, or a **.ptx** file (containing geometry metadata)
- optionally assigning each surface defined in the **.obj** file its own Ptex layout
- polygons of arbitrary size and shape in **.obj** files
- specifying uniform, or worldspace texel density texture resolutions of faces
- painting onto Ptex geometry in 3D views
- resizing selected Ptex face textures from 1x1 (2x2 for triangles) all the way up to 8192x8192
- creating new channels that share Ptex face texture resolutions
- exporting a channel to a **.ptx** file, with the option to include geometry metadata and face/edge adjacency information
- importing **.ptx** files into channels that are topologically equivalent
- preserving paint in multiple Ptex geometries, when creating new geometry versions from **.obj** files, provided they are topologically equivalent
- Python bindings for Ptex project creation, per face operations, and import/export functionality. Refer to the Python HTML documentation within **Python** > **API** for further information
- allowing face texture resizing and data format conversion on import.

Limitations of Mari's GPU Previews with Ptex

Mari's implementation of Ptex face texture storage is targeted at the flexibility of face resizing and the rendering preview performance of the GPU. Unlike UV maps, the locality of faces in 3D space is not necessarily maintained in the Ptex texture storage. As a result of this implementation, when GPU operations require sampling across faces and adjacent faces reside in texture memory, preview artifacts may be observed.

For example, shaders that use a relatively large sample kernel may encounter some preview rendering artifacts when unexpected face textures are sampled.

Artifacts may also be observed when navigating a scene due to the blending of face textures in generated mip-mapped textures that are used to improve GPU performance. This may be more apparent when lots of small face textures are used.

Note: These artifacts only affect GPU accelerated previews, and do not appear in exported texture data files.

Ptex Modeling Requirements

Mari supports both quad and triangular face textures, which are exposed by the Ptex file format. All meshes use quad face textures, including mixed meshes; while wholly triangular meshes use triangular face textures. Quad textures are natively supported in graphics hardware. However, triangular textures are not supported in the current generation of graphics hardware. Mari's canvas preview of triangular textures uses the system described for GL displays at http://ptex.us/tritex.html.

Though Mari supports both types of face textures, Ptex dictates that the dimensions of a triangular face must be equal. Due to this restriction to triangular textures, if they are applied to polygons with an aspect ratio significantly different to one, the texture appears blurred or smeared. When triangular textures are stored in packed format, GPU renderers can only display half of the texels, resulting in Mari filling in the other half of the texture with data that it is not representable in the display.

Note: Some renderers, such as RenderMan, read all texels and, thus, can render triangular texels.

Tip: For more information regarding packed format and triangular textures, please refer to http://ptex.us/tritex.html.

For low-frequency images, the even and odd texels are very similar in color, and the Ptex packing of triangular textures suggests that a simple mirror of texels can be used to fill the other half of the texture when exporting to Ptex. Mari implements this mirroring. For high-frequency images, however, this mirroring is incorrect and visual artifacts appear in software renderers using the exported Ptex file.

With the restrictions of triangular textures, we highly recommend not modeling with only triangular faces. Quad face textures do not have the same triangular face equality restrictions and are a preferable option for use in Mari.

Creating a New Project with OBJ and Ptex Files

Multiple files can be used to create a project in the Ptex workflow. If this is the case, it is recorded in the dialog's history as a set of files instead of only one file.

> **Note:** These instructions primarily discuss the steps for creating a new project that are specific to OBJ and FBX files. For more information on creating a new project, from start to finish, refer to the *Mari Getting Starts Guide*, in the *Managing Projects* chapter.

To create a new project, using either one file or a set of files, follow the steps below:

1. As with most commands in Mari, you can select the **New Project** option a few different ways, pick any one of the options below.
   - from the **File** menu, select **New**,
   - click on the toolbar,
   - click the **New** button,
   - right-click in the **Projects** tab and select **New** from the dropdown menu, or
   - press the **Ctrl**/**Cmd**+**N** shortcut key.

   The **New Project** dialog box displays.
2. Enter a name for your project and the location of the geometry you want to work on.
3. You have the option to load either **.ptx** files or **.obj** files. If you choose to load an **.obj** file, the **Mesh Options** panel displays with options for setting the **Mapping Scheme** for the project, using **Selection Groups**, and working with **Multiple Geometries Per Object**.

Mesh Option	Description
Mapping Scheme	If there is a UV map available, and you would like to use it, select **UV if available, Ptex otherwise**. If you prefer to use Ptex, select **Force Ptex**. This option loads the selected geometry and replaces the existing UV map with a Ptex map.
Selection Groups	If you want to select a specific object group in the geometry while you

Mesh Option	Description
	work, this option allows you to set up selection groups using **Create From Face Groups** during project creation.
Multiple Geometries Per Object	If your **.obj** consists of multiple geometries, you have the option to **Merge Geometries Into One**, which is the default behavior, or **Create Separate Geometries** (Ptex Only). If you select **Merge Geometries Into One**, Mari combines the individual geometries into a single geometry, while selecting **Create Separate Geometries** (Ptex Only) creates individual geometries.

New Project dialog: **.obj** mesh options.

New Project dialog: **.ptx** mesh options.

4. If you choose to load a **.ptx** file, the **Mesh Options** only provide the option for selecting **Separate Meshes** or **Single Mesh** for the **Merge Type** field.

   If you are importing more than one Ptex file, this option allows you to specify whether you prefer for each separate file to constitute a separate object or separate geometries within a single object.

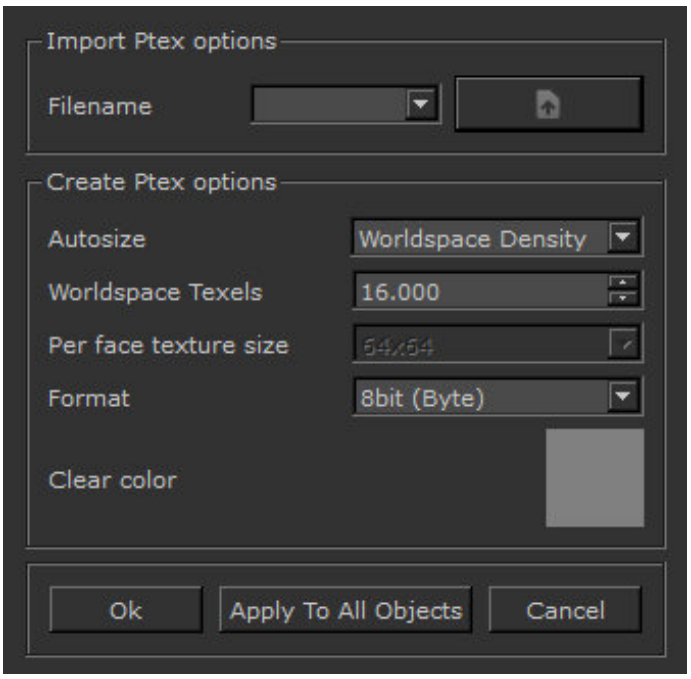
Note: Even if you select **Create Separate Geometries (Ptex Only)**, Mari only displays a single object on the canvas. However, when you export a channel you have the option to export individual **.ptx** files for each geometry.

For Ptex projects, the channel setup and texture import section of the **New Project** dialog can be ignored. A single channel named **Ptex** is created by default, you can create additional channels once the project has loaded.

5. Continue setting up your new project as you would with any other file type.

Setting the Ptex Face Texture Options

If you use an **.obj** file to create a new project, and the file does not have a UV map assigned (or you selected **Force Ptex** as the **Mapping Scheme**), the **Ptex Import or Creation** dialog box displays once the project is created.



You can also manage Ptex face textures by accessing the **Preferences** menu.

Importing Ptex Options

If you have a topologically matching **.ptx** file for the **.obj**, you can import it using the **Import Ptex options** field. However, if you don't have matching texture sizes, it can be resized upon import from the **Ptex Texture Import**
dialog.

Creating Ptex Options

If you don't have a matching Ptex file for the **.obj**:

1. Under **Create Ptex options**, you can select constant colored textures for each face, whose resolutions are based on either:
   - **Uniform Face Size** - for face sizes that are equal and determined by the resolution set in the **Per face texture size** dropdown menu, or
   - **Worldspace Density** - for a given number of texels per unit of world space, where small faces have smaller
     textures and large faces have larger textures. You can set the number of texels in the **Worldspace Texels** field.

> **Note:** Please note that texture quality may be affected when painting across adjacent faces if the faces are spatially different in size.

2. Set the bit depth of the Ptex image data by clicking on the **Format** dropdown menu. You can set this to **Byte**, **Half** or **Float**.
3. Click **OK**.

Managing Ptex Faces

Once you have created the project and set the face texture options, a toolbar displays with a number of options for managing Ptex faces within your project.

Note: When managing Ptex faces, please be aware that all image channels in your project share the same face texture resolutions.

With the **Ptex** toolbar you can:

- - Double the selected face resolution.
- - Halve the selected face resolution.
- - Set the world space face resolution for the selected face.
- - Set the face resolution for the selected face.

With the **Selection** toolbar, navigate to **Fill** to:

- - Fill the selected faces with the foreground color.
- - Fill the selected faces with the background color.

Note: When adjusting the resolution of Ptex models, quad faces are limited to resolutions between 1x1 and 8kx8k, while triangular faces are limited to resolutions between 2x2 and 8kx8k.

You may notice a slight pause after adjusting the resolution of a selected face. This happens while Mari regenerates the texture's proxy on the CPU.

Tip: These options are also available in the menu bar under **Ptex** and **Selection.**

Within **Preferences**, it is possible to set the **Minimum Size** for face textures under **Ptex** > **Face Textures** | **Minimum Size**. The default quadrangular texture is 1 and the default triangular texture is 2. This represents the absolute minimum face texture size that Mari can restrict sizing operations to.

Exporting a Ptex Channel or Layer

1. To export a **.ptx** channel, select:
   - **Channels** > **Export** > **Export Current Channel**, or

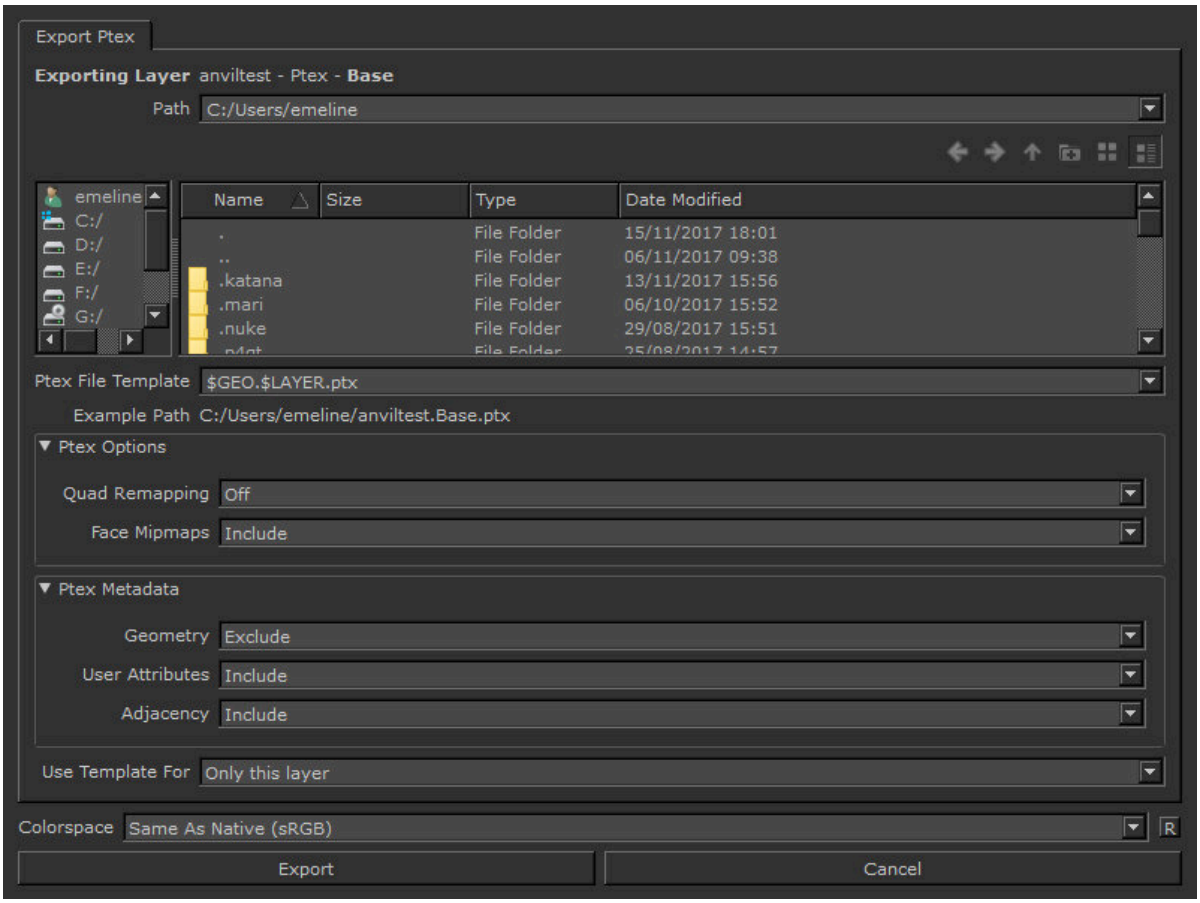
- right-click on a channel and select **Export** > **Export Current Channel**.

> **Note:** If you want to export more than the current channel, select **Export** > **Export All Channels** or **Export** > **Export Everything** from the **Channels** menu or right-click menu.
>
> Alternatively, you can also export channels in a flattened format to save space when exporting. To export flattened channels, navigate to **Export Flattened** in the **Channels** menu or right-click to find the same three options for flattened channels.

2. To export a **.ptx** layer, select layers and navigate to:
   - **Layers** > **Export** > **Export Selected Layers**, or
   - right-click on a layer and select **Export** > **Export Selected Layers**.

> **Note:** If you want to export more than the selected layers, select **Export** > **Export All Layers** from the **Layers** menu or right-click menu.
>
> Alternatively, you can also export layers in a flattened format to save space when exporting. To export flattened layers, navigate to **Export Flattened** in the **Layers** menu or right-click to find the same two options for flattened layers.

The **Export** dialog displays.



3. Enter the Ptex export path in the **Path** field or manually browse to the location.
4. Set the **Ptex File Template** for Mari to use when generating the Ptex filename. You can edit this line manually. As you edit, the **Example Path** updates with an example **.ptx** filename. You can use any combination of the following variables (which Mari replaces with the appropriate values when it exports the files):
   - **$ENTITY** - exports a single **.ptx** file, where the object name is part of the filename.
   - **$CHANNEL** - exports a single **.ptx** file, where the channel name is part of the filename.
   - **$GEO** - exports multiple **.ptx** files, where the individual geometry names is part of the filename
   - **$LAYER** - exports multiple **.ptx** files, where the layer name is part of the filename.
   - **$COLORSPACE** - exports a single **.ptx** file, where the colorspace used is part of the filename.

> **Experiment:** Experiment with the different **Ptex File Template** variables, and see how the **Example Path** changes to show how Mari generates exported filenames.

5. In the **Ptex Options** panel set whether Mari includes or excludes **Face Mipmaps**.
6. In the **Ptex Metadata** panel set whether Mari includes or excludes the following information with the file when exporting:

- **Geometry**
- **User Attributes**
- **Adjacency**

7. Set the **Use Template For** field to determine if the template above is used for **Only this layer** or **Everything**.
8. For the **Colorspace** field, this is the colorspace to which the output values are converted. **Automatic** is determined by the file name, size, and type of data in file.
9. Click **Export**.

   Mari exports your textures to disk in **.ptx** format.

> **Note:**
> As the options above only export channels one dialog at a time, to export multiple channels at once, see the *Export Manager Dialog* section in the *Mari Reference Guide*.

Import a Ptex Channel or Layer

1. To import a **.ptx** file to a channel, select:
   - **Channels** > **Import** > **Import into Current Channel**, or
   - right-click on a channel and select **Import** > **Current Channel**.

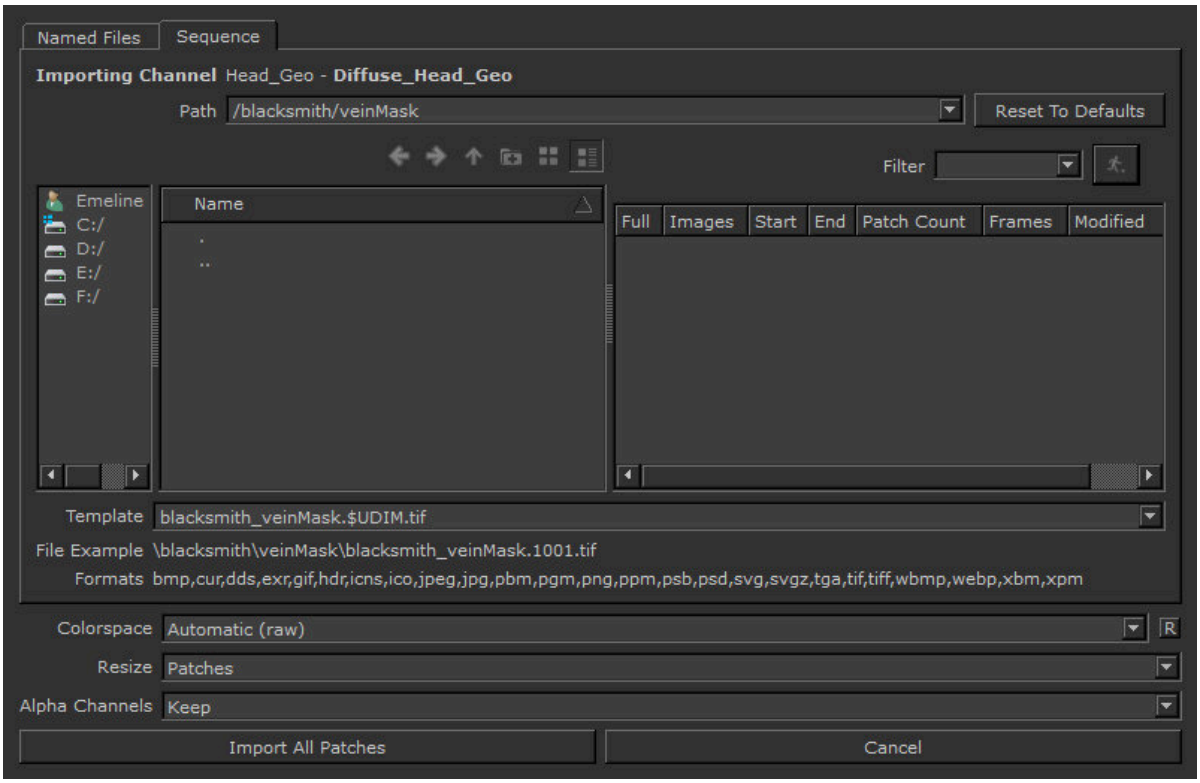
> **Note:** If you want to import into a new channel instead of the current layer stack, select **Import** > **Import into New Channel** from the **Channels** menu or right-click menu.

2. To import a **.ptx** layer, select layers to import into and navigate to:
   - **Layers** > **Import** > **Import into Current Layer**, or
   - right-click on a layer and select **Import** > **Import into Current Layer.**

> **Note:** If you want to import into the layer stack as a new layer, select **Import** > **Import into Current Layer** from the **Layers** menu or right-click menu.

Note: Ensure that you don't import textures into a layer you want to preserve current textures on, as importing into a layer updated the previous paint.

The **Import** dialog displays.



3. Enter the Ptex import path in the **Path** field or manually browse to the location.

Note: The Walk button is toggled off by default. When enabled, Mari looks in all folders of the selected location (refer to the **Path** field in the **Import** dialog) and displays all folders and files in the right pane.

4. You can manually select the Ptex files to import or you can set the **Ptex File Template** to automatically search for Ptex files that match the object, channel, or geometry variables. You can edit this line manually. As you edit, the **Example Path** updates with an example **.ptx** filename. You can use any combination of the following variables:
   - **$ENTITY** - imports a **.ptx** file, where the object name is part of the filename.
   - **$CHANNEL** - imports a **.ptx** file, where the channel name is part of the filename.
   - **$GEO** - imports **.ptx** files, where the individual geometry names are part of the filename.
   - **$LAYER** - imports **.ptx** files, where the layer name is part of the filename.
   - **$COLORSPACE** - imports a single **.ptx** file, where the colorspace used is part of the filename.

5. In the **Ptex Metadata** field you can set whether Mari should:
   - **Ignore** the geometry metadata, or
   - **Import as New Version**.
6. For the **Colorspace** field, this is the colorspace to which the input values are converted. **Automatic** is determined by the file name, size, and type of data in file.
7. Click **Import**.

   Mari imports the images.

> **Note:** If you are attempting to import geometry with degenerate faces into a Ptex project, Mari removes these faces on import. A warning dialog displays notifying you of this occurrence.

> **Note:** For more information on importing or exporting channels or layers, see Channels and Layers.

To import a texture to a channel from multiple **.ptx** files, where multiple geometries are in an object, follow the steps below.

1. Select:
   - **Channels** > **Import** > **Import into Current Channel**,
   - **Channels** > **Import** > **Import into New Channel,** or
   - Right-click on a layer in the **Layers** palette and select **Import** > **Import into Current Layer**.

   The **Import** dialog displays.
2. Enter the **.ptx** file import path in the **Path** field or manually browse to the location.

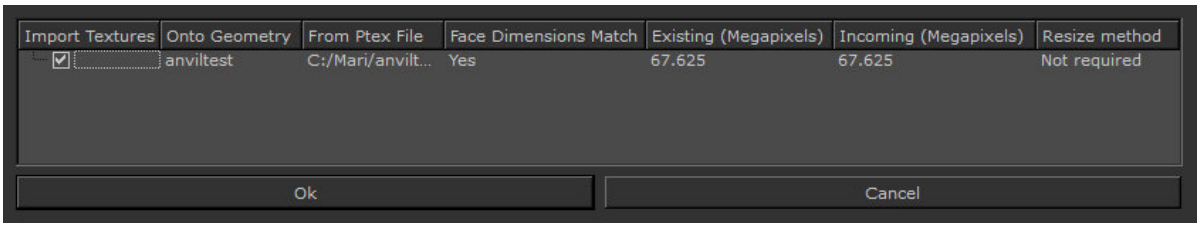
   The same **$ENTITY**, **$CHANNEL**, and **$GEO** variables are displayed, as described above, and displays an **Example Path** below the dropdown menu**.**
3. Click **All**.

   The **Ptex Texture Import: Face Texture Resize Options** dialog displays.

   If the face dimensions of the incoming import match, it displays the information, but no further action is required.

   If the face dimensions of the incoming import do not match, a **resize method** dropdown menu is listed. The options available are:
   - **Resize incoming faces to existing dimensions** - the imported texture sizes are changed to match those of the existing textures.
   - **Resize existing faces to incoming dimensions** - the existing textures are changed to match those of the incoming import.

- **Largest dimensions available per face** - whichever texture is the largest, whether existing or incoming import, this is the texture size used. This is the default option when resizing is necessary.



4. Click **Ok**.

   Mari imports the textures and loads them into the selected channel.

> **Note:** The resize options are applied per face of the geometry. The **Ptex Texture Import: Face Texture Resize Options** dialog is also displayed when importing new geometry versions from **.ptx** files.

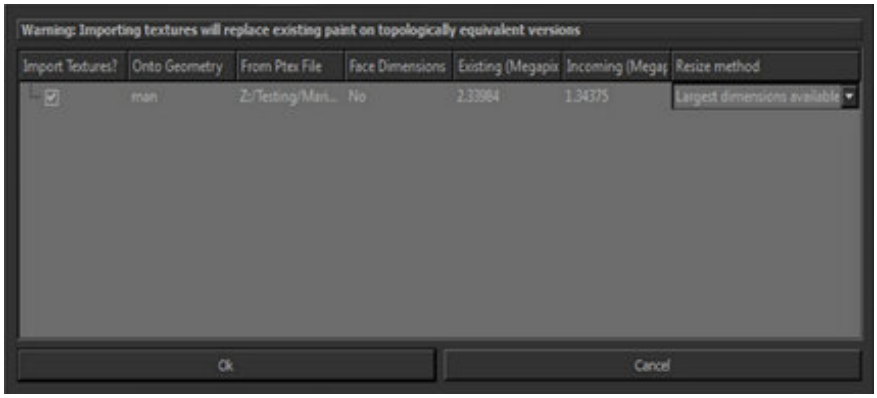
Painting Preservation with Multiple Ptex Geometries

Geometry versions from **.obj** files can be added to a Ptex project, provided all the versions are topologically equivalent, that is, they have the same number of faces with the same shape, so that the model does not change. This preserves the paint that has already been applied to the existing version.

Though **.obj** files can be added to Ptex projects with topologically equivalent versions - that is to say, versions that have the same number of faces, with the same shapes - there are caveats to the functionality:

- If the **.obj** file contains multiple geometry groups within it and the project has been created to treat these as separate geometries within a file, the **.obj** file can still be used as a version with fewer groups. These are matched and the paint preserved.
- If more than one geometry is topologically equivalent, then a match for their **.obj** file group names are used to attempt to distinguish between them.
- If an **.obj** file with a previously unknown group is used as a version, a new Ptex texture space is allocated to these new geometries.
- If there are fewer geometry groups in the new version than in the existing geometry, the extra geometries are hidden.

Any topological changes to geometries added as a version, such as a change to the model shape or size, do not transfer existing paint. If a **.ptx** file is imported as an object version over **.obj** geometry, any paint that has been applied is overwritten by the incoming version. A warning message on the **Ptex Texture Import: Face Texture Resize Options** dialog alerts you to the risk of losing existing paint on topologically equivalent versions, as shown in the example below.



Alembic and FBX

An Alembic file can contain several entities, such as geometry (polygon meshes and subdivided surfaces), cameras, materials, and lights. Alembic files allow different groups of people to work on the same assets without losing any data.

Mari supports the use of Alembic files, especially when handling multiple geometries within a single Alembic (**.abc**) file. On project creation or import, Alembic files can be used in place of **.obj** or **.ptx** files, with their own mesh options and parameters for specifying how Mari treats the different entities. However, there are differences. Mari only allows geometry and cameras to be loaded or imported, and there are some considerations that must be made if you're used to working with **.obj** or **.ptx** files.

An FBX file can contain several entities, such as geometry (vertices and patches), materials, texture, and animation. As opposed to the Alembic format, which is a point cache format that bakes rigs and constraints to vertices, FBX files export these entities to FBX nodes, which gives you the flexibility to have the original data, and manipulate it as needed.

Alembic Support

Alembic allows Mari to import hierarchies of objects within a single file. Any Alembic file containing multiple objects as part of a hierarchy can have these objects merged on project creation or load. Cameras, also

packaged as part of the file, can be imported and treated as Mari would treat other projectors in the **Projectors** palette.

Currently, Alembic support in Mari includes:

- starting a new project with either multiple objects or several objects merged into one from the same **.abc** file
- optionally choosing to merge geometry as a single mesh or in such a way as to preserve the geometry and metadata
- working with object hierarchies in order to retain multiple child objects as part of a parent object, just as you would within your modeling package
- creating a project with, or loading in, projectors from an **.abc** file to store in the **Projectors** palette
- loading in animated geometry as a single **.abc** file as opposed with **.obj** files, which require a file for each frame of animation
- loading or importing face sets packaged with the **.abc** file to store in the **Selection Groups** palette
- Python bindings for Alembic project creation, object and camera loading, and import/export functionality. Refer to the Python HTML documentation within **Python** > **API** for further information.
- detecting whether to subdivide each Alembic geometry piece, based on subdivision flags set in the file.

Limitations to Mari's Support of Alembic

Mari does not support all the functionality that comes along with Alembic files. Some entities that the file format contains cannot be handled in Mari or are restricted.

Currently, the limitations of Mari's Alembic support include:

- loading or importing only geometry and cameras, not lights or materials, even if the **.abc** file contains them
- loading or importing only one **.abc** file, as one file can contain several objects, unlike **.obj** or **.ptx** files
- creating selection groups from geometry faces cannot be done in the same way as with **.obj** or **.ptx** files; instead face sets are imported from the **.abc** file and stored in the **Selection Groups** palette.
- adding new versions of child objects all at once by loading or importing the parent object cannot be done. Child objects' versions need to be updated individually.

FBX Support

Mari supports importing cameras, geometry, and animation from **.fbx** files, and loads **.fbx** and **.abc** files similarly. Additionally, the scene data from the file is presented in the **New Project** dialog in the same tree widget format. As with Alembic files, FBX GeoEntities are created from polymeshes and sub-divisions, projectors are created from imported cameras, and selection sets are created from face sets and assigned materials on a polymesh.

Currently, FBX support in Mari includes:

- creating GeoEntities from polymeshes and sub-divisions
- creating Mari selection groups out of FBX face sets and polymesh materials
- creating projectors from imported cameras
- selecting animation takes and applying the animation to a GeoEntity in Mari
- importing keyframe animation from selected geometry and cameras
- importing object hierarchies and transform animation
- loading FBX deformations (mesh skinning and blendshapes) onto a selected geometry
- Python bindings for FBX project creation, object and camera loading, animation take management, and import/export functionality. Refer to the Python HTML documentation within **Python** > **API** for further information.

Limitations to Mari's Support of FBX

Mari does not support all the functionality that comes along with FBX files. Some entities that the file format contains cannot be handled in Mari or are restricted.

Currently, the limitations of Mari's FBX support include:

- only loading polymesh and sub-division geometry, cameras, and any transform or group nodes that contain geometry. NURBS, patches, markers, locators, lights, and curves are not loaded into Mari.
- only loading the skeleton data for a mesh (that is to say, joints, bones, their transforms, and keyframe animation). The associated mesh is deformed using the skeleton. Mari does not read in higher level IK/FK solver systems.

Note: If you want to bring any modeler-specific solver animation data into Mari, you need to bake the IK/FK animation into the skeleton and then export the mesh and skeleton to Mari.

Alembic and FBX Modeling Requirements

With the ability to bring in multiple objects at once from an Alembic or FBX file, you need to make sure that each of the objects occupies separate sets of UDIMs. You can merge objects together, which implies that the merged objects all occupy the same shading context. Without each object having its own set of UDIMs,

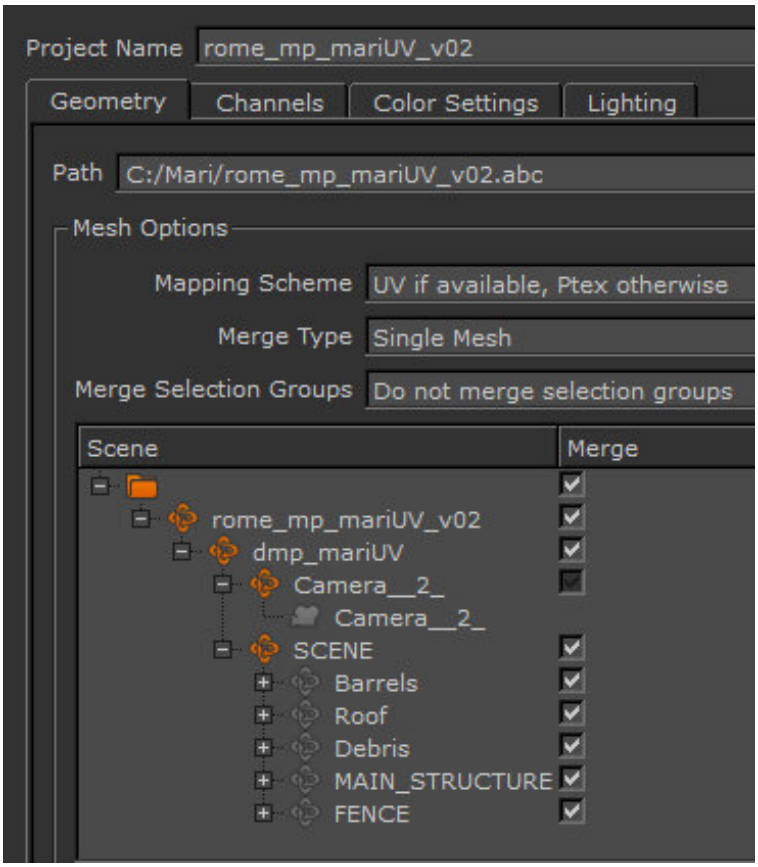
painting on one object causes paint to be projected on other objects that have been merged. Separating the UVs is done in the modeling package before the model is exported to Mari. Separating the UVs cannot be done in Mari, so it's important you ensure this is done prior to export. When modeling with FBX, please note that patches cannot be touching the edge of the UV, else it counts the patch as two separate patches.

For more information regarding UDIMs, please refer to Using UDIM Values.

Objects and Object Hierarchies

Alembic and FBX files can contain many objects, both on their own or as part of an object hierarchy. In the object hierarchy, there are a combination of transform objects, polymeshes, and subdivided geometries. All of these can be loaded or imported into Mari. These are treated, more or less, as multiple **.obj** objects are treated on project creation or import.

When working with Alembic or FBX files, the first place you encounter these objects and geometries is the **New Project** dialog. Once you have input an **.abc** or **.fbx** file into the **Path** field, the objects and cameras that are contained in the file, are shown in the Object Hierarchy Tree in the dialog.



Due to the hierarchical object structure, child objects inherit properties, such as transforms, from the parent object. Mari thus must make the distinction between entities and GeoEntities. Entities represent a transform. A GeoEntity contains all the geometry information, such as faces, vertices, normals, and the like. A GeoEntity is used to represent geometry in Mari. Only GeoEntities are visible in the Object Hierarchy Tree.

The entire Alembic or FBX file is loaded by default unless you make a selection. If you make a selection, Mari assumes that only selected objects are to be loaded. If this is the case, simply having the **Merge** checkbox ticked does not tell Mari to load that object; it must be selected as well.

Warning: If you don't select individual objects in the Object Hierarchy Tree, the entire Alembic or FBX file is loaded. Depending on the size of the file, this could take longer than expected to load into Mari.

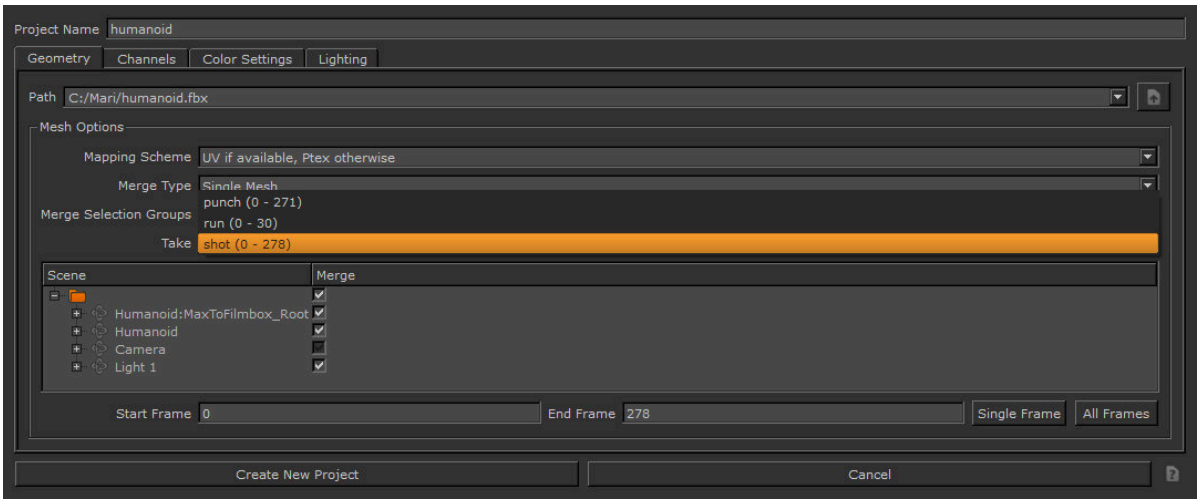
Animated Geometry

Alembic Animated Geometry

When bringing animated geometry into Mari from an Alembic file, Mari displays **start frame** and **end frame** controls in the **New Project** dialog. You can set the range you want to load or import or, if you want to bring in all frames or only a single frame, click on the **Single Frame** or **All Frames** buttons. Some **.abc** files, exported from older version of the Alembic exporter, do not expose their frame range when brought into Mari. In this case, Mari displays the notice "All frames from this archive will be loaded" and loads all frames.

FBX Animated Geometry

FBX stores animation on geometry in a similar manner to Alembic, and supports the concept of animation takes on a skeleton, which allows you to use the same skeleton structure on your mesh to create different animations. Each take is related to the same object, and can have different animation ranges as well. In Mari, this data is presented in the object loader widget in the **New Project** dialog. You can also select the specific takes you want from the **Take** dropdown under **Mesh Options**. However, within Mari, the FBX concept of an animation take is only applicable during geometry load.



Warning: Once the geometry is imported into Mari, it has the selected animation take bound to it. You won't be able to access any other animation takes or change the animation on the fly within Mari.

Deformations

Vertex Cache Deformations

For FBX files with heavy deformations to the mesh, it may be more effective to cache the deformations to disk. In a big production pipeline, this gives the artists the flexibility to pass the cache alongside the meshes down the pipeline, and to view the static meshes as well as to bring in the animation from disk. When you export this mesh and vertex cache to FBX, Mari loads in the mesh, locates the cache, and loads in the geometry with cached animation.

Skin Deformations

In cases where the FBX file contains the skeleton deformation of a mesh, where the geometry is rigged by adding a hierarchy, this is bound to the mesh, and keyframes are animated to deform the mesh, you need to follow certain binding methods to ensure that the **.fbx** file is loaded into Mari correctly. The three binding methods you can use are:

- Linear/classic binding - the default binding method, where the position of a vertex is determined by linear interpolation of the weights of the surrounding bones.
- Dual quaternion binding - a binding method that provides an improved solution to skinning geometry, where the volumes are preserved in case of extreme stretching movements.
- Blended binding - a binding method that mixes both linear and dual quaternion methods through the use of a blend weight.

Using one of these methods, both the mesh and the associated skeleton is exported to the **.fbx** file and, when imported into Mari, this information is read and loaded into the final mesh with the deformation.

Note: Mari does not display the skeleton; only the deformed mesh.

Cluster Deformations

Most modeling packages allow you to define extra weights on selected sets of vertices for a mesh, called a cluster deformation. Cluster deformations are supported by FBX at the point of export. Adding cluster deformations can work with skinned meshes, but you need to ensure that the order of input and output is

correctly set up before exporting the scene to an **.fbx** file. Once the file is loaded into Mari, the mesh is loaded, the associated clusters are gathered, and the weights are applied to the vertices, before the deformed mesh is then loaded into the project.

Blendshape Deformations

Most modeling packages allow you to create duplicates of original geometry in order to blend between the original and duplicate using a weight. This allows you to achieve in-between poses, and is especially useful for altering a character's facial expressions in a production environment. You can then keyframe animate the blendshapes and export them to FBX. One the file is loaded into Mari, the mesh is loaded, the associated blendshapes are gathered, and the weights are applied to load in the final blended geometry.

Creating a New Project with Alembic and FBX Files

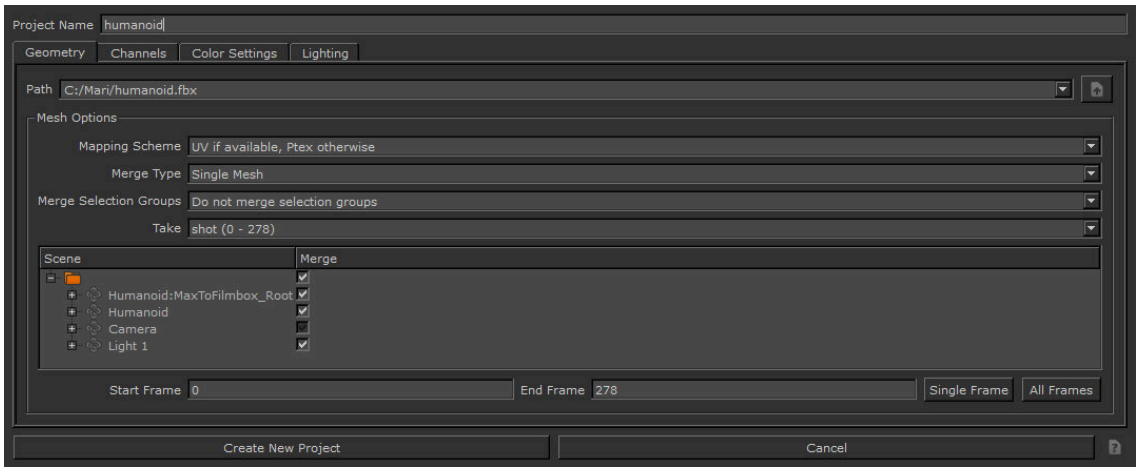
Note: These instructions primarily discuss the steps for creating a new project that are specific to Alembic and FBX files. For more information on creating a new project, from start to finish, refer to the *Mari Getting Starts Guide*, in the *Managing Projects* chapter.

To create a new project, using an Alembic or FBX file, follow the steps below:

1. Start Mari.
2. As with most commands in Mari, you can select the **New Project** option a few different ways, pick any one of the options below.
   - from the **File** menu, select **New**,
   - click on the toolbar,
   - click the **New** button,
   - right-click in the **Projects** tab and select **New** from the dropdown menu, or
   - press the **Ctrl**/**Cmd**+**N** shortcut key.

   The **New Project** dialog box displays.
3. Enter a name for your project and the location of the **.abc** or **.fbx** file you want to import from.

4. Once you load an **.abc** or **.fbx** file, the **Mesh Options** panel displays with options for setting the **Mapping Scheme**, **Merge Type,** and **Merge Selection Groups**. The Object Hierarchy Tree also displays beneath these options.



Mesh Option	Description
Mapping Scheme	If there is a UV map available, and you would like to use it, select **UV if available, Ptex otherwise**. If you prefer to use Ptex, select **Force Ptex**. This option loads the selected geometry (GeoEntity) and replaces the existing UV map with a Ptex map.
Merge Type	If you want to merge any of the objects (Entities or GeoEntities) on project creation, the merge type controls how Mari handles the merge operation. When merging as a **Single Mesh**, all the various geometries are merged into one geometry inside of a GeoEntity, and any object data for the merged objects is lost. When merging as **Just Merge Nodes**, all the merged objects are merged into one geometry inside a GeoEntity, but they keep their metadata and object data.
Merge Selection Groups	You can choose to **Merge selection groups with the same name** or choose the **Do not merge selection groups** option. For more information on selection groups, see Selection Groups.
Take	For FBX only - animation takes exported with the geometry can be selected in the dropdown menu. Takes can only be chosen at the time of project creation/object load and cannot be changed once it has been set. In cases where there is only one take from the file, the default is selected

Mesh Option	Description
	in the mesh options.
Object Hierarchy Tree	Displays all the geometry and meshes, cameras, transforms, and groups from the **.abc** or **.fbx** file.

Note: If the Object Hierarchy Tree seems to be taking too long to display objects from an **.fbx** file, ensure that the file was saved as binary FBX rather than ASCII FBX.

5. Click on the objects in the Object Hierarchy Tree to select which geometries you want to load into the project on project creation. You can also select any cameras that you want to load at the same time.
6. If you want to merge any of the objects, toggle the **Merge** checkbox to select which objects are merged together. For more information on how to merge objects, refer to Adding and Merging Alembic or FBX Geometry.
7. If the objects are animated, select the start and end frames for a specific selection, or click **Single Frame** or **All Frames**. If these options don't appear, you may be given a message that "All frames from this archive will be loaded."
8. Set a **Frame Offset** to specify the start position for loading the objects on the Mari timeline. The **Frame Offset** is only available when loading **.abc** files.

   For example, if you want to import frames 1005 to 1010, with an **.obj** file, you would select files xxx.1005.obj, which would populate the **start frame** field with 1005 and **end frame** field with 1010.

   Whereas with an **.abc** file, you would select the file and that would populate the **start frame** field with 1 and the **end frame** box with 5.

   In order to load the frames correctly into Mari's timeline, starting at 1005, you need to set the offset to 1004, so that it loads the geometry into Mari's timeline at 1005 to 1010. Failure to do this causes Mari to load the geometries at Mari's frame 1 to 5.
9. Continue setting up your new project as you would with any other file type.

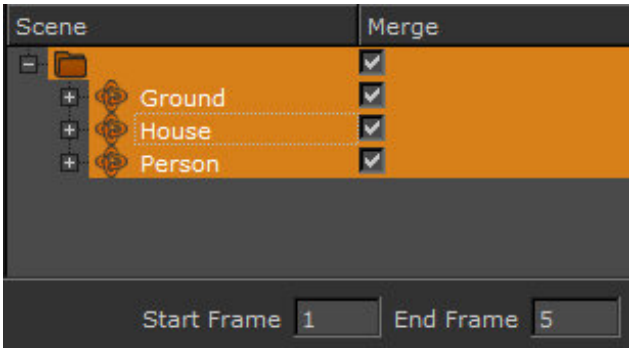
Adding and Merging Alembic or FBX Geometry

Each of the objects or entities, shown in the Object Hierarchy Tree, has a checkbox next to it for selecting which objects are merged. The **Merge Type** in the **New Project** dialog's **Mesh Options** determines how the objects are handled on merge: as a **Single Mesh** or **Just Merge Nodes**. Toggling the **Merge** checkbox of a parent object ticks the checkboxes of all the child objects as well, and signals that the parent and child objects should be merged into one GeoEntity, or object in Mari. You can also select a root object and toggle the **Merge** checkbox to determine if any other objects are merged into the root. When a camera appears in the Object Hierarchy Tree, the camera is shown with the **Merge** checkbox ticked by default. Cameras are not stored in a hierarchy in Mari, thus the state of the checkbox cannot be changed.

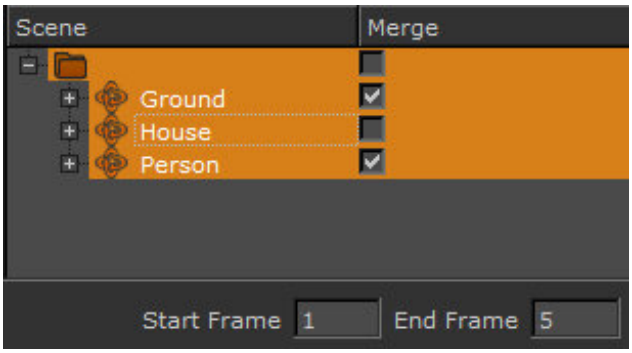
Merging objects can happen in a number of ways, so that different parts of the geometry (child objects) are merged with the parent object to create one GeoEntity, several child objects are merged together but not merged with the parent object, or several parent objects in an Alembic or FBX file are merged together so there are fewer geometries being created on project creation.

In an example Alembic file, where there are three parent objects (Ground, House, and Person), the following are examples of common ways to merge objects:

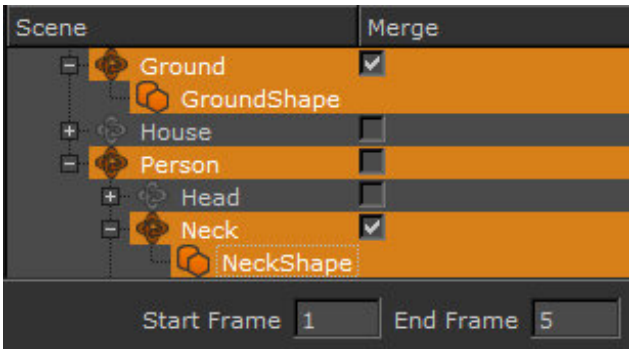
- Everything is merged together - select the merge checkbox for each of the selected parent objects and the root folder.



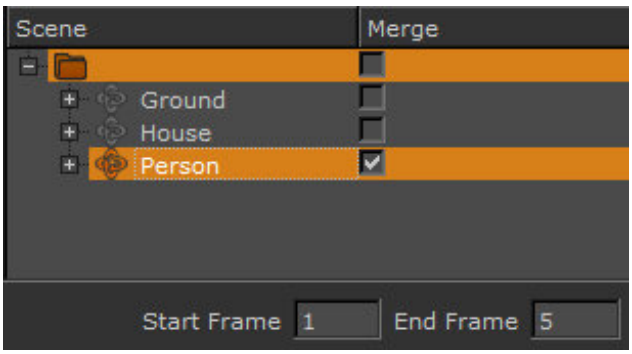
- Specific selected objects are merged together - selected objects with the **Merge** checkbox ticked are merged together. Any selected objects that do not have the merge checkbox ticked are loaded as separate objects.



- Selected objects aren't merged but selected cameras are merged by default - none of the selected objects are merged together but the selected camera has the **Merge** checkbox ticked by default and is disabled so that it cannot be changed.
- Selected objects are partially merged - selected child objects are merged into the selected parent object but the parent object is not merged into the other selected parent objects.



- Selected frame ranges are merged - selected objects with a frame range specified are merged into a single geometry.



To merge Alembic or FBX geometry on project creation, refer to the steps in Creating a New Project with Alembic and FBX Files for more information. If you have already created a project and want to load in merged Alembic or FBX geometry, follow the steps below.

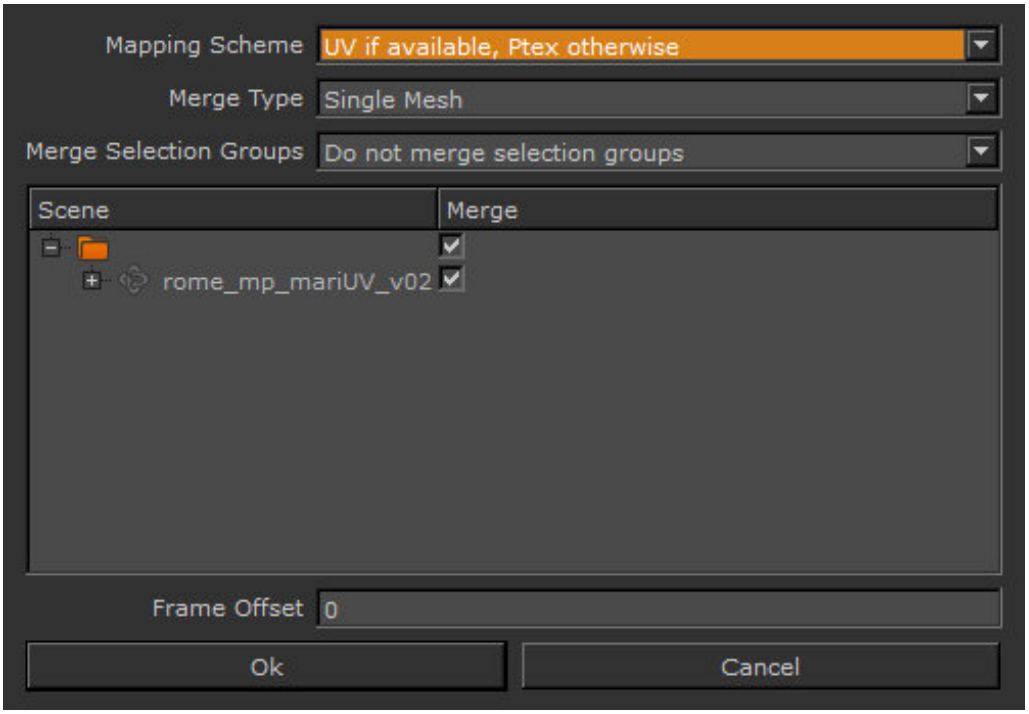
Merging Objects on Object Import

To merge objects that you are importing into an existing Mari project:

1. Open the **Objects** palette and click the icon, or select **Objects** > **Add Object**.

   The **Add Objects** dialog appears.
2. Select the Alembic or FBX file with the objects that you want to import into the project, and click **Open**.

   The **Add Object Settings** dialog appears.



3. Set the **Mapping Scheme**, **Merge Type**, and **Take** fields. These are the same as the fields in the **New Project** dialog, described in Creating a New Project with Alembic and FBX Files.
4. Select the objects that you want to import into the project, and click the **Merge** checkbox for each of the objects that should be merged together on import.
5. If all the frames are loaded into the project, the dialog specifies this, and does not show a start and end frame range.
6. You can also set a **Frame Offset** for the start position for loading the objects.

   For example, if you want to import frames 1005 to 1010, with an **.obj** file, you would select files xxx.1005.obj, which would populate the **start frame** field with 1005 and **end frame** field with 1010.

   Whereas with an **.abc** or **.fbx** file, you would select the file and that would populate the **start frame** field with 1 and the **end frame** box with 5.

In order to load the frames correctly into Mari's timeline, starting at 1005, you need to set the offset to 1004, so that it loads the geometry into Mari's timeline at 1005 to 1010. Failure to do this causes Mari to load the geometries at Mari's frame 1 to 5.

7. Click **Ok** to import the objects or **Cancel** to abandon the import and close the dialog.

Importing Objects

At any time after project creation, you can choose to import objects into the project. These objects can be GeoEntities from the same or another **.abc** or **.fbx** file, from an **.obj** file, or even from a **.ptx** file. To import more objects into the project, add objects as you would through the **Objects** palette or the **Objects** menu. For more information, see Adding Objects

Importing New Versions

As with objects, you can add object versions from an **.abc** or **.fbx** file to the **Objects** palette any time after project creation. The object versions must be of the same file type and merge structure; thus, if you already have an object from an **.abc** file and you loaded it into Mari with merged geometry, the new version should ideally be an **.abc** file with the same merged geometry. Just the same, if the objects were loaded into Mari with unmerged geometry, the versioned file must also be unmerged.

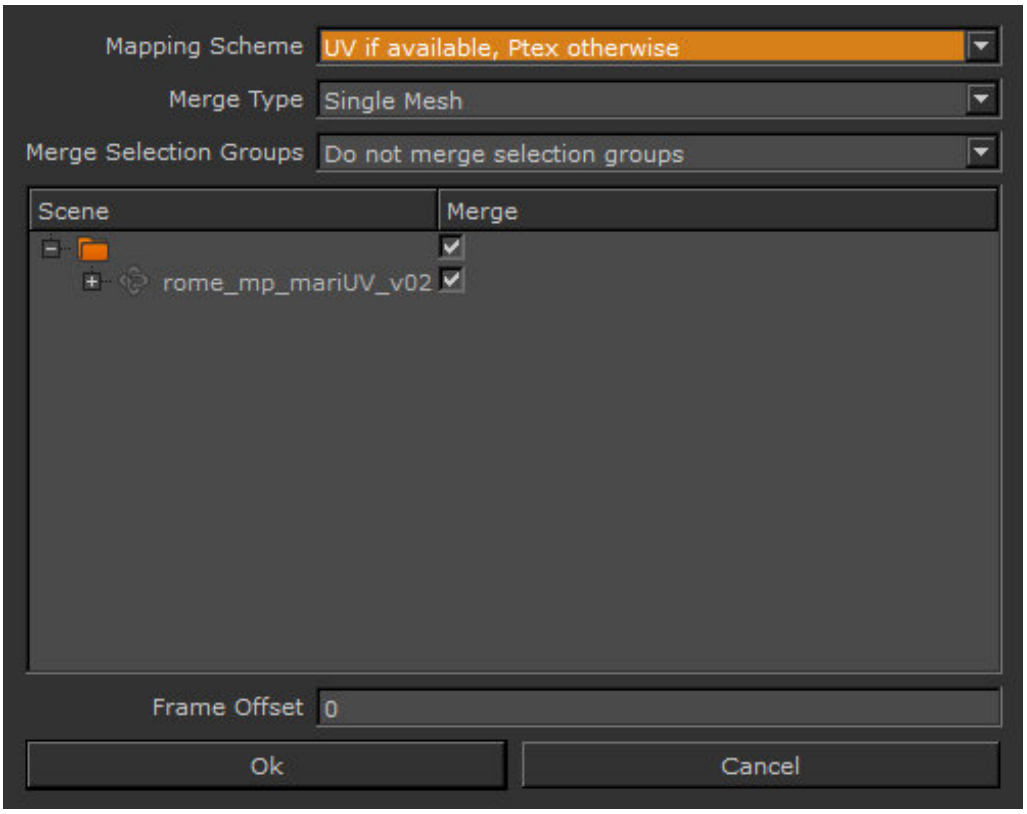
To add new objects to a project, refer to the steps in Adding and Merging Alembic or FBX Geometry for more information. If you have already added objects to your project and want to load in new versions of them, follow the steps below:

1. Open the **Objects** palette and right-click on the object for which you want to load a new version.

   The **Add Version** dialog appears. This looks like a Mari-standard file browser.

2. Select an object from the file browser or type in the file path, then click **Ok**.

   The **Add Version Settings** dialog appears.



3. Set the **Mapping Scheme**, **Merge Type**, and **Take** options for the new object version. For more information about these controls, refer to the steps in Creating a New Project with Alembic and FBX Files.

> **Note:** When choosing whether or not to merge geometries, bear in mind that new versions must have the same merge structure as the original object version. For instance, if you already have an object from an **.abc** file and you loaded it into Mari with merged geometry, the new version must also be an **.abc** file with the same merged geometry.

4. If the objects are animated, select the start and end frames for a specific selection, or click **Single Frame** or **All Frames**. If these options don't appear, you may be given a message that "All frames from this archive will be loaded."
5. Set a **Frame Offset** to specify the start position for loading the objects on the Mari timeline.
6. Click **Ok** and the new object version is loaded into the **Objects** palette.

> **Note:** Mari preserves selection groups between object versions. See Adding and Removing Object Versions.

Importing Cameras

As with objects and versions, you can add cameras from an **.abc** or **.fbx** file to the **Projectors** palette and, similarly to the **Objects** palette, you can load cameras from different file formats. For this reason, you can have a mix of **.fbx** and **.abc** cameras in the **Projectors** palette.

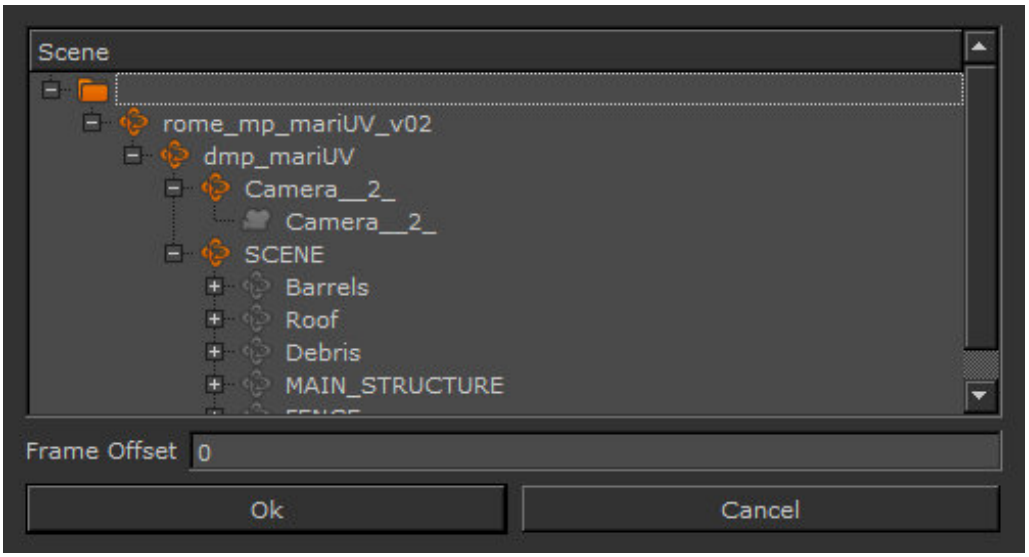
To import Alembic or FBX cameras on project creation, refer to the steps in Creating a New Project with Alembic and FBX Files for more information. If you have already created a project and want to load in FBX cameras, follow the steps below.

To import cameras from an Alembic or FBX file into an existing Mari project:

1. Open the **Projectors** palette and click the icon, or select **Camera** > **Load Camera**.

   The **Load Camera** dialog appears.
2. Select the Alembic or FBX file with the cameras that you want to import into the project, and click **Open**.

   If you are importing cameras from an Alembic file, the **Add Camera Settings** dialog appears. Otherwise, the **Select Entities** dialog appears.



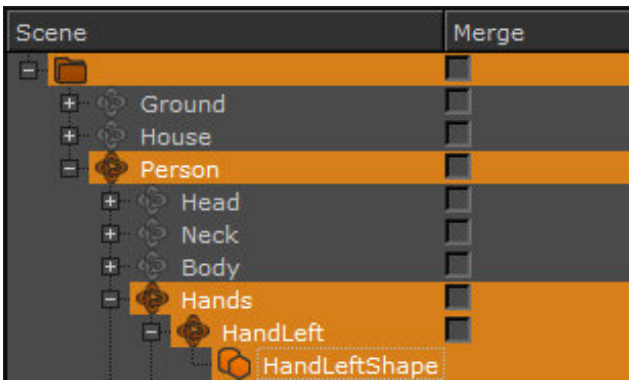
   The GeoEntities, transforms, and cameras in the file are all shown in the object hierarchy tree, but only the cameras can be loaded from this dialog. Note that selecting a geometry disables it in the object tree so that it cannot be selected again.
3. Select the cameras that you want to import into the project.
4. If the objects are animated, select the start and end frames for a specific selection, or click **Single Frame** or **All Frames**. If these options don't appear, you may be given a message that "All frames from this archive will be loaded."
5. Set a **Frame Offset** to specify the start position for loading the objects on the Mari timeline.

6. Click **Ok** to load the cameras into the **Projectors** palette.

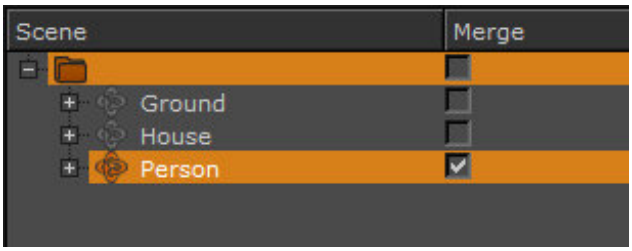
Naming of Objects

Objects are imported with the name of the hierarchy, with **/** replaced by _ if they are single unmerged geometries. If the child geometries are merged beneath a parent geometry, the naming changes so that the object name is the same as the parent geometry. In cases where some of the child geometries are merged while others are not, the unmerged child geometries take the name from the parent geometry/shape, and the merged child geometries take the name of the parent geometry.

For example, for the following village example Alembic file:



- none of the selected geometries are merged, so the imported object is named **Person_Hands_HandLeft_HandLeftShape** because the hierarchy path was Person/Hands/HandLeft/HandLeftShape when importing.



- all of the child geometries are merged together under the Person parent geometry, so the imported object is named **Person.**

- some of the child geometries under the Person parent geometry are merged but not all of them, so two objects are imported. The imported object from the merged geometry is named **Person_PersonShape** (it contains merged geometry from the face, hands, feet, and neck) and the imported object from the unmerged geometry is named **Person_Body_BodyShape** (it contains the unmerged body geometry).

Selection Groups

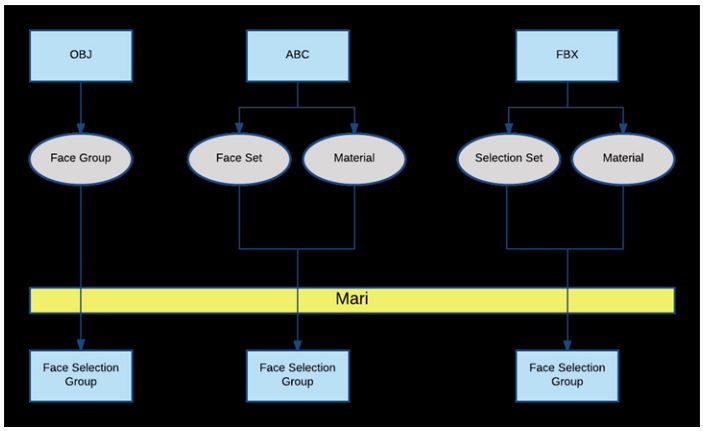
Mari automatically maps any face set or selection set found in an Alembic or FBX file to a selection group, when loading geometry. This makes your job easier, since you already have access to the various selection groups created by the modeling package and can paint straightaway.

Alembic Selection Groups

Unlike **.obj** files, selection groups are not created from faces on import. Instead, when loading an **.abc** file, Mari searches for whether it contains face sets exported from your modeling package. If it detects that there are face sets present, these are loaded on project creation or object import, as Mari's selection groups and added to the **Selection Groups** palette. Once you've created a project, you can create your own selection groups, if you wish.

FBX Selection Groups

Similarly to Alembic files, selection groups can be created from selection sets with **.fbx** files. Additionally, selection groups can also be created from FBX materials. These selection sets are automatically loaded on project creation or object import, and added to the **Selection Groups** palette. Once you've created a project, you can create your own selection groups, if you wish.


OBJ
ABC
FBX
Face Group
Face Set
Material
Selection Set
Material
Mari
Face Selection Group
Face Selection Group
Face Selection Group

Shading Networks

Shading networks in Mari consist of Channels, Layers, Shaders, and the Node Graph.

Channels

Channels hold layers stacks, filled with paint layers, procedurals, and adjustments in your project. For example, a project might have channels for diffuse color, displacement, or specularity, but each of those channels contain individual layers for paint, masks, and filters. Channels can then be used in shader inputs so you can adjust the amount of diffuse or specularity, among other things.

Mari supports multiple channels. New projects start with a single channel. You can add as many additional channels as you need. A single project can hold all the channel data required for the model - diffuse, dirt, specular, luminescence, displacement, and so on. Each object in a Mari project has its own set of channels. See Working with Objects for details on working with multiple objects.

Video: Watch Understanding Shaders and Channels for a brief overview about Channels. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Different Ways to Create Channels

When you create a channel, you set the color depth and patch size. You can change both of these after creation. If channels are resized after creation, Mari automatically resizes all the layers in the channel's layer stack. Channels are either color or scalar. See Color Data and Scalar Data for more information.

Channels can be created:

- using the settings for existing or recently-created channels as a template.
- in bulk, using Mari's inbuilt presets, optionally importing textures into the channel at the same time.
- from a preset size, color depth, and type.
- entirely custom and *ad hoc*.

- by copying a layer to a new channel (see Layers for details).
- by sharing a layer as a new channel (see Layers for details).

HDR Channels

The dynamic range of luminance in the real world is much greater than the range that is usually displayed on a screen. Low dynamic range (8-bit) color values can represent RGB levels only within the range of 0 (black) to 1 (white). In Mari, using 16-bit or 32-bit color values, you can clone from, or paint through, images outside of the 0 to 1 range.

For details on creating and working with channels that support HDR images, see Creating and Deleting Individual Channels.

Editing Paint on Your Channels

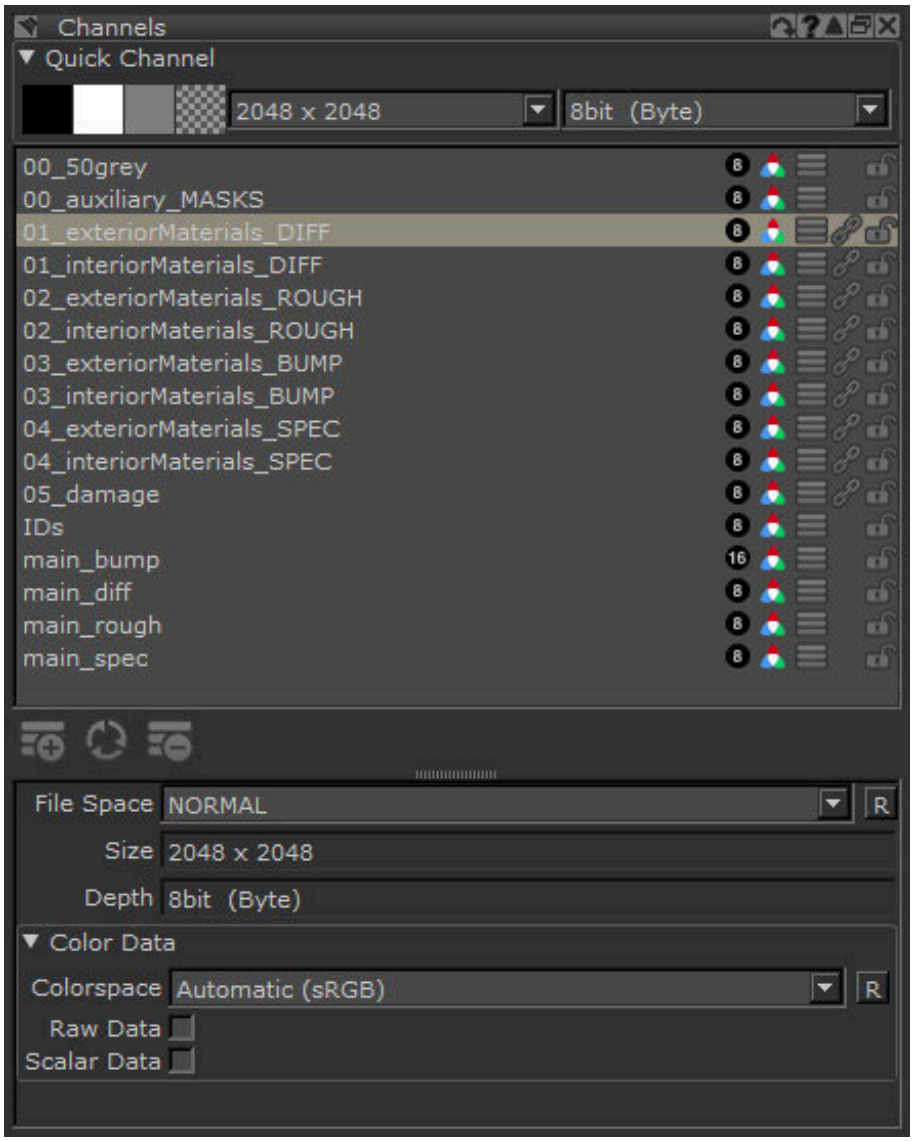
The paint in your channels can be edited or transformed, much the same as with patches. Where the transform functions differ is in how they change the paint on the model. Patches are very specific and only transform paint on the selected UV patch. Channels, on the other hand, can include paint across many patches and faces in their layer stacks. In this way, transforming a channel can affect a larger amount of paint across a greater part of a model, depending on how extensive the paint in the selected channel is.

Creating and Deleting Individual Channels

To create a channel:

1. Open the **Channels** palette by:
   - right-clicking on the toolbar, and selecting **Channels** from the dropdown menu, or
   - from the **View** menu, selecting **Palettes** > **Channels**.

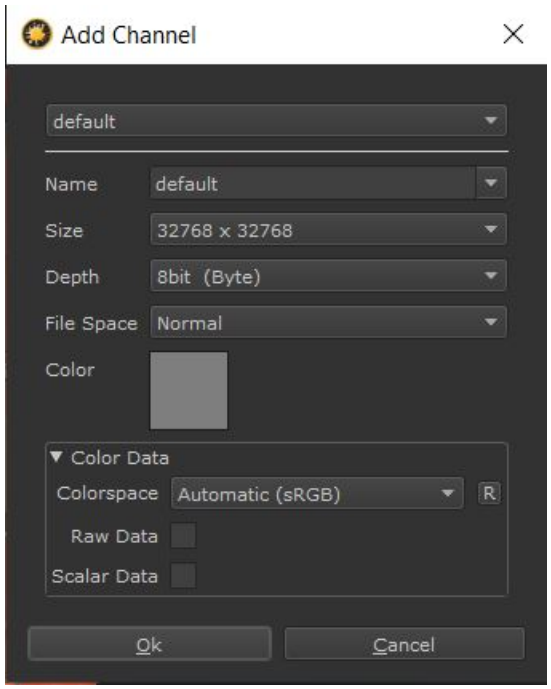
   The **Channels** palette displays, showing the channels in the current project, a quick channel creation bar, and channel attributes in the bottom pane.



Note: A selection of channel presets are created based on your shader model. Mari creates the **Principled BRDF - Non-Metallic-VFX** shader by default.

2. Add a single channel by:
   - right-clicking in the **Channels** palette, and selecting **Add Channel**,
   - clicking the button, or
   - from the **Channels** menu, selecting **Add Channel**.

   The **Add Channel** dialog box displays, allowing you to set the options for the new channel.



3. The dropdown list at the top shows the names of channels that you've created recently. Select one from the list.

   The **Name**, **Size**, **Depth**, **File Space**, and **Colorspace** populate automatically.

> **Tip:** If you have never created a channel, the dropdown at the top of the dialog does not appear, and the **Name** field is not populated automatically.

4. As needed, you can change:
   - **Name** - the name to identify the channel (which you can change later).
   - **Size** - the patch size for the new channel.
   - **Depth** - the color depth (8-, 16-, or 32-bit).
   - **File Space** - the type of file space used for vector painting: **Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**.
   - **Colorspace** - the colorspace to which the channel data or mask data is converted.
   - **Color** - the default color of the channel, on creation.

> **Note:** If you are working with a Ptex project, the patch **Size** option appears grayed out, as the **.ptx** format doesn't work with patches.

5. If you intend on cloning from, or painting through, either 16-bit or 32-bit images, the **Depth** needs to be set to either **16bit (Half)** or **32bit (Float)**, respectively. In addition, the color depth of the paint buffer needs to match the depth of the channel.

   Furthermore, if you do not want the range of 16-bit or 32-bit images clamped to values between 0 and 1, you need to uncheck the **Clamp** checkbox under the **Paint Buffer** settings in the **Painting** palette.
6. Set the **Colorspace** for the channel to which the channel data is stored. **Automatic** determines the colorspace based on the **Depth** (data type) and whether it is color data or scalar data.
7. Set the **Colorspace** for the channel's masks to which the mask data is stored. **Automatic** determines the colorspace based on the **Depth** (data type) and whether it is color data or scalar data.

> **Tip:** If you are not familiar with colorspaces, you may want to keep the default colorspace that Mari automatically selects for the **Colorspace** fields. For more details on the color and mask data, see the Channels Palette.

8. Select the initial **Color** for the channel. To select a different color, click on the swatch and choose a color with the color picker.
9. Click **Ok** to add the channel to the object.

   Mari creates your new channel, and displays it in the **Channels** palette.

> **Note:** If you are adding channels in a project with more than one object, instead of the **Ok** button, there are two buttons in the **Add Channel** dialog. You can opt to either add a channel to only the selected object or to all objects in the project, or cancel.

> **Note:** If you have already created a channel with a given name and attempt to create a new channel with the same name, Mari increments the name to differentiate between the original and new channel. For instance, if you already have a "diffuse" channel and create another channel named "diffuse", Mari automatically gives it the name "diffuse1".

To create a copy of a channel, press **Alt**+drag-and-drop the channel. Mari creates your new channel as a copy of the original - including the layers, but not all the paint in those layers.

To delete a channel:

- right-click on the channel in the **Channels** palette, and select **Remove Channel**,
- with the channel selected, click ,
- from the **Channels** menu, select **Remove Channel**, or
- with the channel selected, press **Delete** on the keyboard.

If you have other objects in the project that have a channel with the same name, you have the option of removing the channels from only the selected object, or all objects in the project.

Flattening Channels

Channels with large layer stacks or complex groups of layers, adjustments and procedurals can be flattened into a single base layer to save space and simplify channel organization. Flattening layers does not affect the final output, as the texture from the accumulated layers is not changed, only combined into a single layer.

If you are considering flattening the layer stack for a channel so that you can save space when exporting the channel, please bear in mind that it is possible to flatten channels on export. Flattening a channel on export does not affect the layer stack in your project, but combines the layers into a single base layer in the exported file.

To flatten the selected channel, right-click in the **Channels** palette and select **Flatten**, or from the **Channels** menu, select **Flatten**. A progress bar displays showing the time left until the channel has finished being flattened.

Note: To flatten channels containing multi-channel layers, you first need to either disconnect the required channels from the shader choosing the option to convert them to paintable layers, or flatten all the multi-channel layers within the Channel.

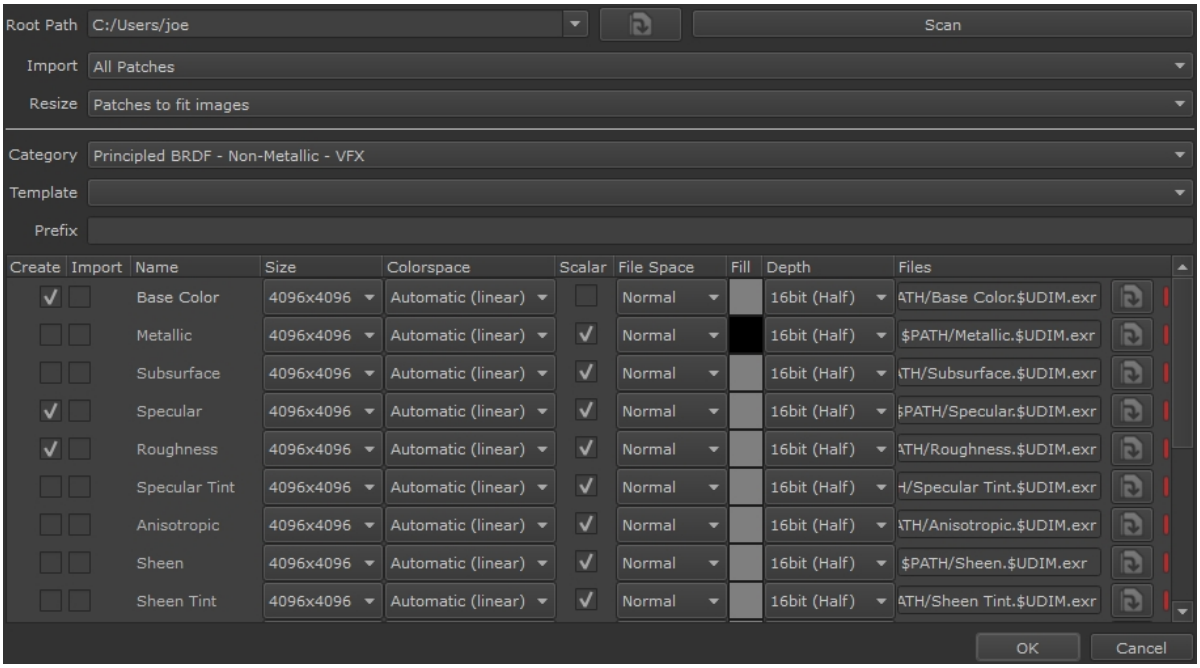
Creating Multiple Channels from Presets

Mari includes channel presets. These are sets of the “standard” channels required to paint various types of assets - for example, the channels required for a vehicle or a digital double. You can use these presets to create a group of channels at once. If you already have some existing textures for the new channels, you can automatically load these into the new channels as they are created.

Tip: You can customize the channel presets functions by writing your own channel preset and channel template files, to create custom presets and templates. For more information, refer to the steps below and the Extending Mari section.

1. To create multiple channels from presets:
   - right-click in the **Channels** palette, and select **Channel Presets**, or
   - from the **Channels** menu, select **Channel Presets**.

   The **Channel Presets** dialog opens.



2. The **Root Path** is the base directory for your Mari project. Mari looks here for textures to import when creating the channels. If the textures are in a different directory, change the **Root Path** to point to the appropriate directory, or use the button to display an **Open** dialog to find the right directory.
3. If you want to import existing textures into the channels as you create them, select whether to import **All Patches** or only the **Selected Patches**.
4. If you want to import existing textures, select in **Resize** how Mari should deal with patch/image size mismatches (for example, when trying to import a 2048x2048 image into a 1024x1024 patch). This can be either:
   - **Patches to fit images** - for example, Mari would automatically resize a 1024x1024 patch to 2048x2048 to fit the image.
   - **Images to fit patches** - for example, Mari would automatically resize a 2048x2048 image to 1024x1024 before importing it onto the patch.
5. Click **Category** and select a Mari channel preset.

6. Enter in a **Prefix** to assign a designation to the channels. This allows you to organize them according to categories in your list.

> **Note:** The list of channels at the bottom of the dialog box allows you to set whether you want to create the preset channels from scratch or from an imported file. Set the channel options as you would in the **Channel Presets Dialog**, see the Channel Presets Dialog.

7. Select the channels to create and import from the list by clicking in the boxes. You can either click to check the boxes individually, or right-click the list and select **Add Custom**, **Create All**, **Create None**, **Import Existing**, or **Import None**.
8. You can change any selected channel's **Size**, **Colorspace**, **Scalar** state, **File Space**, **Fill**, **Depth**, or **Files** path.

> **Note:** Automatically resolved 32-bit channels cannot be converted to **Scalar** data. Only 8- and 16-bit channels are supported.

> **Tip:** You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.

9. If you want to import textures listed for the channel, check that Mari has found them (as indicated by a green bar or a number listing the number of textures found for the model). If Mari hasn't found them (as indicated by a red bar), you can click to bring up the **Open** dialog box in order to manually find the texture files.
10. Once you are happy with the channels and textures selected, click **OK**.

    Mari creates the channels, and imports any selected textures.

> **Tip:** Depending on the number of channels and the size of the textures, the import could take a long time. Consider taking a little break!

> **Tip:** To toggle quickly between the last two selected channels (while in the painting canvas), press **T**. For other options, such as locking or resizing a channel, right-click in the **Channels** palette or click on the **Channels** menu.

Navigating Between Channels

There are several ways to navigate between the channels in the **Channels** palette list:

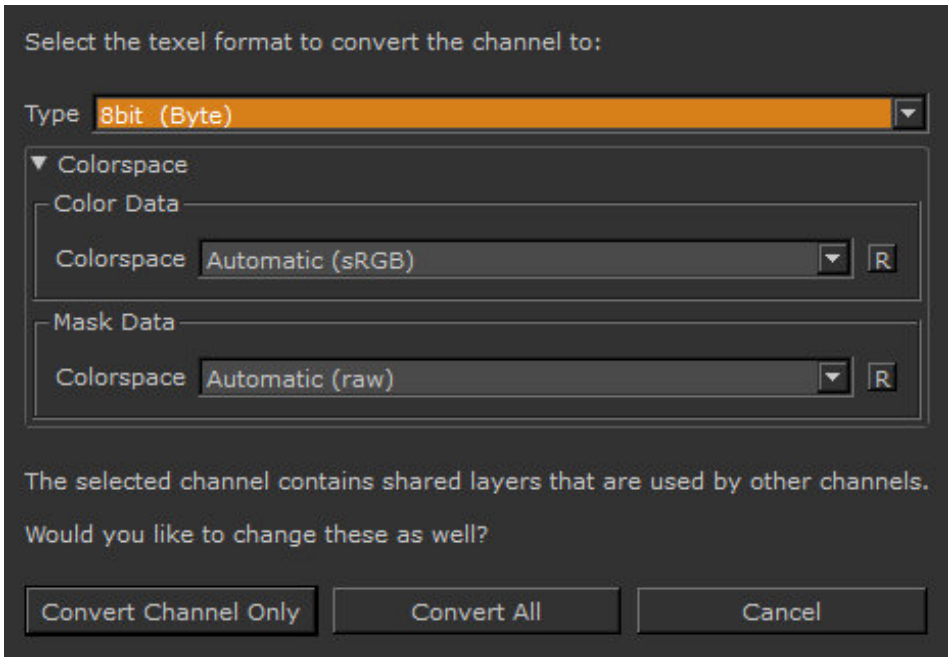
- To view a particular channel, click on the channel in the **Channels** palette.
- To move to the next or previous channels, click **Channels** > **Next Channel** or **Previous Channel**, or use the up and down arrow keys to step through the list.
- Press **Page Up** to move to the first channel visible in the list and **Page Down** to move to the last visible in the list. If you have a long list of channels, you can resize the pane to view more channels at once. Pressing **Page Up** and **Page Down** selects whichever channels are visible at the top and bottom of the pane.
- Press **T** to toggle between the last two channels viewed.

> **Note:** When using the **T** shortcut, the mouse focus should not be in the **Channels** palette, or the selection doesn't toggle. If you aren't seeing the channel selection toggle, click outside of the palette, then try again.

Setting and Changing Channel Properties

1. To change the color depth of an existing channel, either:
   - right-click in the **Channels** palette, and select **Convert Channel** from the dropdown menu,
   - with the channel selected, click , or
   - from the **Channels** menu, select **Convert Channel**.

   The **Convert Channel** dialog box displays.



2. Select the new **Type** (color depth) and the base color for the channel, and click **Ok**. For more details, see the Channels Palette.

Video: Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Taking and Managing Channel Snapshots

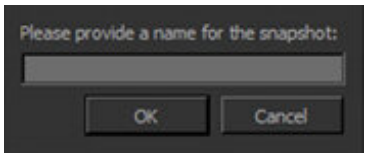
You can take snapshots of the current channel or all channels in a project.

Current Channel Snapshots

1. To take a snapshot of the current channel:
   - select **Channels** > **Snapshots** > **Snapshot Current Channel**, or

- from the **Snapshots** palette, with **Channels** set to **Current**, click New, or
- right-click the channel and select **Snapshots** > **Snapshot Current Channel** from the dropdown menu.

The **Snapshot Channel** dialog box displays.

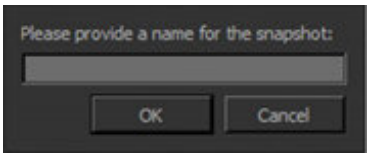


2. Enter a name for the snapshot and click **OK**.

All Channel Snapshots

1. To take a snapshot of all channels in the project, either:
   - select **Channels** > **Snapshots** > **Snapshot All Channels**, or
   - from the **Snapshots** palette, with **Channels** set to **All**, click New, or
   - right-click the channel and select **Snapshots** > **Snapshot All Channels** from the dropdown menu.

   The **Snapshot All Channels** dialog box displays.



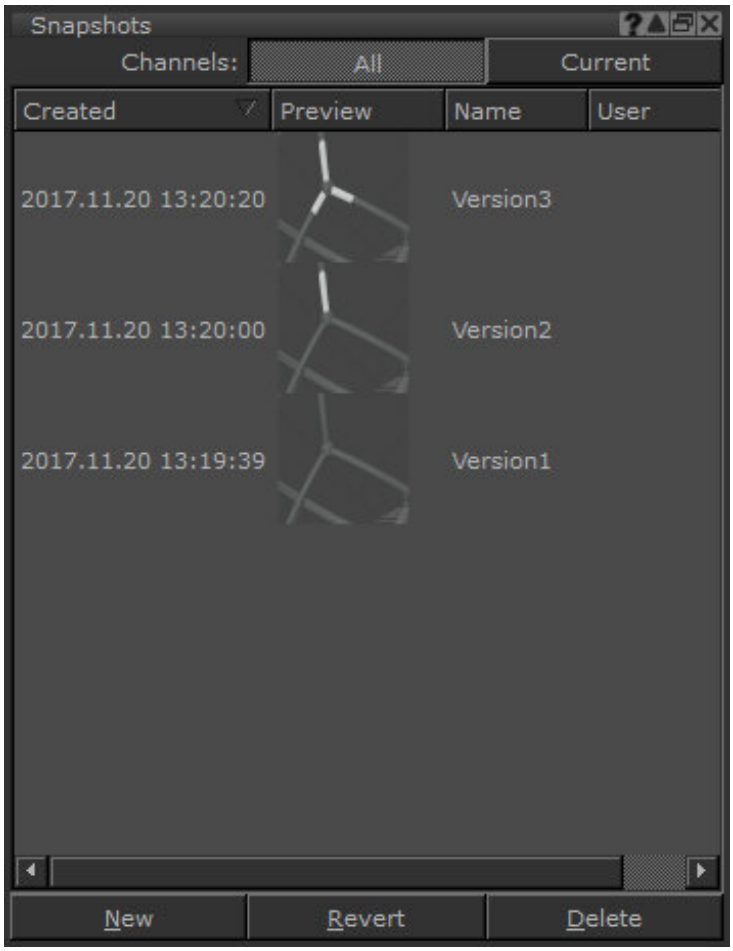
2. Enter a name for the snapshot and click **OK**.

> **Note:** Before a snapshot is taken you need to bake or clear the paint buffer. If you have not done this already, a dialog displays asking you to do so.

Using the Snapshots Palette

Once taken, snapshots are added to the **Snapshots** palette. To open the **Snapshots** palette:

- right-click on the toolbar, and select **Snapshots** from the menu,
- from the **View** menu, select **Palettes** > **Snapshots**, or
- from the **Channels** menus, select **Snapshots** > **Manage Snapshots**.



The **Snapshots** palette displays, showing the existing snapshots for the current channel and for the entire project. Information about the snapshot, such as when it was taken and by which user, along with a thumbnail preview, is also displayed.

There are a number of options available for managing existing snapshots. From the **Snapshots** palette you can create **New** snapshots, **Revert**, or **Delete** snapshots.

Loading an Existing Snapshot:

Select the snapshot and click Revert.

Mari loads the selected snapshot into the current channel.

Deleting a Snapshot:

1. Select a snapshot and click Delete.

The **Delete Snapshot** dialog displays, asking you to confirm.

2. Click **OK**.

   The snapshot disappears from the **Snapshots** palette.

Autosnapshots

Mari lets you create incremental copies of your textures as you paint. These copies, or snapshots, save the current state of either a single channel, or all channels in a project. Snapshots are useful for:

- Keeping multiple versions of a texture while trying out new ideas.
- Maintaining a development history between sessions, and having the option to undo mistakes.

Once a snapshot is taken, it is added to the **Snapshots** palette where you can see a list of existing snapshots and options for managing them.

When saving a project, Mari automatically takes a snapshot of all available channels and adds it to the **Snapshots** palette. This feature is disabled by default but can be managed in the **Mari Preferences** dialog.

To manage autosnapshots:

1. Click **Edit** > **Preferences**.

   The **Mari Preferences** dialog displays.

2. From the dropdown menu under **Data** > **Channels** > **Autosnapshot** select whether autosnapshots are:
   - **Disabled**,
   - **Enabled with Limit**, or
   - **Enabled - Unlimited**.

3. When **Enabled with Limit** is selected, Mari limits the number of autosnapshots that are stored in the **Snapshots** palette to the number specified by the **Autosnapshot Limit**. If the number of autosnapshots stored exceeds this limit, the oldest autosnapshot is removed from the palette.

> **Note:** The **Autosnapshot Limit** does not affect the number of regular channel snapshots that can be stored in the **Snapshots** palette.

Sharing Channels

Unlike the layer sharing functionality in the **Layers** menu and palette, channel sharing doesn't have a menu option in either the context menu or **Channels** menu. Sharing channels is different from sharing layers, in that the functionality is much more limited. Channels can be, however, be shared into the layer stack of other layers. Follow the steps below to share channels into layers.

To share a channel into the layer stack of another channel in the **Channels** palette:

1. Select the channel (channel A), press **Shift** and drag the channel you want to share over another channel in the **Channels** palette (channel B).

   Channel B becomes highlighted, to show that it is now the current selected channel.
2. Don't drop channel A on top of channel B. Once channel B becomes highlighted, drag channel A into the layer stack in the **Layers** palette.

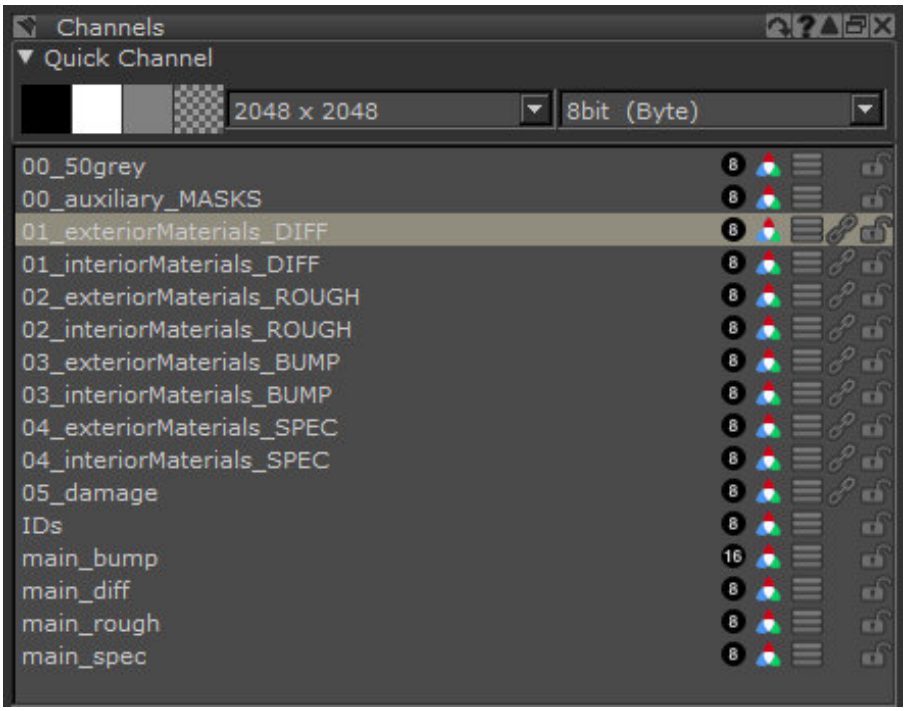
   Channel A is now shared and appears in channel B's layer stack as a layer of the same name. Both the shared channel and layer display linked icons in the palette.

> **Tip:** It is much easier to share channels into a layer stack if both the **Channels** and **Layers** palettes are open, not only simultaneously, but side by side or near each other in your palette configuration.

Locking and Unlocking Channels

1. Lock the current channel:
   - with the channel selected, from the **Channels** menu, select **Lock Channel**,
   - from the **Channels** palette, right-click on the channel and select **Lock Channel** from the dropdown menu, or
   - from the **Channels** palette, click on the lock icon next to the channel you want to lock.

   The current channel is locked.

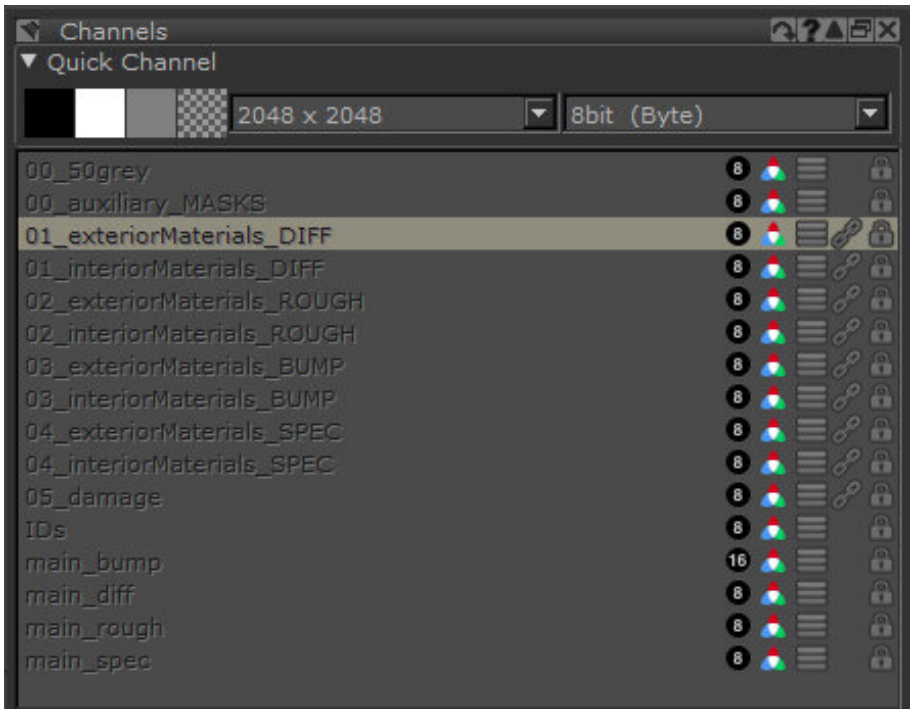


The channel's listing in the **Channels** palette changes to show the “closed lock” icon .

> **Note:** If you attempt to paint on the canvas when a channel is locked, the paint shows up in the paint buffer. When you attempt to bake, the paint buffer is cleared without baking the paint down into the locked channel. If you didn’t want the paint buffer to be cleared, simply undo the bake.

2. Lock all channels:
   - from the **Channels** menu, select **Lock All Channels**, or
   - from the **Channels** palette, right-click on the channel and select **Lock All Channels** from the dropdown menu.

   All channels in the **Channels** palette are locked.



3. Unlock the current channel:
   - from the **Channels** menu, select **Unlock Channel**,
   - from the **Channels** palette, right-click on the channel and select **Unlock Channel** from the dropdown menu, or
   - from the **Channels** palette, click on the lock icon next to the channel you want to unlock.

   Mari unlocks the current channel and changes its listing in the **Channels** palette back to the "open lock" icon.
4. Unlock all channels:
   - from the **Channels** menu, select **Unlock All Channels**, or
   - from the **Channels** palette, select **Unlock All Channels**.

   All channels in the Channels palette are unlocked.

Note: Please be aware that if you have used the **Lock All** or **Unlock All** functions from the dropdown menu on the canvas, this does not affect the lock state of the channels. However, **Lock Everything** and **Unlock Entire Object** applies to all objects, patches, and faces, and channels.

Exporting Channels

Exporting channels to save work to an external source, or to pass on to the next stage of your pipeline is possible on both an individual and collective channel basis.

You can export channels with all layers together in a single file, either in a multi-layered file format, such as **.psd** or **.psb**, with each layer separate in a non-layered format, such as **.png**, or with the layers all flattened together. You can also export the current channel, all channels, or the channels for all objects in your project.

To export the currently-selected channel:

1. From the **Channels** menu, select **Export** > **Export Current Channel**, or right-click the channel in the **Channels** palette and select **Export** > **Export Current Channel** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your textures for the selected channel.

To export all the channels in the **Channels** palette:

1. From the **Channels** menu, select **Export** > **Export All Channels**, or right-click in the **Channels** palette and select **Export** > **Export All Channels** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your textures for all channels in the **Channels** palette.

To export all the channels from all the objects of your project:

1. From the **Channels** menu, select **Export** > **Export Everything**, or right-click in the **Channels** palette and select **Export** > **Export Everything** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your textures for all objects.

> **Note:**
> As the options above only export channels one dialog at a time, to export multiple channels at once, see the Export Manager Dialog.

Exporting Flattened Channels

For channels with large layer stacks, you can also opt to flatten layers before export to save space and organize finished work.

> **Note:** Exporting channels flattened does not flatten the layer stack in your current project; only the layer stack in the exported file. For this reason, you won't need to worry about changing the work in your current project, but can save time and space when exporting files.

To export the currently selected channel with flattened layers:

1. From the **Channels** menu, select **Export Flattened** > **Export Current Channel Flattened**, or right-click the channel in the **Channels** palette and select **Export Flattened** > **Export Current Channel Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your flattened textures for the selected channel.

To export all the channels in the **Channels** palette with flattened layers:

1. From the **Channels** menu, select **Export Flattened** > **Export All Channels Flattened**, or right-click in the **Channels** palette and select **Export Flattened** > **Export All Channels Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your flattened textures for all channels in the **Channels** palette.

To export all the channels, from all the objects in your project, with flattened layers:

1. From the **Channels** menu, select **Export Flattened** > **Export Everything Flattened**, or right-click in the **Channels** palette and select **Export Flattened** > **Export Everything Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Channels before clicking **Export All Patches**.

   Mari exports your flattened textures for all objects.

To export the flattened channel to a Geo-Channel with a channel selected:

1. Click the **Channels** menu, select **Export Flattened** > **Export Flattened to Geo-Channel**.

   A **Export to Geo-Channel** dialog displays.

2. Select a Geo-Channel from the **Geo-Channel** dropdown, or write the alias of a new Geo-Channel inside the dropdown and click **OK**.

   When a new alias is written in, a new Geo-Channel is created with that name. The flattened channel is exported to the selected Geo-Channel.

> **Note:**
> As the options above only export channels one dialog at a time, to export multiple channels at once, see the Export Manager Dialog.

Importing Channels

Importing channels allows you to import textures that have already been worked on or that come from another source. Importing channels is done by importing the textures into the layer stack or into a new channel.

To import textures into the layer stack of the currently selected channel:

1. From the **Channels** menu, select **Import** > **Import into Current Channel**, or right-click channel in the **Channels** palette and select **Import** > **Import into Current Channel** from the dropdown menu.

   The **Import** dialog displays.
2. Follow the steps in Importing Textures to Channels before clicking **Import All Patches**.

   Mari imports the textures into the layer stack of your currently selected channel.

To import textures into a new channel:

1. From the **Channels** menu, select **Import** > **Import into New Channel**, or right-click in the **Channels** palette and select **Import** > **Import into New Channel** from the dropdown menu.

   The **Add Channel** dialog displays.
2. Set the options for your new channel and assign it a name. Click **OK**.

   The **Import** dialog displays.
3. Follow the steps in Importing Textures to Channels before clicking **Import All Patches**.

   Mari imports the textures as a new channel.

Transferring Channels

You can transfer existing channels from one object or version to another. This is useful for re-creating work on an object with minimal changes and similar topology or an object with the same topology but a different UV layout. By using the old textures transferred across onto the new object, you can save time instead of starting again from scratch.

Mari uses the Bakery to transfer the nodes and paint data to the target object, avoiding the need to manually redo existing work on a new version of the geometry.

Using the **Channel Transfer** dialog, you can specify which object, version, and patches to use as a source for the texture, and which of these to use as a destination.

You can define the **Transfer Method** option to specify how the textures are remapped onto the destination object.

In some cases, where the topology of the object might have changed between versions, there may be inconsistencies from the texture transfer. You can specify a patch **Fill Color** to highlight where the textures didn't transfer to a patch, and then fix only these patches separately after the transfer.

The Texture Transfer enables you to bake textures quickly and move them between assets.

To Transfer textures in Mari:

1. Select the channels you want to transfer.
2. Select **Channels > Transfer**. Alternatively right-click to open the context menu and navigate to **Selection > Transfer**. The Channel Transfer dialog opens.
3. Adjust the Channel Transfer settings:
    - **Source** - Select the **Object**, **Version**, and **Patches** that the channels can currently be found on.
    - **Destination** - Select the **Object**, **Version**, and **Patches** that the channels are being transferred to.
    - Use the arrow buttons to move the selected channel(s) back and forth between the source and destination channel lists.
    - **Transfer Method**. Specify the method for transferring the texture onto the destination UV map. Choose between:
        - **UV Remap From Bakery** - Uses the Bakery to generate the UV coordinate mappings from the source object to the destination object.

- **UV Remap From File** -Uses an import-like template to remap pre-generated textures based on UV coordinates. For example, if you have generated the UV map beforehand, you can load it here.

- **Bleed Edges** - Enable to generate edge bleed on the resulting image set.
- **Fill Color** - Define the initial color to fill the resulting image set with. If an object has extra patches that aren't covered by the transfer, then the fill color specifies whether these extra patches are filled with Transparent, Black, White, Mid-gray, Magenta, Foreground, or Background.

4. Click **Ok**.
5. The transfers will then be added to the node graph of the object you are transferring to.

Layers

Layers are the primary system for painting in Mari, they are the source of your paint textures in a project. Layers are created as part of a layer stack, which is in turn held within a specific channel. Each channel has a different layer stack that contains a new set of layers. You can create paint layers, layer masks, adjustments, procedurals, and layer groups within the **Layers** palette. The layer system combines effects in a non-destructive manner. The real strength of the layer system lies in the way layers integrate with shaders and channels.

Video: Watch this video to understand Mari's Layer System.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Layer Integration

Mari's layer system does not function in isolation; instead, it works together with shaders and channels to create a cohesive structure for painting. Layers are entirely dependent on channels and need to have diffuse, specular, or bump channels already set up in order to be created.

As in previous versions of Mari, channels are created with the choice of channel type, size, and depth; however, instead of painting directly into the channel as before, channels now allow the use of separate layers within them to create your texture. This gives you more control over how you paint, and gives you the

option of creating non-destructive textures that can be integrated into the rest of your project. Channels are your foundation for working with layers, but what are these layers capable of, and what are your options?

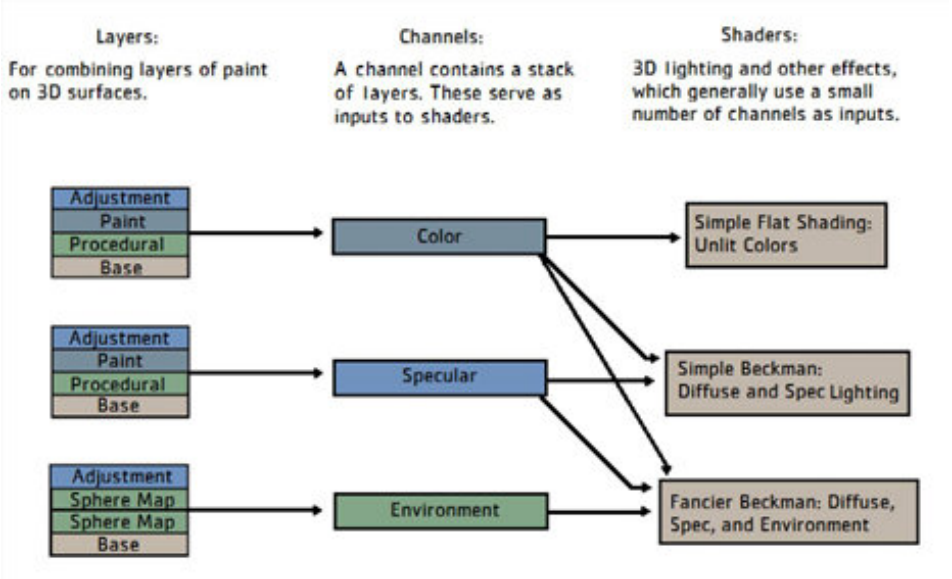
Layers can be divided into a few categories: paint layers, layer masks, adjustments, and procedurals. In turn, these layers can be ordered into groups, as well as shared, merged, or exported, to name only a few features. Layers can even be converted into new channels with existing paint in the base layer. Experiment with them more to discover what you can do.

Note: For more information about channels, please see Channels.

Shader Dependence

Shaders are also important in the use of channels and layers, but they have received a makeover compared to their use in previous versions of Mari. Shaders now allow you to set up lighting and specify which channels should be used for each channel type in the shader system. Depending on the lighting system used, you can also control elements that affect your channels, such as bump weight, reflectiveness, and ambient occlusion.

Shaders also serve the vital purpose of giving you a quick way of comparing your texture under different conditions. Switching between the pre-existing shaders gives you a fast way of viewing your textures within the lighting shader you've set up, or simply viewing the current channel or layer stack.



Co-dependence of channels, layers, and shaders.

Note: For more information about shaders, please see Shaders.

Getting Started with Layers

You can find most of the new features regarding layers and masks within the **Layers** palette or the **Layers** menu. The **Layers** palette is divided into two panes. The top pane displays all the layers for that channel, and the bottom pane provides information and controls to adjust the selected layer, when applicable.

All the controls for adding, editing, or deleting layers can be found on the right-click menu or the buttons on the palette. Adding layers and masks immediately adds the selected component, whereas adding adjustments and procedurals opens a dropdown menu for further selections.

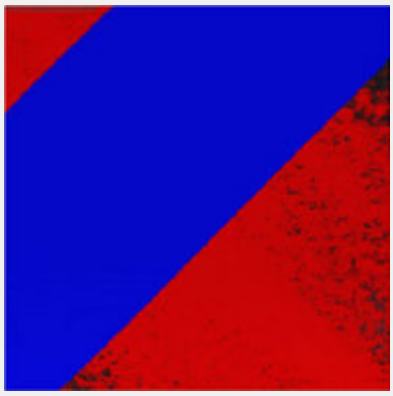
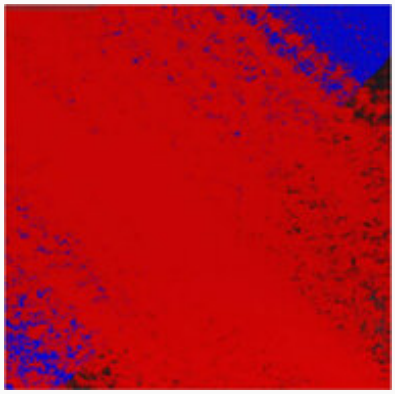
Blend modes are also found on the **Layers** palette, as well as advanced blending options. Unlike the **Paint Mode**, found on the **Paint Buffer** toolbar, these blend options don't affect the whole model, but only the selected layer(s).

Article: Read Q100502: Using Materials in the Layer Stack for more information on the unified stack rules when using Multi-Channel Layers.

Ordering Layers

Mari applies the layers in the order they appear in the palette, from the bottom of the list up (as in Photoshop®). You can drag and drop them around in the palette to change the order. Every channel that you've created automatically has a base layer in the layer stack. This is the foundation all other layers are built on. If the base layer is deleted, the object displays the transparency checkerboard wherever paint from the above layers doesn't cover the model.

The following table illustrates a basic layer stack with two layers in a diffuse channel. One layer shows a red stripe, the other shows a blue stripe. The order of the two layers determines which texture appears on top. The table shows the results of swapping the order of the layers.

Layer Order	Result
	 The diffuse texture (Base) is on the bottom, then the Red, then the Blue.
	 The diffuse texture is on the bottom, but this time the Red is on top of the Blue.

If you like the look of a layer and want to replicate it elsewhere in your project, you can copy and paste the layer into the layer stack or another stack. In addition, layers can also be duplicated within the stack.

Note: For those unfamiliar with layers, the nearer the layer is to the bottom of the stack, the further down it sits in the paint order. Any overlapping paint on layers higher in the stack covers the original paint lower in the stack.

Filtering and Searching for Layers

In the **Layers** palette, there are four methods for quickly finding individual layers in a large layer stack: **Name**, **Type**, **Attribute**, and **Color Tag**. You can choose to filter by name, or to select a layer type or attribute, such as only procedurals, or layers that are visible. Alternatively, you can assign layers with one of eight color tags, and filter according to color.

Locked and Hidden Layers

Layers can be locked to protect them from being mistakenly deleted or edited. If the layer being locked is part of a group, only that layer is locked; however, if the layer being locked is the parent layer of the group, then all the child layers are locked too.

Hiding layers, on the other hand, allows you to examine your work in isolation. Hiding a layer non-destructively removes the texture from the object so that you can see the results of the rest of the stack without it.

Note: For more information about locking or hiding other selected items, see Hiding and Locking.

Dragging Layers for Moving, Copying, and Sharing

You can move, copy, and share layers between different layer stacks - even between layer stacks of different channels - by using modifier keys when dragging:

- **Shift**+drag to share layers.
- **Alt**+drag (or **Ctrl**/**Cmd**+**Alt**+drag) to copy layers.
- Drag (without any modifiers) to move layers.

Note that the modifier keys need to be held down until the layers are dropped, otherwise they have no effect.

To drag layers between the layer stacks of different channels, you need to open those stacks in their own layer stack palettes in advance. Remember to pin the layer stack palettes using the button in the top-left corner so that they do not automatically disappear. Alternatively, you can drag and hover over the relevant channel in the **Channels** palette to activate its layer stack in the **Layers** palette, and then continue dragging and drop the layers there.

Sub-stacks, such as adjustment stacks or mask stacks, automatically pop up additional dialogs if you drag your layers over their icons, so you don't need to open these pop-ups in advance.

Tip: Enabling Pin Palette mode in the Palettes Toolbar makes sure layer stacks, adjustment stacks, and mask stacks always appear over the UI, even when your cursor is moved away from them.

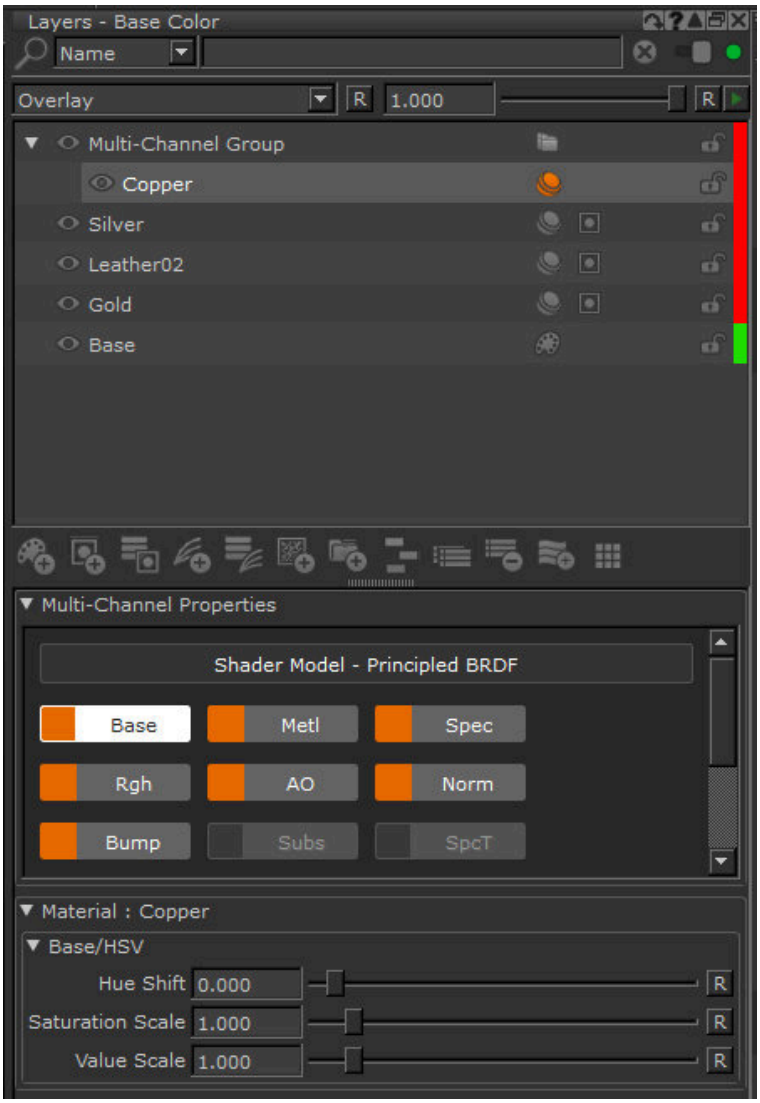
Layer Properties

Any layer in the **Layers** palette can be inspected to view the layer properties, such as **File Space**, **Colorspace**, and information about the channel to which the layer belongs. To view the layer properties, right-click on a layer in the **Layers** palette and select **Layer Properties** from the dropdown menu. The **Layer Properties** dialog appears with the layer format and channel information for the selected layer.

Opening the Layers Palette

1. Start Mari and open your project.
2. Set up the shaders, channels, and layers, as described in Layer Integration.
3. Open the **Layers** palette:
   - right-click on the toolbar and select **Layers** from the dropdown menu,
   - from the **View** menu, select **Palettes** > **Layers.**

   The **Layers** palette displays, with a default base layer in the layer stack.



Layer Blend Modes

Near the top of the palette there are blend mode options in the dropdown menu. This mode can be set for each individual layer, setting it apart from the Painting **Mode** in the **Paint Buffer** toolbar. The key difference is that where Painting **Mode** applies to the paint strokes that are baked down onto the model, the blend modes on the **Layers** palette apply to paint in the entire layer; not just what is baked.

Note: If the blend mode applied to the layer affects the appearance of the overall stack, ensure that the layer is placed in the ideal order in the layer stack. If necessary, the layer can be moved after the blend mode has been set.

Adding Layers and Masks

To add a new layer, either:

- from the **Layers** menu, select **Add New Layer**,
- right-click any layer in the **Layers** palette and select **Add New Layer**, or
- in the **Layers** palette, click .

The new layer appears in the layer stack, above any other layers in the layer stack, with the paintable layer icon.

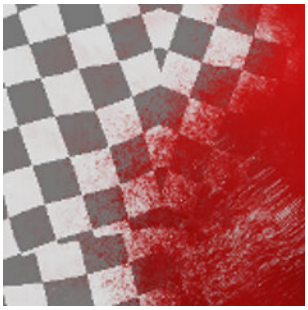
The paintable icon appears orange when the layer is selected and the paintable component is active.

Layer masks allow you to mask out the paint you've already created in that layer. The mask uses a simple black and white map, which is visible by clicking on **Current Paint Target** in the **Shaders** palette. The mask uses black to completely mask out the texture, whereas white completely unmasks the texture.

Creating a mask on a layer (painted in black and white) and deriving alpha values from the mask (where the alpha on a pixel is controlled by the shade on the original mask), combines the alpha values with the diffuse color from the layer to create results as shown below:

Initial B/W mask, shown in **Current Paint Target**

Diffuse layer



Result of creating a layer mask

Any other color used only partially masks out the texture, with the intensity of the mask depending on the amount of black or white in the color.

Note: It's not necessary to switch to **Current Paint Target** in the **Shaders** palette when painting a layer mask. You can still paint in the layer as you would normally, but **Current Paint Target** provides you with an easy way to examine your mask in isolation.

Tip: We recommend that you create:
• mask channels with a white background - so you can paint the mask data black.
• displacement channels with a 50% gray background - this corresponds to the default surface. Mari treats darker colors as negative displacement (going into the surface) and lighter colors as positive displacement (coming out from the surface).

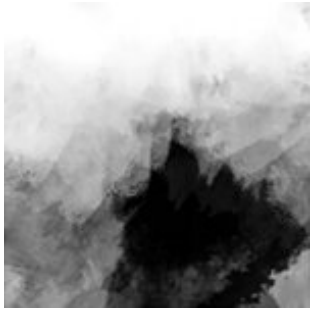
Video: Watch this video to learn how to create Layer Masks and Layer Mask Stacks.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Alpha and Masks

To make creating masks easier, Mari has another function to easily convert between alpha and mask values.

Creating a mask **FromAlpha** does the opposite of a standard mask. It converts a layer with alpha values to a black and white mask, where the shade on each pixel is controlled by the alpha on the original layer to give the results shown below:

Initial channel with alpha values.



Result of **Alpha to Mask.**

To add a layer mask, either:

- from the **Layers** menu, select **Layer Mask** > **Add Mask** > **Reveal All** or **Layer Mask** > **Add Mask** > **From Alpha**,
- right-click the desired layer for the mask, and select **Layer Mask** > **Add Mask** > **Reveal All** or **Layer Mask** > **Add Mask** > **From Alpha**, or
- in the **Layers** palette, click .

The layer mask is added to the selected layer, with the mask icon next to the paintable icon.

The mask icon appears orange when the layer is selected and the mask component is active.

Tip: To switch back and forth between components on a layer, simply click on the component icons.

Note: You can right-click any of the layers in the **Layers** palette to access the dropdown menu for more layer options. Right-clicking an existing layer also selects it.

Adding Adjustments and Procedurals

Adjustment layers are filters that can be added to the layer stack to affect certain layers in the stack. Adjustment layers are location-dependent in the layer stack and affect only the layers below them.

In addition to adding adjustment layers to the layer stack, adjustments can also be added to selected layers. These adjustments are exclusive to the layer they were added to. Whenever an adjustment is added to an existing layer, it automatically creates an adjustment stack that can be opened in a separate palette.

An adjustment layer with a mask on the layer can have a secondary adjustment added to it to apply another adjustment to masked out areas. Like layer-specific adjustments, these are applied directly to the selected adjustment layer and can be modified in the lower pane of the palette.

Video: Watch this video to know more about Creating Adjustment Layers and Adjustment Stacks. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

For a full list of adjustment layers and a more detailed description please refer to Layer Types and Examples.

Follow these steps for working with adjustment layers:

1. To add an adjustment layer to a layer stack, either:
   - from the **Layers** menu, navigate to **Add Adjustment Layer** and select the filter from the dropdown menu,
   - right-click any layer in the **Layers** palette, navigate to **Add Adjustment Layer** and select the adjustment filter from the dropdown menu,
   - in the **Layers** palette, click , or
   - with the **Layers** palette open, press **Tab** and begin typing the adjustment you want to add. The text field is populated with the adjustment.

   The adjustment layer appears in the layer stack, with the adjustment icon where the paintable layer icon would normally reside.

2. To add an adjustment to a layer:
   - with the layer selected, from the **Layers** menu, navigate to **Adjustment Stack** > **Add Adjustment Stack** and select the filter from the dropdown menu,
   - right-click the desired layer, navigate to **Adjustment Stack** > **Add Adjustment Stack** and select the filter from the dropdown menu, or
   - in the **Layers** palette, click , and select the filter from the dropdown menu.

   The adjustment filter is added to the selected layer, with the adjustment stack icon one space down from where the mask icon would appear.

Note: Each filter has adjustment controls located in the bottom pane of the **Layers** palette. Use these to modify how strongly the filter affects your texture.

3. When an adjustment or adjustment stack has been added to an existing layer, you can flatten the adjustment or stack into the layer. To do this:
   - with the layer selected, from the **Layers** menu, select **Adjustment Stack** > **Bake Adjustment Stack**, or

- right-click on the layer and select **Adjustment Stack** > **Bake Adjustment Stack** from the dropdown menu.

Note: The layer blend mode of an adjustment stack is **Overlay**. Due to the fact that 8-bit channels are in an **Automatic (sRGB)** colorspace by default, the base value of 0.5 is converted to 0.2 when it gets to the rendering stage. Layers are then blended with the **Overlay** mode. If an adjustment is then applied to the top layer, it results in other layers than the adjusted layer appearing as if they've been affected. In actuality, they haven't been affected but because of the **Overlay** mode, the final look changes significantly.

Experiment: In order to see how blend modes can affect a layer stack, try:

1. Fill the bottom layer in a stack with 0.5, 0, 0 values.
2. Fill the middle layer in a stack with 0, 0.5, 0 values.
3. Fill the top layer in a stack with 0, 0, 0.5 values.
4. Change all the layer blend modes to **Add**.

   You see a gray 0.5, 0.5, 0.5 value as the result.
5. Now, change the contrast of the top layer by applying a **Contrast** adjustment.

   Notice that it becomes more bluish as you turn up the contrast. This illustrates how the contrast is only applied to the top layer.

Procedural layers allow you to add texture patterns, grid patterns, environments, and masks to the layer stack using a variety of parameter settings. Procedural layers behave similarly to paint layers, with the exception that their image content cannot be painted on - it can only be modified by changing the parameters of the layer.

Video: Watch this video to learn how to create Procedural Layers.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Note: In Procedurals that include color attributes, the color controls contain a toggle that sets the context in which the procedural effect is to be used for Scalar or Color data. By default, Mari detects the channel's color data type and sets the toggle accordingly for layers created through the **Layers** palette. For Procedural nodes created through the Node graph, this color/scalar toggle defaults to color. Procedural nodes that are to be used within masks and scalar channels have to be set to scalar manually. See Color Data and Scalar Data for more information.

For a full list of procedural layers and a more detailed description please refer to Layer Types and Examples.

To add a procedural layer, either:

- from the **Layers** menu, navigate to **Add Procedural Layer**, and select a procedural from the dropdown menu,
- right-click any layer in the **Layers** palette, navigate to **Add Procedural Layer** and select a procedural from the dropdown menu,
- in the **Layers** palette, click , or
- with the **Layers** palette open, press **Tab** and begin typing the procedural you want to add. The text field is populated with the procedural.

  The procedural layer appears in the layer stack, with the procedural icon where the paintable layer icon would normally reside.

Note: You can also create Smart Masks to import into the layer stack. Click > **Smart Masks** to see a list of available Smart Masks in the **Shelf**. See Smart Mask Node for more information.

Projection Procedurals

Cubic and spherical projection procedural layers allow you to import a texture into the scene and project it onto the geometry. Using multiple **Cube Map Projector** or **Sphere Map Projector** procedurals enables you to get the best possible projection from the different camera locations in a scene. Using multiple camera locations allows you to account for occlusion and perspective when projecting textures onto the geometry.

There are three projection procedurals:

- **Cube Map Projector**,
- **Sphere Map Projector**, and
- **Triplanar Projection**.

You can also use masks with the projection procedurals to mask out parts of the projection. This is helpful in eliminating unwanted artifacts, for example, where the projection overlaps with other projections in the scene.

Each projection procedural allows you to control:

- texture offset and rotation,
- edge and distance falloff start and end points, and
- falloff curve shapes, using the curve editors.

Note: Adjusting options such as falloff and texture offset or rotation affects the whole scene.

Mipmap Behavior

For the **Cube Map** procedural, **.dds** files with missing mipmaps or partial mipchains continue to load, but the behavior for these chains is slightly different. Mari disregards all other mipmaps in the chain; instead, it regenerates the chain from the top level mip.

In this way, Mari continues to load partial mipmap chains in a workable manner for **.dds** files.

Adding Graph Layers

Mari's graph layers are building blocks used in Mari's Node Graph.

To add a new graph layer, in the **Layers** palette, click the Add Graph Layer button.

The new layer appears in the layer stack, with the Node Graph layer button.

To view the Node Graph, navigate to **View** > **Palettes** > **Node Graph** or in the **Palettes Toolbar** (see Palettes Toolbar), click the Node Graph.

Note: To open a graph layer in the **Node Graph**, open the **Node Graph** palette and in the **Layers** palette, double-click on the Node Graph layer button of the layer. This opens a Node Graph layer tab in the **Node Graph** palette.

Graph layers are actually Groups, in Node Graph terms, containing an Input node and an Output node. Any graph layers you add are included in the node tree for the current channel.

To add nodes to the graph layer, either:

- press the **Tab** key in the Node Graph and start typing the name of the node you want to create, and then press **Return** to add the node, or
- right-click in the Node Graph and select the required node from the menu.

Note: See Working with Nodes and Layer Types and Examples for more information on the Node Graph and the nodes available. Graph layers can be exported and imported in exactly the same way as other layers. See Exporting and Importing Layers or Masks.

Cube Mapping

Cube maps provide a context for reflecting images of surrounding environments on the model, similar to the Sphere Map procedural layer. The difference is that the environment is modeled by six textures mapped on a cube surrounding the object as opposed to a single texture mapped on a sphere surrounding the object. Cube mapping uses the **.dds** file format to load files into the Cube Map procedural layer. As with the sphere map, the cube allows you to modify the falloff starting and ending points to create the exact level of reflection you want on your model.

Note: When exporting **.dds** files, they should be encoded as 8.8.8.8 ARGB files to avoid any complications when bringing them into Mari.

Vector Brushing

A Python example script for setting up default vector shaders can be found under **Python** > **Examples** > **Setup Vector Brush** or by clicking the icon on the **Vector Painting** toolbar. This script automatically creates two Vector shaders in the **Shaders** palette, three channels in the **Channels** palette, and layers in each of these channels to get you started. For more information on using the setup vector brush script and the default vector shaders, refer to the section on Adding Vector Shaders.

The two vector brush shaders are **Vector (World Space)** and **Vector (Screen Space)**. The three channels are **Vector (Paint)**, **Vector (World Space)**, and **Vector (Screen Space)**, and these are automatically assigned to the shader component inputs for the relevant shaders created by the script. Each of the channels has layers in the layer stack to get you started. These layers include **Vector**, **Vector (Tangent To Screen Space)**, and **Vector (Tangent To World Space)**.

These channels and layers assist in mapping textures according to direction, by converting pixels on the object's surface to RGB values representing -1 to 1 in xyz space. For more information on using the shaders, channels, and layers together with the **Vector Painting Brush**, refer to Vector Brush.

Additionally, you can opt to add a **Flow** adjustment, which allows you to see the effects of your vector paint over time. The **Time Offset** slider represents the flow of time, while the **Speed** slider represents the sensitivity of **Time Offset.** Adjusting the sliders allows you to see the effects your vectors play on the flow map, though animating this also displays the effect continuously. For more information on the vector adjustments and procedurals, as well as their layer property fields, refer to the section on Layer Types and Examples.

Removing and Disabling Layers

1. To remove a layer:
   - with the layer selected, from the **Layers** menu, select **Remove Layers**,
   - right-click on the layer and select **Remove Layers**,
   - select the layer and press **Delete** on the keyboard, or
   - in the **Layers** palette, click .
2. To remove masks from existing layers:
   - with the layer selected, from the **Layers** menu, select **Layer Mask** > **Remove Mask**, or
   - right-click on the layer and select **Layer Mask** > **Remove Mask** from the dropdown menu.

3. To only disable the mask on the layer, so you can compare the results of the layer without the mask in place:
   - with the layer selected, from the **Layers** menu, select **Layer Mask** > **Disable Mask**, or
   - right-click on the layer and select **Layer Mask** > **Disable Mask** from the dropdown menu.

   Once you want to re-enable the mask visibility again, use the same methods above to navigate to **Layer Mask** > **Enable Mask**.
4. To remove the adjustment stack from an existing layer:
   - with the layer selected, from the **Layers** menu, select **Adjustment Stack** > **Remove Adjustment Stack**, or
   - right-click on the layer and select **Adjustment Stack** > **Remove Adjustment Stack** from the dropdown menu.
5. To only disable the adjustment stack, so you can compare the results in the layer without the stack in place:
   - with the layer selected, from the **Layers** menu, select **Adjustment Stack** > **Disable Adjustment Stack**, or
   - right-click on the layer and select **Adjustment Stack** > **Disable Adjustment Stack** from the dropdown menu.

Once you want to re-enable the adjustment stack visibility again, use the same methods above to navigate to **Adjustment Stack** > **Enable Adjustment Stack**.

Groups, Pass-Through, and Mask Stacks

Multiple layers can be grouped together either by selecting the existing layers and creating a group with them, or by creating a new group without any initial layers and moving layers into them later. Layers can also be moved out of a layer group by simply dragging them back to the main layer stack or into another group.

Creating Groups

1. To add an empty layer group, either:
   - from the **Layers** menu, select **Add Empty Layer Group**,

- right-click any layer in the **Layers** palette and select **Add Empty Layer Group** from the dropdown menu, or
- in the **Layers** palette, click .

2. To create a group from existing layers:
   - with layers selected, from the **Layers** menu, select **Group Layers**,
   - right-click the selected layers and select **Group Layers** from the dropdown menu,
   - from the **Layers** palette, click , or
   - Press **Ctrl**+**G**.

Tip: You can also drag existing layers into the group.

Removing Groups

To remove a layer group:

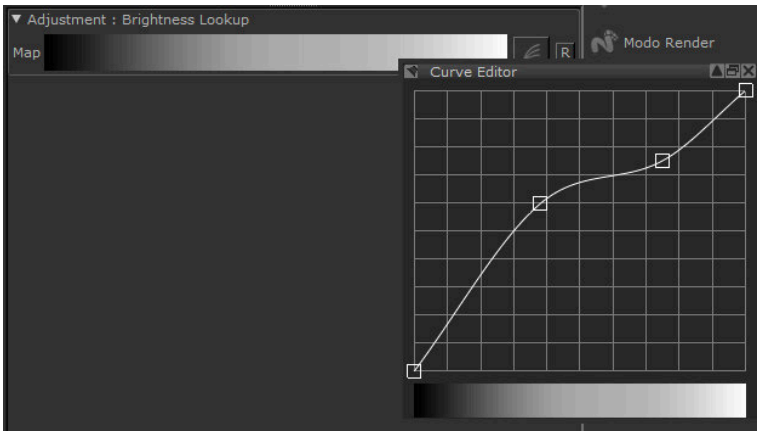
- From the **Layers** palette, right-click one or more layer groups and select **Ungroup Layers**,
- From the **Layers** menu, select **Ungroup Layers**, or
- Press **Ctrl**+**U**.

The layer group is deleted and the layer order is preserved.

Using Pass-Through with Groups

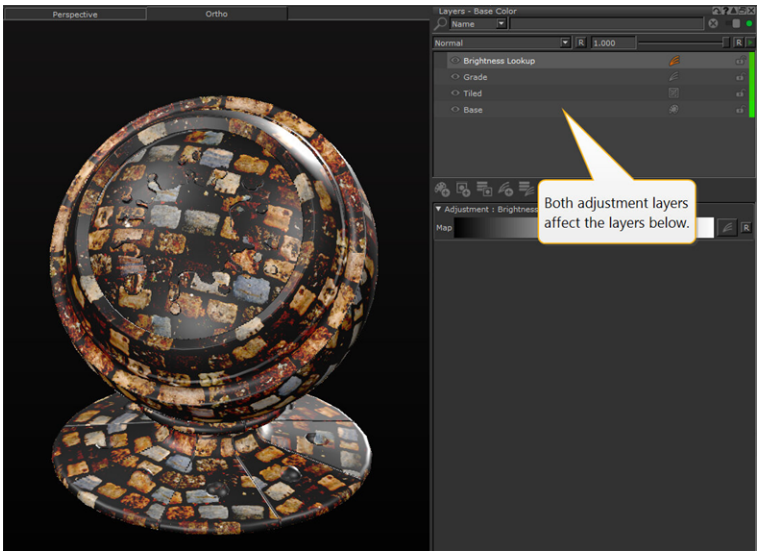
You can use the Pass-Through mode with groups to make the layers within them affect the layers below the groups. The steps below show you one way to use the Pass-Through mode in a particular scenario:

1. In the **Layers** palette, click the Add Procedural Layer button, navigate to **Procedural** > **Pattern** > **Tiled** and add a **Tile Image**.
2. Click the Add Adjustment Layer button and add a **Grade** and a **Brightness Lookup** adjustment layers.
   - Change the brightness value of the **Brightness Lookup** adjustment layer.



• Change the offset of the **Grade** adjustment layer to -0.350 for example.

This is how your scene might look now.

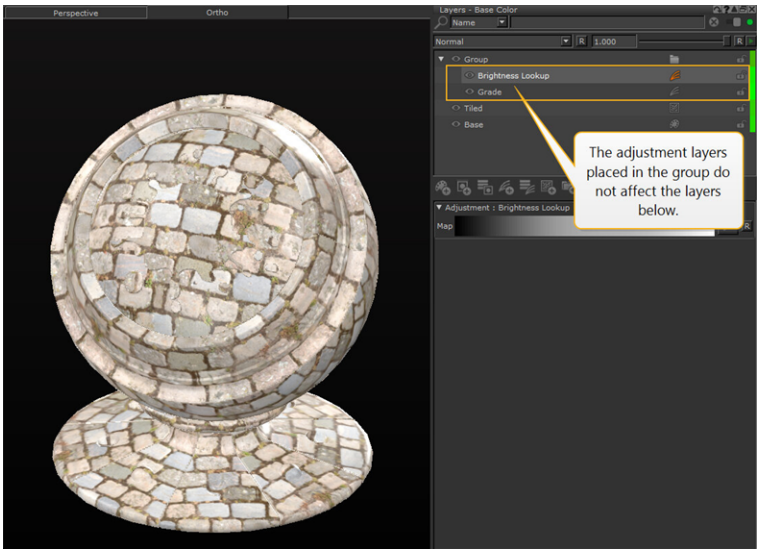


3. Select both adjustment layers, then click the **Group Layers** button.

Tip: If you want to add an empty group, make sure that you haven't selected any layers in the layerstack, otherwise it creates a group with the selected layer(s) in it.

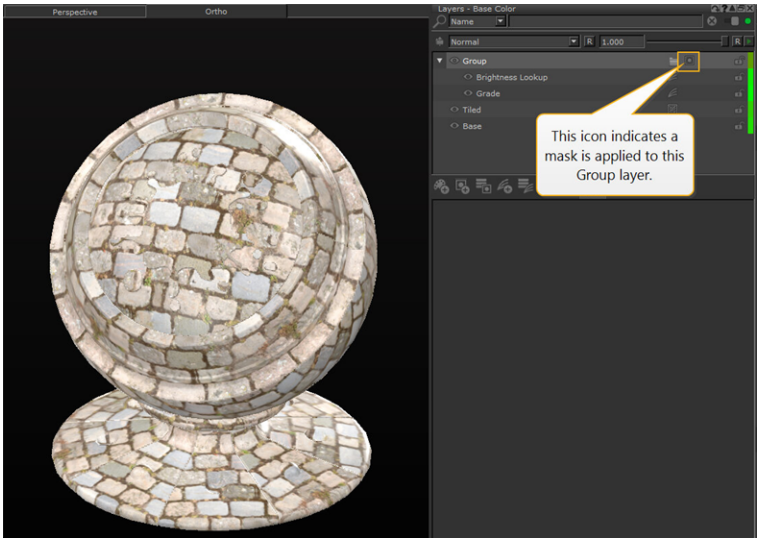
This adds a group, which contains both of your adjustment layers.

Since both of your adjustment layers are in a group, they are not applied to the layers below, the Tiled layer in this example, as they are applied to the bottom of the stack within the group, which is transparent.



4. To have more control on what the group influences, you can add a mask to your group. On the canvas, select a part of your object, in the **Layers** palette, right-click on the group layer and navigate to **Layer Mask** > **Add Mask** > **From Selection**.

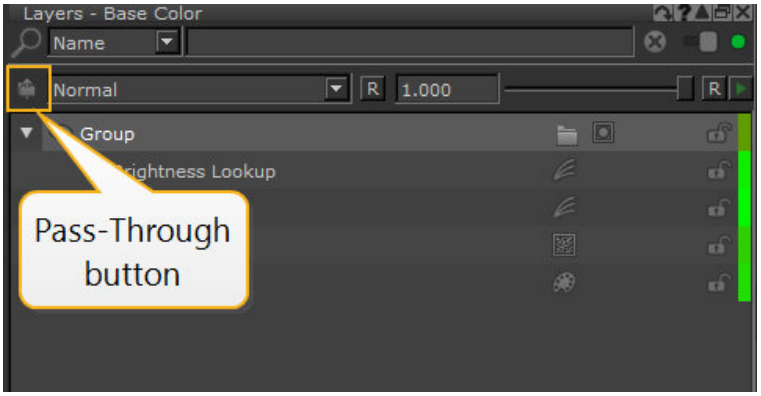
   This adds a mask to your group.



To pass through the effects of the adjustment layers to the layers below:

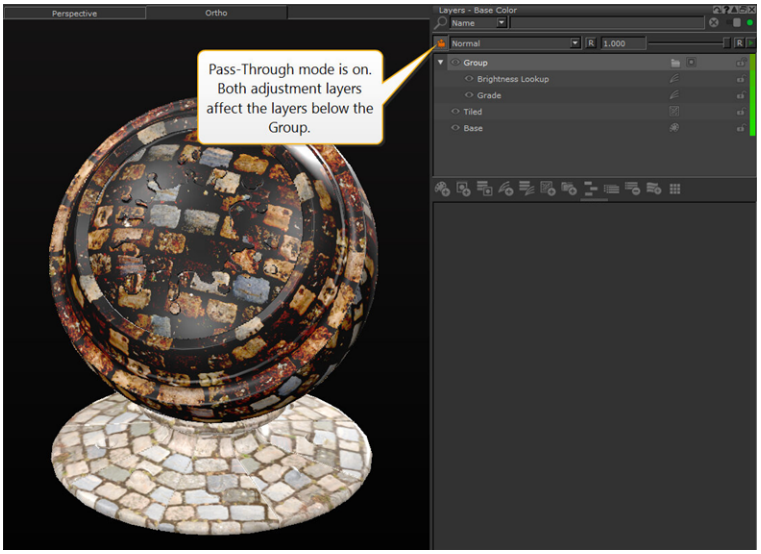
5. In the **Layers** palette, select the group.

   The Pass-Through button appears at the top of the **Layers** palette.



6. Click the Pass-Through button.

   The adjustment layers' effects are passed through the layers below on the part where the mask is applied.



Flattening Groups

Either for organizational or performance reasons, you may decide you want to flatten a group into a single layer. Layers that were part of a group appear as if they are a single layer once flattened. Once a group has been flattened, it can be added to other groups or used to create a new group as with any other layer.

To flatten a layer group into a single layer:

- with the group selected, from the **Layers** menu, select **Flatten Layer Group**, or
- right-click on the group and select **Flatten Layer Group** from the dropdown menu.

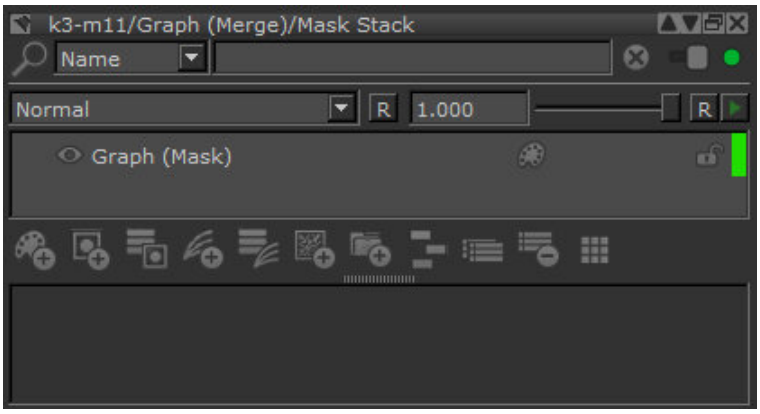
You can also create mask stacks by turning an existing mask into a mask stack. You must already have a mask attached to a layer to create a mask stack, but once you do, you can open a separate **Mask Stack** palette to see all the mask layers in the group.

Adding and Removing Mask Stacks

To add a mask stack to a layer, in the **Layers** palette, click . To create a mask stack from an existing mask on a layer:

- with the layer selected, from the **Layers** menu, select **Layer Mask** > **Make Mask Stack**, or
- right-click on the layer, and select **Layer Mask** > **Make Mask Stack** from the dropdown menu.

The mask stack icon takes the place of the mask icon in the layer list. Click on the icon to open the mask stack palette.



To remove a mask stack:

- From the **Layers** palette, right-click on a mask stack and select **Layer Mask** > **Remove Mask**.
- From the **Layers** menu, select **Layer Mask** > **Remove Mask**.

Flattening Mask Stacks

Mask Stacks can also be flattened. The only difference between flattening a layer group and flattening a mask stack is when a mask stack has been flattened, the group icon disappears from the layer and is

replaced with the mask icon again. All the mask layers that were part of the mask stack now exist in one single mask.

To flatten a mask stack into a single mask:

- with the layer selected, from the **Layers** menu, select **Layer Mask** > **Flattened Mask Stack**, or
- right-click on the layer with the mask stack and select **Layer Mask** > **Flatten Mask Stack** from the dropdown menu.

The mask stack icon reverts back to the normal mask icon and the mask stack is now only a single mask.

Merging Layers

Merging layers is similar to flattening layers, but can be done with any type of layer. Instead of flattening a group of layers into a single layer, merging allows you to combine the textures from two or more layers into a single layer, including any masks or adjustments that might have been applied to them. However, be aware that this is a destructive process, and the results may not be the same as anticipated in the layer stack.

> **Note:** You cannot merge a layer with a layer group that has not been flattened. If you want to merge a group with another layer or layers, you need to first flatten the group.

1. To duplicate layers, in the **Layers** palette, click .

   The layer is duplicated as a copy.
2. To merge layers, right-click on the selected layers, and select **Merge Layer**s.

   The selected layers are merged into a new layer.

Sharing Layers

If you want to duplicate a layer as a shared (connected) layer in your layer stack, you can use the **Share Layer** option from the context menu. This allows you to link the layers so that any changes made to the original layer are also produced on the shared layer. Unlike simply duplicating a layer, sharing a layer lets you make changes to multiple copies by changing only a single linked layer.

If you want to duplicate finished layers as shared layers in a new channel, you can select **Share Layers As Channel** from the context menu or **Layers** menu. As with the shared layer in a layer stack, any changes made to the original layers are also made to the shared layers in the new channel.

If you want to make changes to the original layers and don't want them to be copied over to a shared layer, you can select **Unshare Layer** from the context menu to stop sharing the layer, before continuing your work.

1. To share a layer in the same layer stack:
   - with the layer selected, from the **Layers** menu, select **Sharing** > **Share Layer**, or
   - right-click on the layer and select **Sharing** > **Share Layer** from the dropdown menu.

   A new channel appears in the channels list. Selecting the channel displays the shared layer in the layer stack.
2. To share a layer in a different layer stack:
   - use **Sharing** > **Share Layer** (as outlined above) to create a shared layer, and then drag that to the stack where you need it, or
   - use modifier keys and dragging as outlined in Dragging Layers for Moving, Copying, and Sharing.
3. To share a layer, or layers, as a new channel:
   - with the layer(s) selected, from the **Layers** menu, select **Sharing** > **Share Layers As Channel**, or
   - right-click on the layer and select **Sharing** > **Share Layers As Channel** from the dropdown menu.

Note: For layers shared as new channels, you can make the new channel your current channel by selecting **Sharing** > **Make Shared Channel Current** from either of the menus above.

4. To unshare a layer:
   - with the shared layer select, from the **Layers** menu, select **Sharing** > **Unshare Layer**, or
   - right-click the shared layer and select **Sharing** > **Unshare Layer** from the dropdown menu.

Note: A shared layer is denoted by the link icon on the layer.

Tip: With the possibility of multiple shared layers, it can be difficult to keep track of them. Ensure that you rename any relevant layers or channels in order to limit confusion.

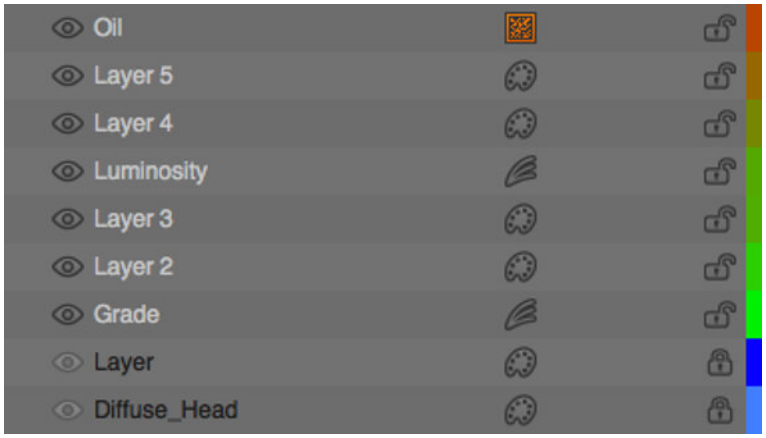
Video: Watch the Sharing Layers in Mari video to learn how to share layers between channels. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Caching and Uncaching Layers

Layers can be cached in order to prevent Mari from slowing down. A large number of layers, or computationally-expensive layers, can lower performance. Fortunately, caching layers can mitigate this.

> **Warning:** Changing colorspace settings of a channel removes any cache from all layers of that channel.

Each layer displays a colored bar to the right of the layer. This tells you how expensive the layer is and its cached state. Inexpensive layers are green, cached layers are light blue, the top layer when using **Cache Up To Here** is dark blue, and as the layers become more complex or higher up in the layer stack they progress to yellow, orange, and red.



When a layer has been cached, that layer is locked. Any change you may wish to make requires you to unlock the layer. For more information about locking other selected items, see Hiding and Locking.

1. To cache a layer, or layers:
   - with the layer(s) selected, from the **Layers** menu, select **Caching** > **Cache Layers**, or
   - right-click on the layer and select **Caching** > **Cache Layers** from the dropdown menu.
2. To cache all layers up to a point in the layer stack:
   - with the layer selected, from the **Layers** menu, select **Caching** > **Cache Up To Here**, or
   - right-click on the layer you want to cache up to and select **Caching** > **Cache Up To Here** from the dropdown menu.

   Like caching a single layer, all layers cached using **Cache Up To Here** are locked and cannot be edited until uncached.

Note: **Cache Layers** and **Cache Up To Here** both cache material layers across all siblings channels. See Managing Materials in Mari for more information on materials.

Tip: You can also combine these two functions in order to work more quickly on your layer stack.

3. To uncache a layer, or layers:
   - with the layer selected, from the **Layers** menu, select **Caching** > **Uncache Layers**,
   - right-click on the layer and select **Caching** > **Uncache Layers** from the dropdown menu, or
   - click the locked padlock icon .

   The cached layer is unlocked and uncached.
4. To uncache all layers that have been locked with **Cache Up To Here**:
   - with the layer selected, from the **Layers** menu, select **Uncache Up To Here**,
   - right-click on the layer you locked with **Cache Up To Here**, and select **Uncache Up To Here** from the dropdown menu, or
   - click the locked padlock icon .

   The cached layer stack is unlocked and uncached.

Note: If you have used the **Cache Up To Here** option for your caching, you can only right-click on the top layer in the cached stack to uncache. This layer is indicated by a blue cache bar.

Alternatively, you can uncache and unlock a layer by clicking on the padlock icon . This uncaches and unlocks all the cached layers locked by **Cache Up To Here**.

If you want to try caching and uncaching layers while using both the **Cache Layers** and **Cache Up To Here** options, try the following:

- **Cache Up To Here** in your layer stack when, for instance, you need to increase the frame rate while working on a large layer stack.
  If you want to make changes to a layer under the **Cache Up to Here** layer but want to keep your frame rate low, **Cache Layers** on any computationally expensive layers around the layer you want to edit.
- **Uncache Up To Here** on your original, top layer in the cached layer stack.
  You can now edit just that one layer, while still having some of the expensive layers around it cached to prevent your performance suffering.

Article: Take a look at the Caching Layers to Improve Performance and Cache colours and their meanings in Mari articles for more information.

Exporting and Importing Layers or Masks

You can choose to either export layers or masks normally or export them in a flattened state. If you export layers or masks normally, Mari exports the selected layer or mask to an external location, as one of the supported file formats.

If you export layers or masks flattened, Mari flattens the selected item(s) into a single layer or mask and exports this to an external location. This does not affect the layers or masks in your current project, only the exported file.

Importing files into Mari to use in the **Layers** palette lets you import the file(s) into an existing layer. This takes the saved texture and imports it over any textures in the existing layer, thus overwriting the pre-existing textures. Alternatively, you can import the file(s) into the layer stack as a new layer. This automatically adds the new layer to the top of the stack, but you can move it to anywhere in the stack that you want.

Exporting Layers

To export the current selected layer:

1. From the **Layers** menu, select **Export** > **Export Selected Layers**, or right-click the layer(s) in the **Layers** palette and select **Export** > **Export Selected Layers** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your textures for the selected layer(s).

To export all the layers in the **Layers** palette:

1. From the **Layers** menu, select **Export** > **Export All Layers**, or right-click in the **Layers** palette and select **Export** > **Export All Layers** from the dropdown menu.

   The **Export** dialog displays.

2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your textures for the layer stack.

Exporting Masks

To export the mask on the current selected layer:

1. From the **Layers** menu, select **Export** > **Export Selected Masks**, or right-click the layer(s) in the **Layers** palette and select **Export** > **Export Selected Masks** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your textures for the selected layer(s).

To export all the masks in the **Layers** palette:

1. From the **Layers** menu, select **Export** > **Export All Masks**, or right-click in the **Layers** palette and select **Export** > **Export All Masks** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your textures for the layer stack.

Exporting Flattened Layers

To flatten and export the currently selected layers:

1. From the **Layers** menu, select **Export Flattened** > **Export Selected Layers Flattened**, or right-click the layer(s) in the **Layers** palette and select **Export Flattened** > **Export Selected Layers Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your flattened selected layers.

To flatten and export all the layers in a layer stack:

1. From the **Layers** menu, select **Export Flattened** > **Export All Layers Flattened**, or right-click in the **Layers** palette and select **Export Flattened** > **Export All Layers Flattened** from the dropdown menu.

   The **Export** dialog displays.

2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your flattened layer stack.

Exporting Flattened Masks

To flatten and export the masks of the currently selected layers:

1. From the **Layers** menu, select **Export Flattened** > **Export Selected Masks Flattened**, or right-click the layer(s) in the **Layers** palette and select **Export Flattened** > **Export Selected Masks Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your flattened masks from the selected layers.

To flatten and export all the masks in a layer stack:

1. From the **Layers** menu, select **Export Flattened** > **Export All Masks Flattened**, or right-click in the **Layers** palette and select **Export Flattened** > **Export All Masks Flattened** from the dropdown menu.

   The **Export** dialog displays.
2. Follow the steps in Exporting Textures from Layers before clicking **Export All Patches**.

   Mari exports your flattened masks from the layer stack.

Importing Layers

To import textures into the currently selected layer:

1. From the **Layers** menu, select **Import** > **Import into Current Layer**, or right-click the selected layer in the **Layers** palette and select **Import** > **Import into Current Layer** from the dropdown menu.

   The **Import** dialog displays.
2. Follow the steps in Importing Textures to Layers before clicking **Import All Patches**.

   Mari imports the textures into the currently selected layer, overwriting any other textures in the layer.

To import textures into the layer stack:

1. From the **Layers** menu, select **Import** > **Import into New Layer**, or right-click in the **Layers** palette and select **Import** > **Import into New Layer** from the dropdown menu.

   The **Import** dialog displays.

2. Follow the steps in Importing Textures to Layers before clicking **Import All Patches**.

   Mari imports the textures into the layer stack as a new layer that appears at the top of the stack.

> **Note:** Importing a PSD file into a layer stack potentially may display a dialog regarding layers of the same name. If layers within the **.psd** have the same name as layers within the layer stack, Mari asks if you want to **Update**, **Skip**, or **Create New**. These options allow you to either overwrite the named layer, skip over the layer during import, or create a new layer, for example **New diffuse** if the original layer name was **diffuse**.

> **Note:** Take care when editing an exported Mari **.psd** file in Photoshop. Due to the way Mari handles mask stacks and adjustment stacks, a layer's sub-stack could be affected by where new layers are inserted in a layer stack.
>
> For instance, mask or adjustment stacks on an existing layer are shown as separate groups from the layer in Photoshop. Inserting a new layer between the group and the original layer before importing the file back into Mari confuses the layer stack order, causing the mask or adjustment stack to be made a component of the new layer and not the original.

Converting Masks

1. Select the layer with the mask you want to convert.
2. Either:
   - right-click on the layer in the **Layers** palette and select **Layer Mask** > **Convert Mask** > <*texel size*>, or
   - with the layer selected, from the **Layers** menu, select **Layer Mask** > **Convert Mask** > <*texel size*>.

   Mari converts the mask to the texel size that you selected, much as it would if you converted the texel size of a channel.

> **Note:** The options for the texel size conversion are **8bit (Byte)**, **16bit (Half)**, and **32bit (Float)**.

Converting Alpha Values to a Mask

1. Select the layer on which you want to create a mask.

2. Either:
   - right-click on a layer in the **Layers** palette and select **Layer Mask** > **Add Mask** > **From Alpha**, or
   - with the layer selected, from the **Layers** menu, select **Layer Mask** > **Add Mask** > **From Alpha**.

   Mari adds a mask to the layer. The mask is black and white, with the color at each pixel taken from the alpha value of the corresponding pixel in the original layer (regardless of the actual color of the original pixel).

Tip: To really see the effects of this, your channel should have some transparent areas. If your channel is completely painted, the resulting mask is just white. If necessary, use the **Paint** tool in **Clear** mode to erase some of the paint off your channel.

Transferring Layers

You can transfer existing layers from the layers on one object or object version to another. This is useful for re-creating work on an object with minimal changes and similar topology or an object with the same topology but a different UV layout. By using the old textures transferred across onto the new object, you can save time instead of starting again from scratch.

Mari uses the Bakery to transfer the nodes and paint data to the target object, avoiding the need to manually redo existing work on a new version of the geometry.

Using the **Layer Transfer** dialog, you can specify which object, version, and patches to use as a source, and which of these to use as a destination. In addition, you can select different source and destination channels where these layers come from and select as many, or all, of the layers in that channel as you wish.

You define the **Transfer Method** to specify how the textures are remapped onto the destination object.

In some cases, where the topology of the object might have changed between versions, there may be inconsistencies from the transfer. You can specify a patch **Fill Color** to highlight where the textures didn't transfer to a patch, and then fix only these patches separately after the transfer.

The Texture Transfer enables you to bake textures quickly and move them between assets.

To transfer layers from one object or version to another:

1. Select the layers you want to transfer. You can then access the Transfer Dialog by selecting **Layers > Transfer** or right-clicking to open the context menu and navigate to **Selection > Transfer.**

   The Layer Transfer dialog opens:
2. Adjust the Layer Transfer settings:

   - **Source** - Select the **Object**, **Version**, and **Patches** that the layers can currently be found on.
   - **Destination** - Select the **Object**, **Version**, and **Patches** that the layers are being transferred to.

   - Use the arrow buttons to move the selected layer or layers back and forth between the source and destination layer lists.
   - **Transfer Method**. Specify the method for transferring the texture onto the destination UV map. Choose between:

- **UV Remap From Bakery** - Use the Bakery to generate the UV coordinate mappings from the source object to the destination object.
- **UV Remap From File** - Use an import-like template to remap pre-generated textures based on UV coordinates. For example, if you have generated the UV map beforehand, you can load it here.

- **Bleed Edges** - Enable this option to generate edge bleed on the resulting image set.
- **Fill Color** - Define the initial color to fill the resulting image set with. If an object has extra patches that aren't covered by the transfer, then the fill color specifies whether these extra patches are filled with **Transparent**, **Black**, **White**, **Mid-gray**, **Magenta**, **Foreground**, or **Background**.

3. Click **Ok**.
4. The transfers will then be added to the node graph of the object you are transferring to.

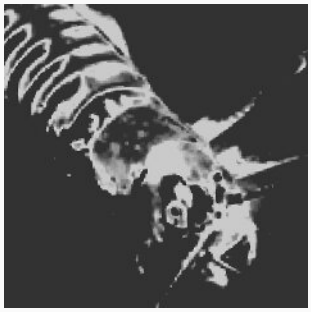
Layer Types and Examples

Below are descriptions and examples of the different adjustments available in the **Layers** palette.

Examples of each of the filters are applied to the following image, for illustrative purposes:

Note: Be sure to check for out-of-bounds values when adding adjustment layers. Ensure that your filters are applied correctly by adding a **Clamp** adjustment to fix any values that don't fall between 0 and 1.

Type	Description	Settings	Example
Adjustment Layers - Filters			
Brightness	Changes the brightness of the paint on the model. You can also change the contrast at the same time. In a layer stack, the **Brightness** filter is always connected to the hidden transparency layer at the bottom of a layer stack if the filter is also at the bottom of the layer stack. To prevent this from disrupting the transparency layer, avoid placing **Brightness** filters at the bottom of a layer stack.	Adjust the **Brightness** using the text field or slider.	
Brightness Lookup	Changes the brightness of the paint on the model. You can adjust the brightness using the curve to get a specific brightness value.	Use the **Map** curve to adjust the specific level of brightness.	
Clamp	Clamps color values to lie within the specified upper and lower values.	Set the **Min** value and **Max** value by using the text field or adjusting the sliders. Select the individual color component to clamp in the **R**, **G**, **B**, and **A** checkboxes.	

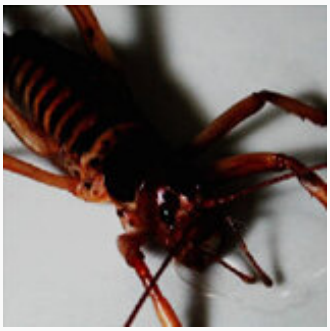
Type	Description	Settings	Example
Color Balance	Adjusts the intensities of the colors on your model in the highlights, midtones, and shadows.	Set whether to **Preserve Luminosity**, then adjust the **Cyan/Red**, **Magenta/Green**, and **Yellow/Blue** values in the **Highlights**, **Shadows**, and **Midtones**.	
Color Lookup	Changes the color of the paint on the model.	Use the curves to set the individual **Red**, **Green**, and **Blue** values.	
Color Switch	Turns individual color channels on or off, as well as the alpha channel.	Use the checkboxes to select which color channels display. When Mari applies the filter, it removes any unchecked color channel or the alpha channel from the painting.	
Color to Mask	Allows a multicolored channel to be used for multiple masks by converting a selected color to a black and white mask.	Select a color using the color swatch, and set the error tolerance to determine how closely the RGB value has to be to the selected color in order to be considered part of the mask.	
Color to Scalar	Converts color RGB values to scalar RGB values, which can	N/A	N/A

Type	Description	Settings	Example
	be useful for using color data in a scalar channel such as Bump.		
Contrast	Changes the contrast levels on the model.	Select the amount of **Contrast** and specify a **Contrast Pivot** point that you want to adjust values around.	
Copy Channel	Copies the value from one RGB color channel to the other two. The result is a grayscale image with the intensity values from the selected channel.	Select the source channel from the list. When you apply the filter, Mari copies the selected channel over the other two channels.	
Flow	Displays an image onto the model's surface and uses vector data to animate it flowing. Note: If painting in the **Clear** blending mode, the **Speed** attribute is removed in that area. The vector inspector directional markers are still shown on the model. This concerns how the **Flow** procedural handles the alpha component.	• **Tile Image** - the image used on the model for flow. • **Repeat** - sets the repetition value of the tiled image. • **Animated** - Enables or disables the animation. • **Time Offset** - controls the flow over time. • **Speed** - controls the sensitivity of **Time Offset**, which affects the speed of flow.	In the example below, the **Flow** procedural would animate the molten liquid so that it appeared to moving across the surface of the texture:

Type	Description	Settings	Example
Gamma	Changes the gamma levels on the model.	Select a particular gamma level using either the text field or the sliders. **Invert** reverses the gamma conversion. That is, a higher setting for **Gamma** results in a more washed-out looking image.	
Grade	Changes the overall color grade of the black and white points, and allows you to adjust the gain and gamma of the paint on the model.	Adjust the **Blackpoint** to allow you to change the depth of the darkest point in the paint. The **Whitepoint** allows you to increase the brightest point in the paint. **Lift**, **Gain**, **Multiply**, and **Gamma** all modify the specified aspects of the paint to change the overall texture; while, **Offset** changes the black and white points as one.	
HSL	Changes the hue, saturation, and lightness of the colors on the model.	Change the hue of the colors by setting the value for **Hue**. Adjusting the slider moves the colors around the color wheel. The rotation value is the degree around the color wheel that each color shifts, between 0 and 360 (adjusts the	

Type	Description	Settings	Example
		slider between -180 and 180). The **Saturation** slider adjusts the intensity of the colors in the image data. Select a multiplier from 0 to 1, where .50 is the original saturation value. You can also set the **Lightness** value the same way.	
HSV	Changes the hue, saturation, and value of the colors on the model.	Change the hue of the colors by setting the value for **Hue**. Adjusting the slider moves the colors around the color wheel. The rotation value is the degree around the color wheel that each color shifts (adjusts the slider between 0 and 1). The **Saturation** slider adjusts the intensity of the colors in the image data. Select a multiplier from 0 to 10, where 1.00 is the original saturation value. You can also set the **Value** the same way.	

Type	Description	Settings	Example
Height As Normal	Interprets the texture as height data, then computes the normals from the height map and converts the normals to a **Diffuse** RGB color. This is useful for baking normal maps from height maps.	**Bump Weight** adjusts how much bump is applied. Lower values are smaller; higher values produce larger and more obvious displacement. **Bump Mode** displays the bump quickly (**Fast**) or accurately (**Accurate**). **Bump Space UV** mode - Normals are calculated relative to the UV coordinates space. The **UV Bump Space** mode is the more traditional approach but can lead to noticeable seams and color variations. **Bump Space World** mode - Normals are calculated relative to the World coordinates space. Using the **World Bump Space** mode gives a more seamless result.	
Hue Shift	Changes the hue of the colors on the model. Set the value for **Hue** by adjusting the slider. This moves the colors around the color wheel.	Select the overall hue. The rotation value for the color wheel is the degree around the color wheel that each color shifts, between -180 and 180.	

Type	Description	Settings	Example
Invert	Inverts the colors or the alpha in the layers lower in the layer stack. Invert replaces colors with their "opposite" in the color chart. For example, a color with a float value of 0.3 is replaced by a color with a float value 0.7.	Set whether to invert the **R**, **G**, **B**, or **A** channels individually.	
Levels	Changes the color levels in the paint. You can change the white, midtone and black points for each color channel or all together.	Set the: • Color Component - to adjust the specific color channel (**Red**, **Green**, or **Blue**). • **WhiteOut** - how strongly to output the white in the filtered image. Higher values show the entire white output; lower values show decreasing amounts. • **BlackOut** - how strongly to output the black in the filtered image. Higher values show the entire black output; lower values show decreasing amounts. • **White** - the upper limit of displayed color intensity. Areas with a value higher than this are mapped to 1 (white).	

Type	Description	Settings	Example
		• **Mid** - the middle point between white and black. Mari remaps the values so that this is the middle of the range between the white point and black point. Moving this pushes the values towards that end of the spectrum. • **Black** - the lower limit of displayed color intensity. Areas with a value lower than this are mapped to 0 (black).	
Luminosity	Outputs just the luminosity value of each pixel. That is, it outputs a grayscale image based on the brightness of each of the pixels in the original image. Layer masking, projection masks, and masks on projectors are all hardwired to use the red color component. If you want to use another component for your mask value, the **Luminosity** adjustment layer allows you to convert an RGB input to a grayscale output.	None.	

Type	Description	Settings	Example
Premultiply Alpha	Either pre- or post-multiplies the alpha in the selected image. If you are painting using an image without premultiplied alpha onto one with it, use this filter to perform the premultiplication, so the images match and you avoid lines around the outside of the patch. Postmultiply works the same, but in reverse: it removes premultiplication to match images that do not have premultiplied alpha.	If the **Unpremultiply** box is checked, this acts in reverse (that is, removes premultiplied alpha).	
Saturation	Changes the intensity of the colors in the image data.	The **Saturation** slider adjusts the intensity of the colors in the image data. Select a multiplier from 0 to 10, where 1.00 is the original saturation value.	
Scalar to Color	Converts scalar RGB values to color RGB values, which can be useful for blending scalar data, such as a roughness map, into a color channel.	N/A	N/A
Scale	Directly modifies the overall scale of the color values for each channel individually.	Adjusting the sliders for **R**, **G**, **B**, and **A** allows you to specify the exact level of scale for each channel, between 0 and 2.	

Type	Description	Settings	Example
Set Value	Changes the values of specific RGBA channels.	Selecting **Red**, **Green**, **Blue**, or **Alpha** from the **Channel** dropdown menu allows you to adjust the slider (from 0-1) to change the value of the specific color channel.	
Shuffle	Shuffles the RGBA channels so that any of them can be replaced by another color channel. Layer masking, projection masks, and masks on projectors are all hardwired to use the red color component. If you want to use another color component for your mask value, the **Shuffle** adjustment layer allows you to change the RGBA components so that any can be replaced by another.	Change the channels of **R**, **G**, **B**, or **A** to affect the overall color of the paint on the model.	
Tangent To Screen	Sets up a **Tangent To Screen** adjustment to be used for Vector Brushing. This adjustment is automatically created when creating default vector shaders.	Suppress the blue value from the color-encoded vectors in screen space by ticking the **Suppress Blue** checkbox.	

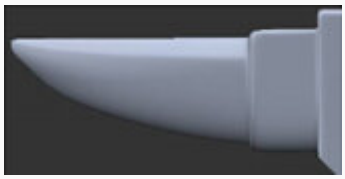
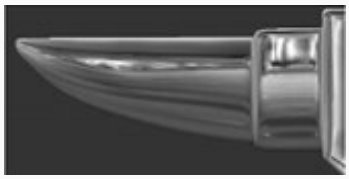
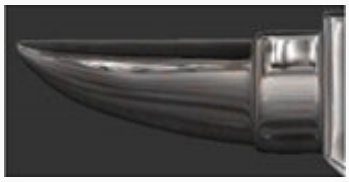
Type	Description	Settings	Example
Tangent To World	Sets up a **Tangent To World** adjustment to be used for Vector Brushing. This adjustment is automatically created when creating default vector shaders.	Suppress the blue value from the color-encoded vectors in the world space by ticking the **Suppress Blue** checkbox.	
World To Tangent	Sets up a **World To Tangent** adjustment so that you can convert normals from world space into tangent space normals.	Suppress the blue value from the color-encoded vectors in the tangent space by ticking **Suppress Blue** checkbox. Choose whether to flip the Y axis or leave the normals as they are on **Flip Y**.	
sRGB2Linear	Applies an sRGB to linear colorspace conversion.	Checking **Invert** applies a linear to sRGB colorspace conversion.	

Below are descriptions and examples of the different procedurals available in the **Layers** palette.

Examples of each of the procedurals are applied to the following image, for illustrative purposes:

Note: In Procedurals that include color attributes, the color controls contain a toggle that sets the context in which the procedural effect is to be used for Scalar or Color data. By default, Mari detects the channel's color data type and sets the toggle accordingly for layers created through the **Layers** palette. For Procedural nodes created through the Node graph, this color/scalar toggle defaults to color. Procedural nodes that are to be used within masks and scalar channels have to be set to scalar manually. See Color Data and Scalar Data for more information.

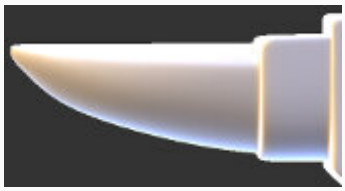
Type	Description	Settings	Example
Procedural Layers - Basic			
Color	Applies the selected color to the rest of the paint on the model.	Changing the **Blend Mode** and **Amount** allows you to adjust how the color is applied, and how strongly, to the model. **Color** allows you to select the color from the HSV swatch palette.	Procedural color set to burgundy: Color procedural (**Screen** blend):
Constant	Generates a constant number across **RGB** or **RGBA** components that may be relied upon for mathematical operations.	The **Constant** slider ranges from 0 to 10,000, and defaults to 1.000 when reset. **Components** may be set to either **RGB** or **RGBA**.	Constant procedural (**Multiply** blend):
Vector	Applies RGBA values to a vector (XYZW) map to produce a specific diffuse color that represents a	The **X**, **Y**, **Z**, and **W** fields correspond to R, G, B, and A respectively to give a specific vector in a diffuse image.	Vector procedural with XYZW values adjusted:

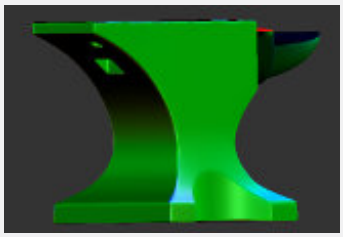
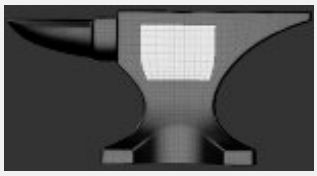
Type	Description	Settings	Example
	vector field.		 Vector procedural (**Multiply** blend):
Procedural Layers - Projection			
Cube Map	Simulates a shiny reflective surface using a cubic environment map as the input. This layer cannot be cached or baked, and is for display purposes only.	• **Cubic Image** - the location of the cube map. • **Falloff Start** - controls where the cubic image starts. • **Falloff End** - controls where the cubic image ends.	Cube Map procedural (**Luminance** blend):  Cube Map procedural (**Luminance** blend) over layer: 
Sphere Map	Simulates a shiny reflective surface using a spherical environment map as the input. This layer cannot be cached or	• **Sphere Map** - the location of the environment map. • **Falloff Start** - controls where the spherical image starts. • **Falloff End** - controls where the spherical image ends.	Sphere Map procedural (**Luminance** blend):

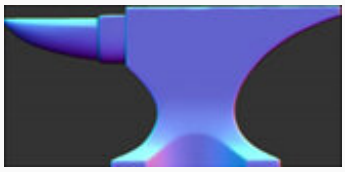
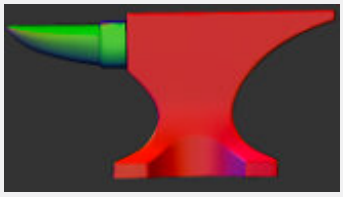
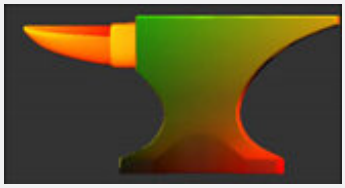
Type	Description	Settings	Example
	baked, and is for display purposes only.		 Sphere Map procedural (**Luminance** blend) over layer:
Cube Map Projector	Uses a cubic image (in a **.dds** file format) as a texture to project onto the model. This layer cannot be cached or baked, and is for display purposes only. **Note:** When saving **.dds** files in other applications, they should be encoded as 8.8.8.8 ARGB files to avoid any complications when bringing them into Mari.	• **Cubic Image** - the filepath to the cubic image to use for projection. • **Cull Backfaces** - when the checkbox is ticked, it ensures that projection does not affect areas facing away from the camera. • Offset **X**, **Y**, and **Z** - the offset of the projection image on the X, Y, and Z axes. • Rotation **X**, **Y**, and **Z** - the rotation of the projection image on the X, Y, and Z axes. • Edge Falloff **Start** and **End** - modifies how far away the falloff starts or ends. • Edge Falloff **Curve** - allows you to adjust the otherwise linear shape of the falloff to your desired shape. • Distance Falloff **Start** and **End** -	

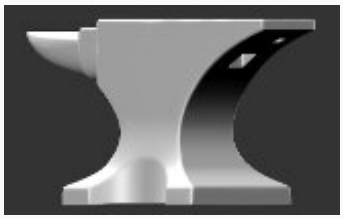
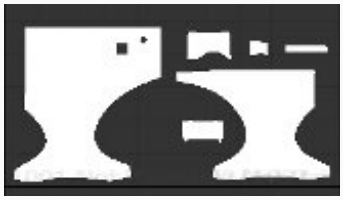
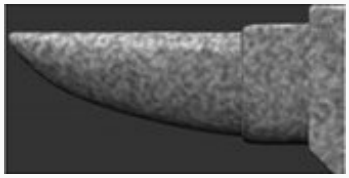
Type	Description	Settings	Example
		adjust the start and end radial distance from the camera. • From 0-100, where 100 represents straight-on projection. • Distance Falloff **Curve** - allows you to adjust the otherwise linear shape of the falloff to your desired shape.	
Sphere Map Projector	Uses a spherical image of any file type as a texture to project onto the model. This layer cannot be cached or baked, and is for display purposes only.	• **Cull Backfaces** - when the checkbox is ticked, it ensures that projection does not affect areas facing away from the camera. • Spherical Image - the filepath to the spherical image to use for projection. • Offset **X**, **Y**, and **Z** - the offset of the projection image on the X, Y, and Z axes. • Rotation **X**, **Y**, and **Z** - the rotation of the projection image on the X, Y, and Z axes. • Edge Falloff **Start** and **End** - modifies how far away the falloff starts or ends. • Edge Falloff **Curve** - allows you to adjust the otherwise linear shape of the falloff to your desired shape. • Distance Falloff **Start** and **End** - adjust the start and end radial distance from the camera. • From 0-100, where 100 represents straight-on	

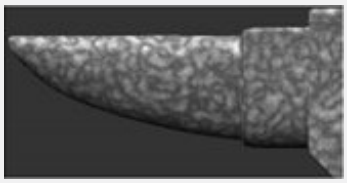
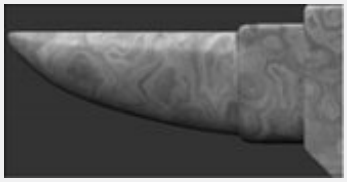
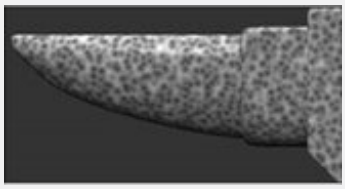
Type	Description	Settings	Example
		projection. • Distance Falloff **Curve** - allows you to adjust the otherwise linear shape of the falloff to your desired shape.	
Triplanar Projection	Uses three different images to project onto the surface of your model from three different directions (Top, Front, and Right). Each of the parameters listed (other than **World Scale**) are for each of these three directions.	• **World Scale** - overall scale control for the image projection. • **Image** - the images you want to project onto each of the three directions. • **Repeat** - frequency that the image is repeated across your model. • **Angle** - the rotation angle of the image on your model. • **U Offset** - how much the image on the model is offset by on the U axis. • **V Offset** - how much the image on the model is offset by on the V axis. • **U Scale** - how much the image on the model is stretched or contracted on the U axis. • **V Scale** - how much the image on the model is stretched or contracted on the V axis. • **Falloff Start** - controls where the image projection starts on the model. • **Falloff End** - controls where the image projection ends on the model.	

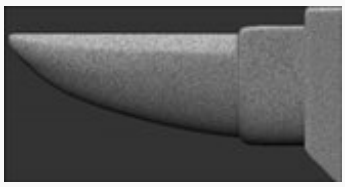
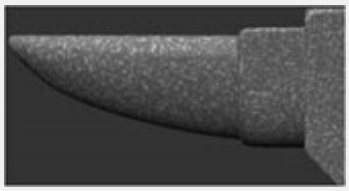
Type	Description	Settings	Example
		• **Falloff** - controls the falloff of the image projection between the start and end values.	
Procedural Layers - Lighting			
Environment Light	Displays the environment light reflection, if the **Image** in the **Environment** light in the **Lights** palette has been set. This layer cannot be cached or baked, and is for display purposes only.	• **Sharpness** - adjusts the sharpness of the reflections on the model, with higher values increasing the sharpness and lower values decreasing it.	Environment Light procedural (using one of the preset environments shipped with Mari): 
Procedural Layers - Geometry			
Ambient Occlusion	Applies an ambient occlusion shading to the surface. This overrides Mari's default ambient occlusion shader, allowing you to change the details. For example, you can edit the channel to use as an ambient occlusion map. If you want to save the default ambient occlusion data for the project as a channel, you can bake the	None. **Note:** This overrides the ambient occlusion shading Mari applies by default. This means that if you just bake out the ambient occlusion and then use this as the input to this shader, you see no difference (you're applying the ambient occlusion that Mari would apply anyway). Try changing the ambient occlusion value under the Lighting component of the shader, or edit the ambient occlusion mask channel.	Ambient Occlusion procedural:

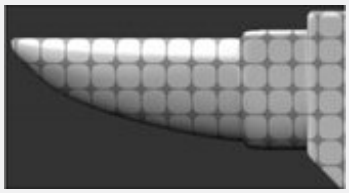
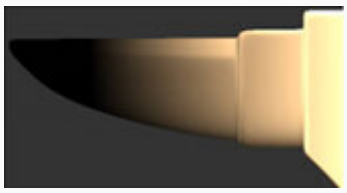
Type	Description	Settings	Example
	ambient occlusion from the default shader. You can then edit this baked channel as you want.		
BiTangent	Outputs bitangent vectors to, along with tangent and normal vectors, create a TBN matrix that can map from world space to tangent space (or the other way around).	None.	BiTangent procedural: 
Position	Displays the graphic representation of the surface positions (XYZ as RGB).	None.	Position procedural:
Selection Mask	Outputs white (1.0) color values for masked regions and black (0.0) color values for unmasked regions, based on the current selection in the selection mode. It is best to convert the procedural to a paintable layer once you are ready to use it.	None. Note: To create a selection mask, you must be using the **Select** tool, not the **Marquee Select** tool, in **Face** mode.	Selection Mask procedural: 
Surface Normal	Converts the present surface normals to a **Diffuse** RGB color.	None.	Surface Normal procedural:

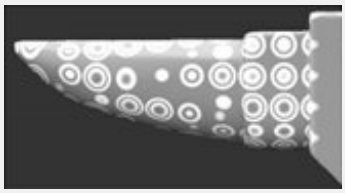
Type	Description	Settings	Example
	This is useful for baking out normal maps from the bump channel.		
Tangent	Outputs tangent vectors to, along with bitangent and normal vectors, create a TBN matrix that can map from world space to tangent space (or the other way around).	None.	Tangent procedural: 
UDIM Mask	Masks out all other UDIMs on a model, except for the selected UDIM.	The **UDIM** selector allows you to select which UDIM you can paint on and masks out all others.	UDIM Mask procedural:
UV	Converts the UV to a color, based on the UDIM values. Useful for debugging UV layouts.	None.	UV procedural:  UV procedural in UV view:
UV Mask	Outputs white (1.0) color values for	None.	UV Mask procedural:

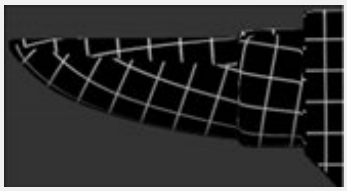
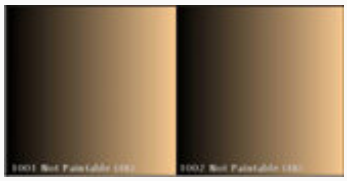
Type	Description	Settings	Example
	regions where there is some geometry and black (0.0) for regions where there is no geometry. When baked to a texture by **Convert to Paintable**, the layer becomes a UV mask texture.		 UV Mask procedural in UV view: 
Procedural Layers - Fractal			
Cloud	Blends seamless 3D noise in a cloud pattern with the shaded surface, using the specified **Blend Mode**. **Note:** If you are accustomed to using **Fractal Noise** from previous versions of Mari, you can still affect fractal properties such as **Contrast** by adding a **Contrast** adjustment to the **Cloud** procedural.	You can control the look of the cloud pattern by adjusting the **Size** and **Roughness** of the pattern, as well as the pattern offset for the X, Y, and Z values. **Color A** and **Color B** allows you to set the cloud pattern colors used on the model.	Cloud procedural:  Cloud procedural over layer:
Turbulence	Blends seamless 3D noise in a turbulence pattern with the	You can control the look of the turbulence pattern by adjusting the **Size** and **Roughness**. **Color**	Turbulence procedural:

Type	Description	Settings	Example
	shaded surface, using the specified **Blend Mode**.	**A** and **Color B** allows you to set the turbulence pattern colors used on the model.	 Turbulence procedural over layer:
Procedural Layers - Misc			
Oil	Blends seamless 3D noise in an oil pattern with the shaded surface, using the specified **Blend Mode**.	You can control the look of the oil pattern by adjusting the **Size** and the **Organicness** or **Roughness** qualities. **Color A** and **Color B** allows you to set the oil pattern colors used on the model.	Oil procedural:  Oil procedural over layer:
Procedural Layers - Noise			
Cellular	Blends seamless 3D noise in a cellular pattern with the shaded surface, using the specified **Blend Mode**.	You can control the look of the cellular pattern by adjusting the **Size** and **Type** of cells to use. Additionally, you can specify the **Distance method** used to achieve a specific look. **Color A** and **Color B** allows you to set	Cellular procedural: 

Type	Description	Settings	Example
		the cellular pattern colors used on the model.	Cellular procedural over layer:
Perlin	Blends seamless 3D noise in a perlin pattern with the shaded surface, using the specified **Blend Mode**.	You can control the look of the perlin pattern by adjusting the **Size** and color. **Color A** and **Color B** allows you to set the perlin pattern colors used on the model.	Perlin procedural:  Perlin procedural over layer:
Squiggle	Blends seamless 3D noise in a squiggle pattern with the shaded surface, using the specified **Blend Mode**.	You can control the look of the squiggle pattern by adjusting the **Size** and color. **Color A** and **Color B** allows you to set the squiggle pattern colors used on the model.	Squiggle procedural:  Squiggle procedural over layer:

Type	Description	Settings	Example
Procedural Layers - Pattern			
Cube	Blends a seamless cubic pattern with the shaded surface, using the specified **Blend Mode**.	You can control the look of the cubes by adjusting the **Repeat** pattern and **Gap** spacing. **Falloff** and **Roundness** give you even more aspects to fine-tune the pattern. **Color A** and **Color B** allows you to set the cube pattern colors used on the model.	Cube procedural:  Cube procedural over layer:
Object Space Linear Gradient	A procedural version of the **Gradient** tool. This behaves the same as **Gradient** without painting the gradient onto the model.	Adjust the gradient **Start** and **End** points to determine where **Color A** and **Color B** are on the object. Use the **Map** to fine-tune the gradient as a color curve. Rotation **X**, **Y**, and **Z** points adjust the gradient along the X, Y, or Z axes.	Object Space Linear Gradient procedural:  Object Space Linear Gradient procedural over layer:
Sphere	Blends a seamless spherical pattern with the shaded surface, using the specified	You can control the look of the spheres by adjusting the **Size** and **Falloff** of the pattern. **Color A** and **Color B** allows you to set	Sphere procedural:

Type	Description	Settings	Example
	Blend Mode.	the sphere pattern colors used on the model.	 Sphere procedural over layer:
Tiled	Lets you select an image to tile across the surface of your model. This layer is disabled for Ptex channels.	• **Tile Image** - the location of the image you want to tile across the surface of your model. • **Rotation Angle (Degrees)** - the angle (in degrees) that the image is rotated by. • **U Offset** - how much the image is offset by on the U axis. • **V Offset** - how much the image is offset by on the V axis. • **U Repeat** - the frequency at which the image is repeated on the U axis. • **V Repeat** - the frequency at which the image is repeated on the V axis. • **Mirror U** - check to mirror the image on the U axis. • **Mirror V -** check to mirror the image on the V axis.	Tiled image: Tiled Image over layer:
UV Grid	Blends a seamless UV	You can control the look of the	UV Grid procedural:

Type	Description	Settings	Example
	grid pattern with the shaded surface, using the specified **Blend Mode**.	UV grid by adjusting the **Color** and **Thickness** of the pattern, as well as the level of **Repeat.**	 UV Grid procedural over layer:
UV Linear Gradient	A procedural version of the **Gradient** tool for use in UV space. Behaves the same as **Gradient** without painting the gradient onto the model.	Adjust the gradient **Start** and **End** points to determine where **Color A** and **Color B** are on the object. Use the **Map** to fine-tune the gradient as a color curve. Adjust the **Angle** of the gradient and how much you want the gradient to be allowed to **Repeat**. Repeating the gradient does so by tiling.	UV Linear Gradient in the UV tab (without UV image rendering):  UV Linear Gradient in the UV tab (with UV image rendering):
Procedural Layers - Plant			
Wood	Blends seamless 3D noise in a wood grain pattern with the shaded surface, using	You can control the look of the wood pattern by adjusting the **Size** of the pattern and of the bands, as well as the **Roughness**	Wood procedural:

Type	Description	Settings	Example
	the specified **Blend Mode**.	of the pattern. The offset can be adjusted in **X**, **Y**, and **Z** space. Additionally, **Color A** and **Color B** allows you to set the wood pattern colors used on the model.	Wood procedural over layer:
Procedural Layers - Smart Masks			
Color Procedural	The available color procedurals from the **Shelf** appear in this list. See Smart Mask Node for more information.		
Scalar Procedural	The available scalar procedurals from the **Shelf** appear in this list. See Smart Mask Node for more information.		
Procedural Mask	The available procedural masks from the **Shelf** appear in this list. See Smart Mask Node for more information.		

Shaders

Shaders control how Mari displays the model on the canvas under certain, user-specified lighting conditions. Mari uses shaders to specify how different channels behave in the lighting module and how the paint on the model reacts to light. Mari *always* uses a shader to generate the surface displayed on screen.

The shaders are also highly dependent on the channels and layers in a project, and you can view current paint layers, targets, and channels specifically in the **Shaders** palette.

Mari's default shaders show the paint in either the **Current Channel**, **Current Layer and Below**, **Current Layer**, or **Current Paint Target**. You can also create your own custom shaders, called 'User Shaders', by combining diffuse and specular shader types then setting up channels in the shader component inputs to specify how they appear in the shader.

For more information on how shaders, channels, and layers integrate with one another, please refer to the Layers section in this guide.

Video: Watch Understanding Shaders and Channels for a brief overview about Shaders. This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Getting Started with Shaders

By default, when you create a project and do not make changes to the **Channels** tab, a principled BRDF shader is created with 4 channels. See Managing Projects. The default channels get plugged into the shader immediately to set up your project. You can create and assign additional channels to the fields in the shader components and build up complex textures on your model.

Color channel, plus...

displacement channel, plus...

specular channel, equals...

...this. Here the shader is pulling in all three channels to make a surface that gives a preview of what the final render may look like.

This lets you, for example, build a shader so you can paint onto the displacement or specular channels while getting a real-time view of how your changes affect the final render.

Tip: Shaders *only* control how Mari displays the model's surface. They do not affect the contents of the channels in your project.

Shaders include:

- basic lighting details, including ambient, diffuse, and specular light intensities.
- a method for quickly viewing paint in your current channel, layer, layer stack, or on your current paint target.

Mari generates the shaded surface by going through the components in the current shader. Each component takes an input from a channel and is adjusted according to relevant component sliders. These components are then added to the current surface. The final shaded surface as you see it on screen is the final result of these components.

For example, you could set up a shader by following the steps below:

1. Create the initial shaded surface from the diffuse channel and set this in your shader's **Diffuse Color** component.
2. Display specular or displacement previews using designated channels as the inputs, and adjust the sliders to give the shaded surface the desired look.

3. Adjust ambient light or apply environment images to further build up the preview of your final rendered image.

As with channels, you set up shaders individually for Working with Objects.

Shaders and Lights

In Mari, shaders and lights are related - they both control how your project appears on screen, but neither affects the actual data in the channels.

- Shaders do include some basic lighting information, but mostly they specify what to show on the surface of the model. For example, shaders may include specular information, which describes how the surface interacts with the lights in your project.
- Lights provide control on how your project is lit on the canvas. You can set up four individual lights, each with its own color, intensity, and direction, by adjusting the settings in the **Lights** palette. You can also set an environment light that relies on an image for the lighting.

Once you have set up the shader, you can still opt to change the lighting on the model by changing the lights in the **Project Controls** toolbar, in the **Lights** palette, or in the **Shading** menu.

Shading and Channels

Mari has a few default shaders that show the contents of the current channel, layer, or paint target, you can always switch back to these to check your work in isolation to see which channels are being used in which shader inputs.

> **Tip:** A useful way to keep track of which shader you're editing is to add it to the Heads-Up Display. From the **Edit** menu, select **HUD Manager**. Under **Project**, put a check in the box next to **Current Shader**. Mari now shows the name of the shader you are currently painting as part of the background information on the canvas. For more information, see the Configuring the User Interface section.

Shaders on Multiple Channels

If you want to copy a shader you've created for one object in your project to use on another object, you can use the keyboard shortcuts or the options in the context menu to cut, copy, and paste shaders between objects. Shaders are exclusive to each object, as are channels and layers. The only way to duplicate a shader for use on another object, is to cut or copy and paste it into the **Shaders** palette for each object in your project.

Physically Plausible Shaders

Physically plausible shaders differ from traditional shaders, such as the default Mari shaders, by obeying the law of conservation of energy. Unlike traditional shaders, changing different inputs, such as **Specular** or **Specular Roughness**, does not effect the overall brightness of the point or environment lights. The distribution of the light might have changed to be concentrated in one location or spread out across the shaded surface, but the sum total of the pixel values should come out to the same value.

Ambient Occlusion

Ambient occlusion can be set as a shader component input for all of the custom-created shaders. Using a component input for the ambient occlusion allows you to create your own occlusion map in a separate channel and plugging this channel into the shader input.

If you don't want to create a channel specifically for ambient occlusion, you can opt to calculate the ambient occlusion for each individual object in your project. For more information, please refer to About Masking and Calculating Ambient Occlusion.

Shader Limits

Most shaders do not require texture slots, but if you exceed the amount of texture slots available you may encounter a rendering error. If this occurs, there are a number of causes that may have contributed to the error. To free up texture slots, try:

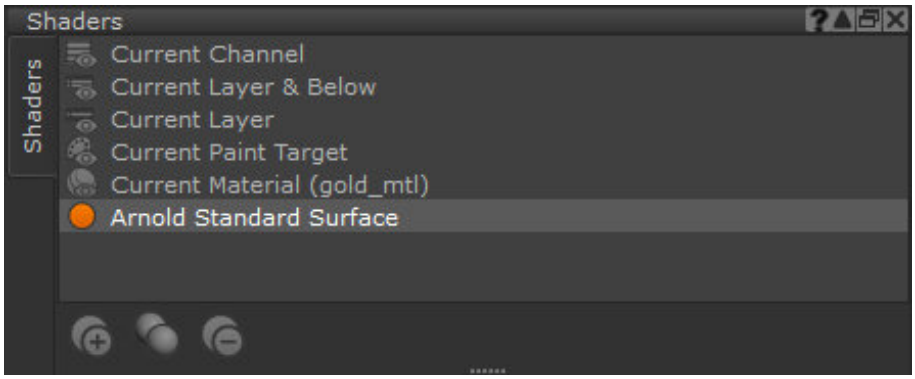
- turning off features that require extra texture units, such as painting masks in the **Painting** palette,
- turning off any filter previews that may be active,
- flattening or caching layers in long or complex layer stacks,
- changing the inputs on the current shader to use fewer texture slots.

Note: Even if there are remaining texture slots available, active features such as depth projection, paint masks, and shadows may not respect the texture slot limit if more expensive shaders components are in use. You can turn off these features to free up more texture slots. See About Masking and Shadow Settings for more information.

Opening the Shaders Palette

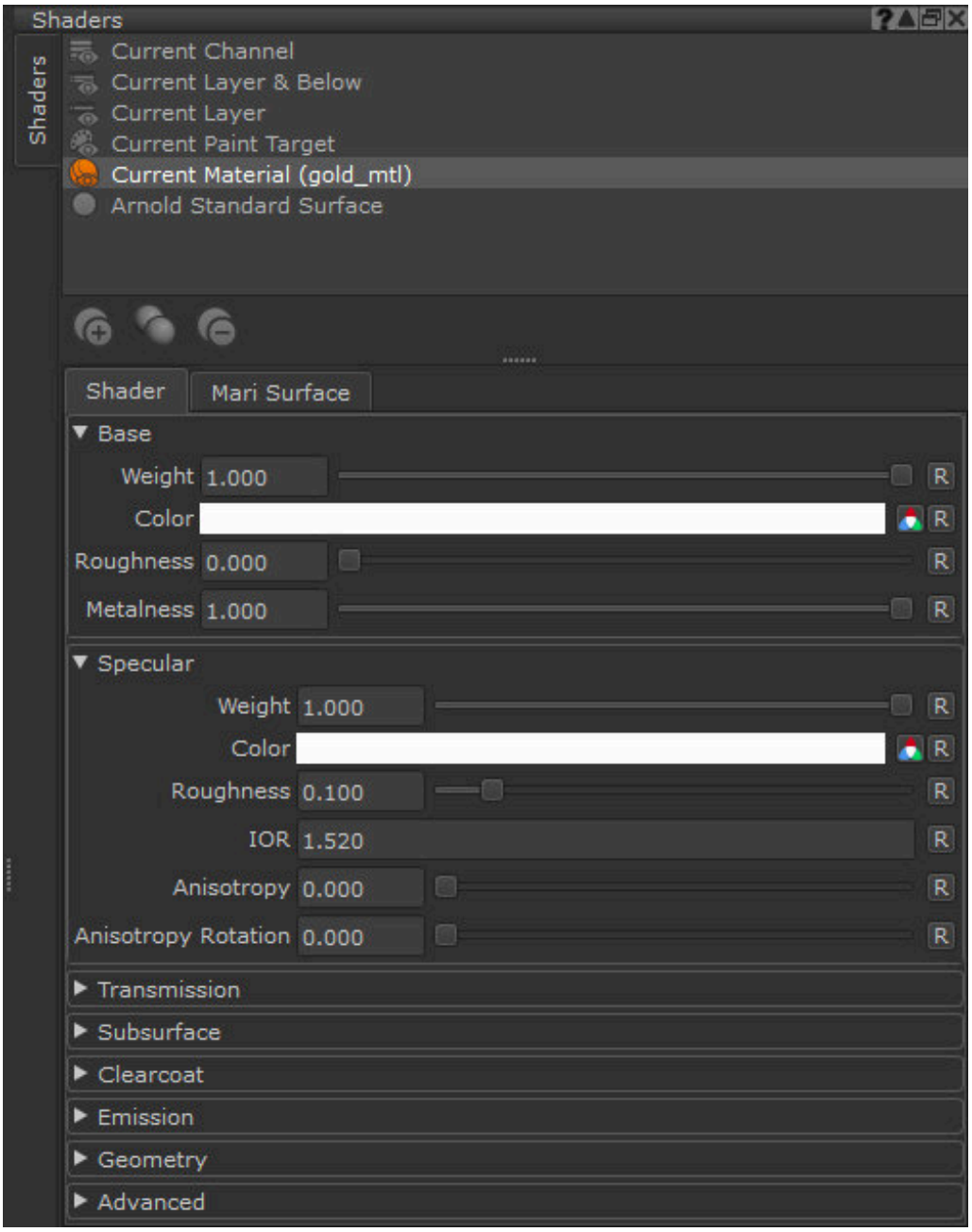
1. Start Mari and open your project.
2. Open the **Shaders** palette:
   - right-click on the toolbar, and select **Shaders** from the dropdown menu, or
   - from the **View** menu, select **Palettes** > **Shaders**.

   The **Shaders** palette displays, showing the default shaders and any shaders you've already set up for your current project.



3. The default shaders in the list are:
   - **Current Channel** - shows only the currently selected channel on the model in the viewport. For example, if you only wanted to see the diffuse color channel on your model, you can select the channel and press **Current Channel**.
   - **Current Layer and Below** - displays the paint in your selected layer or sub-stack. If you have a layer in a mask or adjustment stack selected, the selected layer and all layers below in the sub-stack are displayed in the viewport. If you don't have a layer in a sub-stack selected, all layers up to the currently selected layer are shown.
   - **Current Layer** - shows the selected layer on the model in the viewport.
   - **Current Paint Target** - displays the paint in your selected layer, including any selected mask, layer in a mask or adjustment stack, filter, or group associated with the layer.
   - **Current Material** - shows the currently selected material, node, or group on the model in the viewport. This allows you to isolate your material from the rest of your channel data so you can make edits to your material with a more lightweight shader.

   When you select a shader, the bottom of the palette shows the details, as shown in the example image below.



4. The shader components that are shown in the bottom pane of the palette differ depending on what shader is selected.

Tip: Click the toggle to change the influence of channels connected to the shader without disconnecting them from the shader input. The Displacement shader input is toggled off by default.

Default Shaders

In any project, Mari has five default shaders that you start with before adding any custom shaders. These default shaders are:

- **Current Channel** - displays the paint in your selected channel.
- **Current Layer and Below** - displays the paint in your selected sub-stack (such as a mask or adjustment stack) up to the selected layer. If you don't have a sub-stack selected, it shows the parent stack up to the selected layer.
- **Current Layer** - displays the paintable paint and masks in your selected layer.
- **Current Paint Target** - displays the paint in your selected layer, including any mask or mask stack, adjustment, filter or groups associated with the layer.
- **Current Material** - displays the currently selected material, node, or group on the model in the viewport. This allows you to isolate your material from the rest of your channel data so you can make edits to your material with a more lightweight shader.

These shaders are also found on the **Project Controls** toolbar and allow you to switch between each shader.

Note: If you can't see the **Project Controls** toolbar, right-click on the toolbar area and choose **Project Controls** from the dropdown menu.

Tip: If you want to see how your paint looks for a single layer, then compare it when added to the rest of the layer stack, switching between the **Current Layer** and **Current Layer and Below** views can assist you.

Adjusting the Look of a Material with the Current Material Shader

The Current Material shader is a lightweight solution to viewing and editing materials or multi-channel groups on your models. Instead of viewing your user shader to lookdev materials, which can be an expensive operation, you can isolate a specific material on the model in the viewport.

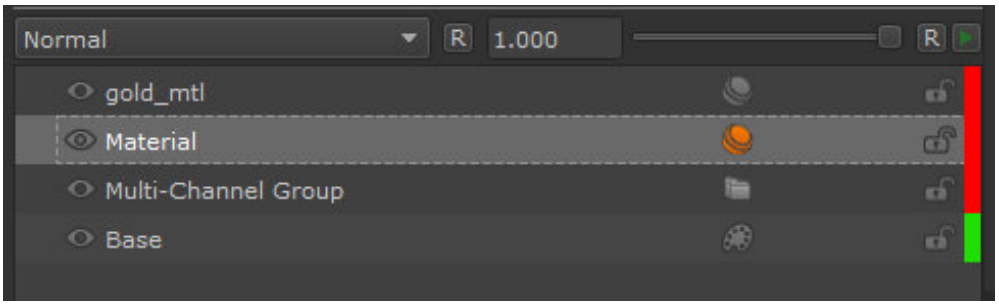
Viewing the helmet green base material with the **Current Material** shader in the viewport.

Viewing the helmet pink plastic material with the **Current Material** shader in the viewport.

Viewing all materials on the model with the **Arnold Standard Surface** shader in the viewport.

To view and edit your material:

1. Select a material or multi-channel group layer in the **Layers** palette.



A material selected in the **Layers** palette.

> **Note:** When a material in the Layers palette is selected, it appears with an active Material icon. Unselected materials appear with an inactive Material icon.

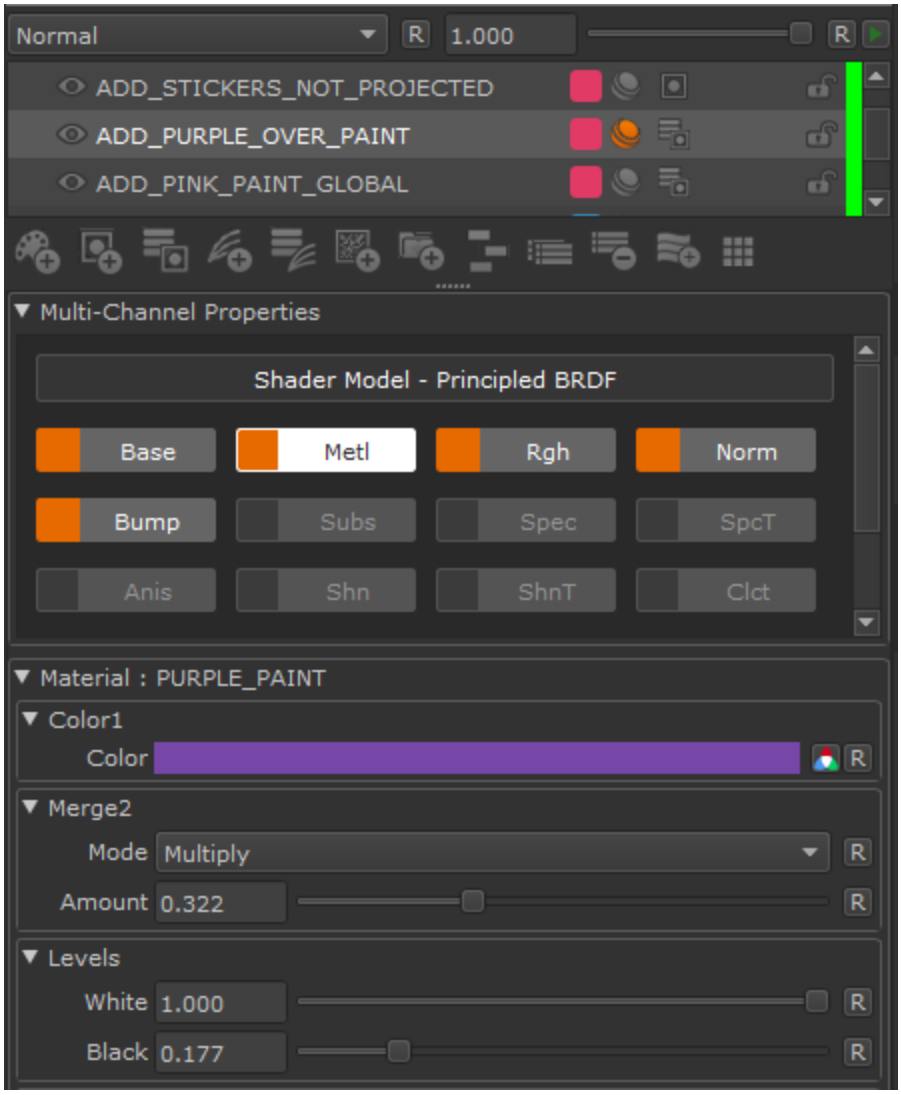
2. Open the **Shaders** palette and select **Current Material**, or
   - click and hold the View Shader button on the **Project Controls** toolbar, and click **Current Material**, or
   - hover over the viewport, press **i** and under the **Shaders** column, select **Current Material**.

   The selected material or multi-channel group is shown in the viewport.

> **Note:** The View Shader button always shows the current mode so the icon may appear differently.

3. Make adjustments to your material with the exposed material properties in the Layers palette.

> **Note:** To learn more about exposing material properties in the layer palette, see Exposing Material Properties Using the Promote Button.



Exposed material properties shown in the Layers palette.

Adjusting the color in the material while viewing the model with the **Current Material** shader.

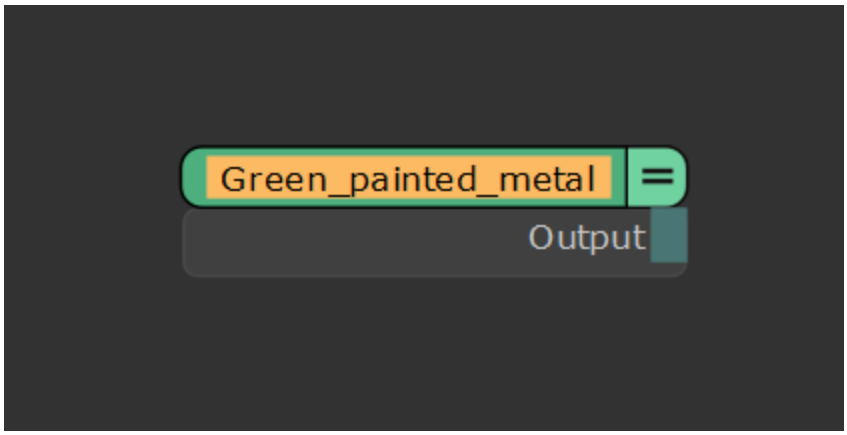
Note: If you switch to a non-material layer while viewing your model with the Current Material shader, the model appears blank in the viewport. Select the **Current Layer** shader to view your currently selected layer.

Tip: You can press the Material icon in the Layers palette to open and edit the material in the Node Graph.

Viewing Multi-Channel Nodes in the Node Graph

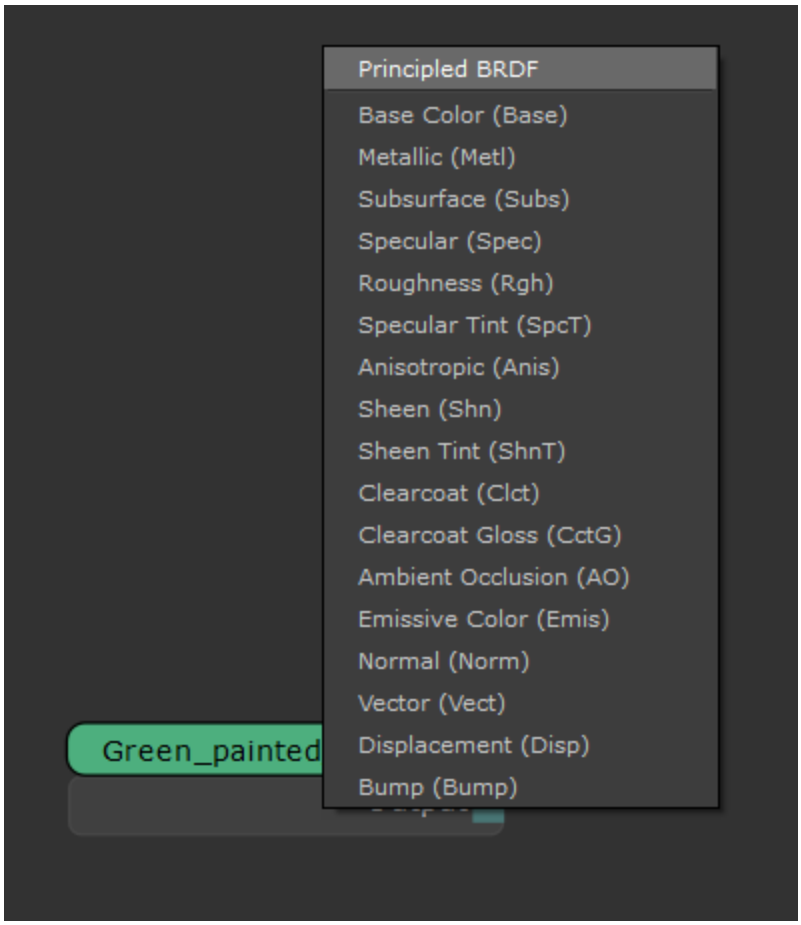
If you would like to view a specific material assigned to a multi-channel node in the Node Graph:

1. Select the Material or Multi-Channel node in the Node Graph.



A material node selected in the Node Graph.

2. Press **1**.
3. Select the user shader from the top of the pop-up list.



Selecting the **Principled BRDF** shader from the pop-up list.

The material assigned to the multi-channel merge node is shown on the model in the viewport. You can now view and edit your material in isolation, with a full PBR lighting response context.

> **Note:** If you would like to learn more about Material and Multi-Channel Merge nodes, see For some quick video tutorials of the new 4.5 Material System, check out Mari 4.5 Fundamentals video series here. and Multi-Channel Merge Node.

Creating a Layered Shader

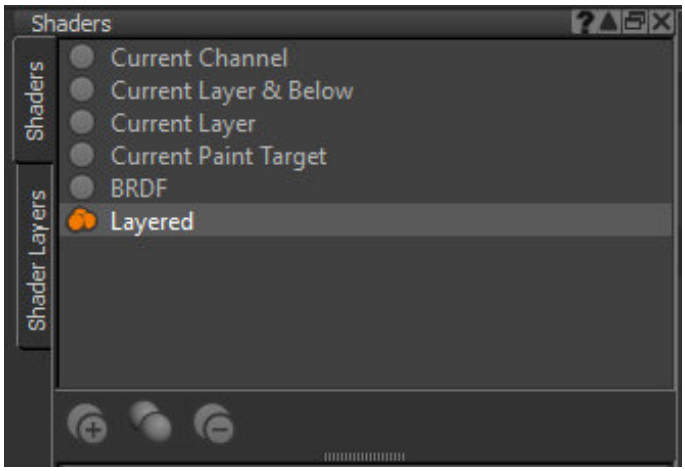
Layered shaders create a layer stack that allows multiple shaders to be blended together. When the layered shader is selected, the layer stack appears as a separate tab on the **Shaders** palette. This allows you to create different shaders within a layer stack that you can mask, adjust, and have independent control over.

For example, you can create a layered shader in order to create a shader for a wood material and another shader for a metal material on the same object. Each of these shaders can then be modified to create a specific look that may not be possible to achieve with an overall shader.

1. To add a layered shader, either:
   - right-click in the **Shaders** palette > **Add New Shader** > **Layered**
   - select **Add New Shader** > **Layered** from the **Shading** menu, or
   - click the icon in the **Shaders** palette and select **Layered** from the dropdown menu.

   The **Masks Format** dialog appears.
2. Select a **Size** and **Depth** for the channel created for the layered shader's layer stack. The options for the channel size and depth are the same as those found in the section to Creating and Deleting Individual Channels in the Channels section.

   The layered shader is created in the **Shaders** palette, and a **Shader Layers** tab is visible on the left side of the palette, when the layered shader is selected.
3. Click on the layered shader to select it and adjust the **Inputs** and attribute fields in the lower pane of the **Shaders** palette.
4. Once you have made the adjustments you want to make, click on the **Shader Layers** tab to see the shader's layer stack in the **Shaders** palette. This tab looks similar to the **Layers** palette or a mask stack palette.



Note: In the event that there are multiple layered shaders in the **Shaders** palette, there is still only one **Shader Layers** tab. Selecting a layered shader ensures it is that layer stack that is shown in the **Shader Layers** tab. To see the layer stack of another layered shader, simply select it before clicking on the tab.

5. Add shaders to the layer stack by clicking the icon and choosing the appropriate shader from the dropdown menu. As with shaders in the primary tab of the **Shaders** palette, the shader is added to the palette and you can adjust the **Inputs** and **Displacement** fields for each individual shader layer in the lower pane of the palette.
6. Similar to the **Layers** palette, you can add masks and mask stacks to the shaders, so that certain shaders in the stack only apply to certain areas on your model. For more information about how to mask layers, refer to Layer Properties.
7. To make it easier to find shaders in a large layer stack, you can assign color tags to individual shaders. Select a shader from the stack and click the icon to assign a color.
8. If you want to remove a shader from the layer stack, select the shader and click the icon. The shader is removed from the layer stack.
9. If you right-click on a shader layer, you can do most of the things you can also do with layers in the **Layers** palette, for instance, adding adjustments and groups.

Tip: For more information about layer stacks, mask stacks, and similar functionality on the **Layers** palette, refer to the Layers section for more information.

Setting Shader Details

For all of the custom shaders (those not mentioned in the Default Shaders list), the lower pane in the **Shaders** palette consists of inputs and sliders for the selected shader. The inputs you have available depend entirely on which of the shaders you have selected. The input fields allow you to specify which channel corresponds to which shader component. For example, if you're using the **Current Channel** shader, you don't need to specify a channel, because you are viewing only the selected channel on the model. However, if you have a **Unreal** shader selected, you are viewing whatever channels you have chosen for each component, such as **Base Color** or **Roughness**.

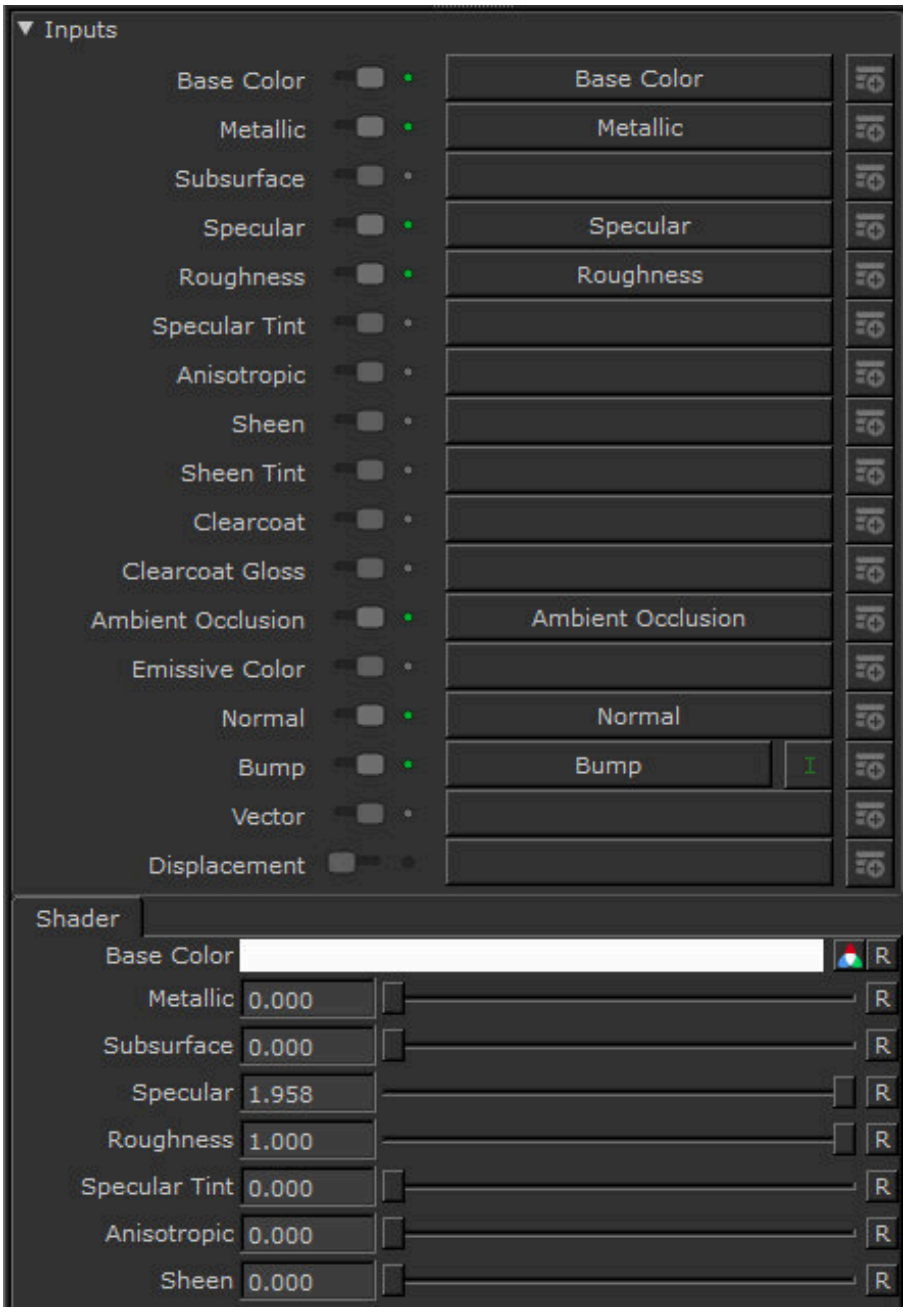
Once you've assigned a channel to the shader input, you can adjust the effect of that channel by adjusting the corresponding slider under the inputs.

> **Note:** This does not apply to all shaders. In the case of the Principled BRDF shader, its slider values are overridden by the values of the users active shader inputs.

Following on from the example above, if you have assigned a channel to the **Bump** input of the **Unreal** shader, you can then adjust the **Bump Weight** slider to control how heavily the bump affects the texture.

1. Select the shader you want to customize from the **Inputs** list at the top of the **Shaders** palette.

   The bottom of the palette displays that shader's details.



2. You can build up your shader by assigning channels to the various shader component inputs in the bottom pane of the palette.

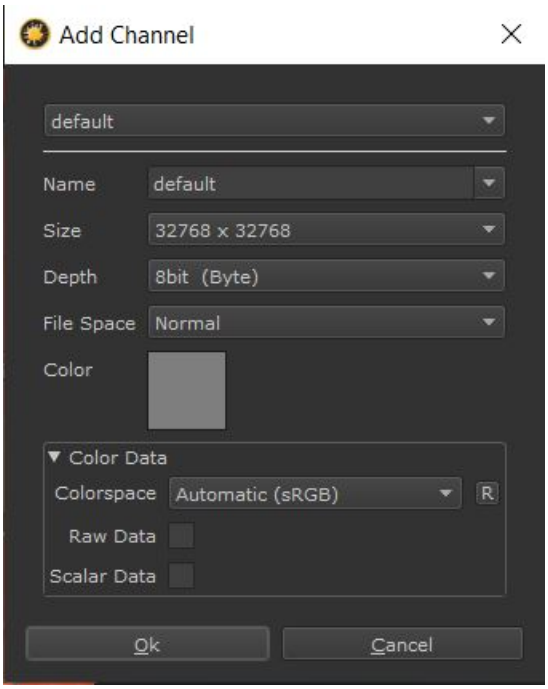
   To assign a channel to an input, click on the dropdown menu and select a channel from the list.

> **Tip:** Click the toggle to change the influence of channels connected to the shader without disconnecting them from the shader input. The Displacement shader input is toggled off by default.

3. If you haven't created a channel for a designated input yet, you can create channels from the **Shaders** palette.

   To create a new channel, click the icon next to the input for which you want to use the channel.

   The **Add Channel** dialog box displays.



4. Enter a **Name** for the channel (recent names are available in the dropdown list). Select the size, depth, colorspace, and file space settings for your channel and click **OK**.

   Mari adds the channel to your shader and automatically makes it the active channel for the designated input.

> **Tip:** To learn more about each of the settings in the **Add Channel** dialog, refer to the *Add Channel Dialog* section in the *Mari Reference Guide*.

5. With some shaders, once you've set up the inputs of each of the shader components you want to use, you can fine-tune these settings using the sliders in the bottom pane of the palette. This does not apply to the Principled BRDF.

   If you prefer, you can fine-tune the slider settings using the text field instead.

Removing a Shader

1. To remove a shader:
   - right-click on the shader and select **Remove Shader** from the dropdown menu,
   - select **Remove Shader** from the **Shading** menu, or
   - click the icon in the **Shaders** palette.

   The shader is removed from the **Shaders** palette.

> **Note:** The default shaders cannot be removed from the **Shaders** palette.

2. You can temporarily see your model as it appears in other shaders by simply selecting one of the default shaders or custom-created shaders. Selecting the shader displays the model as it appears with that shader's specific settings.
3. In all the custom shaders, controls for **Ambient** light do not correspond to a particular channel input, but to the entire model. **Ambient Occlusion** can also be selected from a channel, or created based on ambient occlusion calculation.

> **Note:** For more information on these shader controls, refer to the *Shaders Palette* section in the *Mari Reference Guide*.

Shader Types and Examples

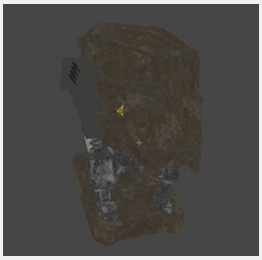
Below are descriptions and examples of the different shaders available in the **Shaders** palette. To know more about shader inputs, see the *Shaders Palette* section in the *Mari Reference Guide*.

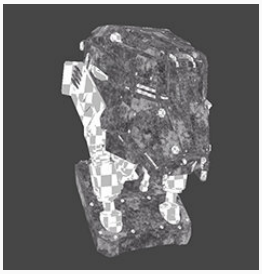
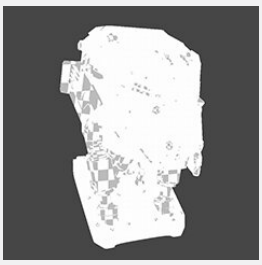
> **Note:** The project the illustrative image is taken from has three channels: diffuse, specular, and bump. For certain shaders, such as **Flat**, not all of these channels are hooked up to the input fields. Other shaders, such as the **BRDF** shader, have many other inputs that affect the shader output, but these may not be demonstrated from the three channels that are hooked into the shader.

Experiment: The illustrative image displays the **Example Project** model that's shipped with Mari. Try adding channels or layers to the **Example Project** in order to experiment with all the inputs and controls of the shaders in the table below.

Don't forget to adjust the sliders in the shader controls in order to modify the effect of the shader on the channel that you are trying out.

Note: The example images for each shader are highly dependent on how the controls have been adjusted for each shader. For the purposes of this table, most of the shaders are shown with default or near default settings unless otherwise specified.

Shader, Description, and Settings	Example
Current Channel A default shader that displays only the output of the selected channel. In the example shader, the diffuse channel is the currently selected channel. **Settings:** Adjust the sliders for the: **Diffuse**, **Specular**, and **Specular Roughness** amounts.	
Current Layer and Below A default shader that displays only the output of your selected sub-stack (such as a mask or adjustment stack) up to the selected layer. If you don't have a sub-stack selected, it shows the parent stack up to the selected layer. **Settings**: Adjust the sliders for the: **Diffuse**,	

Shader, Description, and Settings	Example
Specular, and **Specular Roughness** amounts.	
Current Layer A default shader that displays only the output of the current layer with the mask of the layer applied. **Settings**: Adjust the sliders for the: **Diffuse**, **Specular**, and **Specular Roughness** amounts. Specify whether the **View** is set to **Primary** or **Secondary** in the dropdown menu.	
Current Paint Target A default shader that displays only the selected component of the selected layer, for instance the paint, procedural, adjustment, or mask; whichever is selected on the selected layer. In the example shader, a procedural layer in a "dirt" mask stack is the current selected paint target. **Settings**: Adjust the sliders for the: **Diffuse**, **Specular**, and **Specular Roughness** amounts.	
Shader, Description, and Settings	Example

Shader, Description, and Settings	Example
BRDF (See *BRDF* in the *Mari Reference Guide*) A physically-based shading model that includes Fresnel effects. The **BRDF** shader defines how light is reflected at an opaque surface from both direct and indirect light sources. **Settings**: Set the channel used for the following inputs: **Diffuse Color** **Specular Color** **Glossiness** **Reflectance** **Ambient Occlusion** **Emissive Color** **Normal** **Bump** **Vector** **Displacement** The amount of each of these inputs can be adjusted in the respectively named sliders.	
VRayMtl *(See VRayMtl* section in the *Mari Reference Guide*) A shader that emulates the VRayMtl material shader, which allows physically-correct illumination and convenient reflection and refraction parameters. **Settings**: Set the channel used for the following inputs: **Diffuse Color**	

Shader, Description, and Settings	Example
Diffuse Amount	
Opacity Map	
Roughness Amount	
Self-Illumination	
Reflection Color	
Reflection Amount	
Reflection Glossiness	
Fresnel IOR	
GGX Tail Falloff	
Metalness	
Anisotropy	
Anisotropy Rotation	
Refraction Color	
Refraction Amount	
Refraction Glossiness	
Refraction IOR	
Fog Color	
Translucency Color	
Normal	
Bump	
Vector	
Displacement Map	

Shader, Description, and Settings	Example
The amount of each of these inputs can be adjusted in the respectively named sliders.	
Principled BRDF (See *Principled BRDF* section in the *Mari Reference Guide*) A physically-based shading model for PBR (physically-based rendering) that follows a set of principles to make BRDF shader control more intuitive, less complex, and artist friendly. For more information, see https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf. **Settings**: Set the channel used for the following inputs: **Base Color** **Metallic** **Subsurface** **Specular** **Roughness** **Specular Tint** **Anisotropic** **Sheen** **Sheen Tint** **Clearcoat** **Clearcoat Gloss** **Ambient Occlusion** **Emissive Color** **Normal** **Bump**	

Shader, Description, and Settings	Example
Vector **Displacement** The amount of each of these inputs can be adjusted in the respectively named sliders.	
3Delight Principled **3Delight Principled** (See *3Delight Principled* section in the *Mari Reference Guide*) The 3Delight Principled material is 3Delight's interpretation of the Principled BRDF shader. For more information, see https://www.3delight.com/documentation/display/sfrp/3delight+principled. **Settings**: Set the channel used for the following inputs: **Base Color** **Base Roughness** **Base Specular Level** **Metallic** **Anisotropy** **Anisotropy Direction** **Opacity** **Coat Thickness** **Coat Color** **Coat Roughness** **Coat Specular Level** **Subsurface Weight** **Subsurface Color**	

Shader, Description, and Settings	Example
Subsurface Scale **Incandescence Color** **Incandescence Intensity** **Normal** **Bump** **Normal Intensity** **Vector** **Displacement**	
Arnold Standard Surface (See *Arnold Standard Surface* section in the *Mari Reference Guide*) The Arnold Standard Surface shader is a physically-based shader capable of producing many types of materials. It includes a diffuse layer, a specular layer with complex Fresnel for metals, specular transmission for glass, subsurface scattering for skin, thin scattering for water and ice, a secondary specular coat, and light emission. The non-active inputs (highlighted in gray) help drive the shaders but have no effects on the canvas. For more information, see https://docs.arnoldrenderer.com/display/A5AFMUG/Standard+Surface. **Settings**: Set the channel used for the following inputs: **Diffuse Weight** **Diffuse Color** **Diffuse Roughness** **Metalness** **Specular Weight**	

Shader, Description, and Settings	Example
Specular Color	
Specular Roughness	
Specular IOR	
Anisotropy	
Anisotropy Rotation	
Transmission Weight	
Transmission Color	
Transmission Depth	
Transmission Scatter	
Transmission Scatter Anisotropy	
Transmission Dispersion	
Transmission Extra Roughness	
Subsurface Weight	
Subsurface Color	
Subsurface Radius	
Subsurface Scale	
Clearcoat Weight	
Clearcoat Color	
Clearcoat Roughness	
Clearcoat IOR	
Sheen Weight	
Sheen Color	

Shader, Description, and Settings	Example
Sheen Roughness **Thin Film Thickness** **Thin Film IOR** **Emission Weight** **Emission Color** **Opacity** **Normal Map** **Bump Map** **Vector Map** **Displacement Map** **Note:** Displacement Map is toggled off by default. The amount of each of these inputs can be adjusted in the respectively named sliders.	
Unreal (See *Unreal* section in the *Mari Reference Guide*) A physically-based material shader for assigning textures and materials that match what would be seen in the Unreal Engine 4. **Settings**: Set the channel used for the following inputs: **Base Color** **Roughness** **Metallic** **Specular**	

Shader, Description, and Settings	Example
Ambient Occlusion **Normal** **Bump** **Emissive Color** **Vector** **Displacement**	
USD Preview Surface (See *USD Preview Surface* section in the *Mari Reference Guide*) This shader approximates the look of Pixar's UsdPreviewSurface. This shader can be used in tandem with your primary render vendor's shader or Mari's BRDF. **Settings**: Set the channel used for the following inputs: **Diffuse Color** **Emissive Color** **Use Specular Workflow** **Specular Color** **Metallic** **Roughness** **Clearcoat** **Clearcoat Roughness** **Opacity** **Opacity Threshold** **Index of Refraction**	

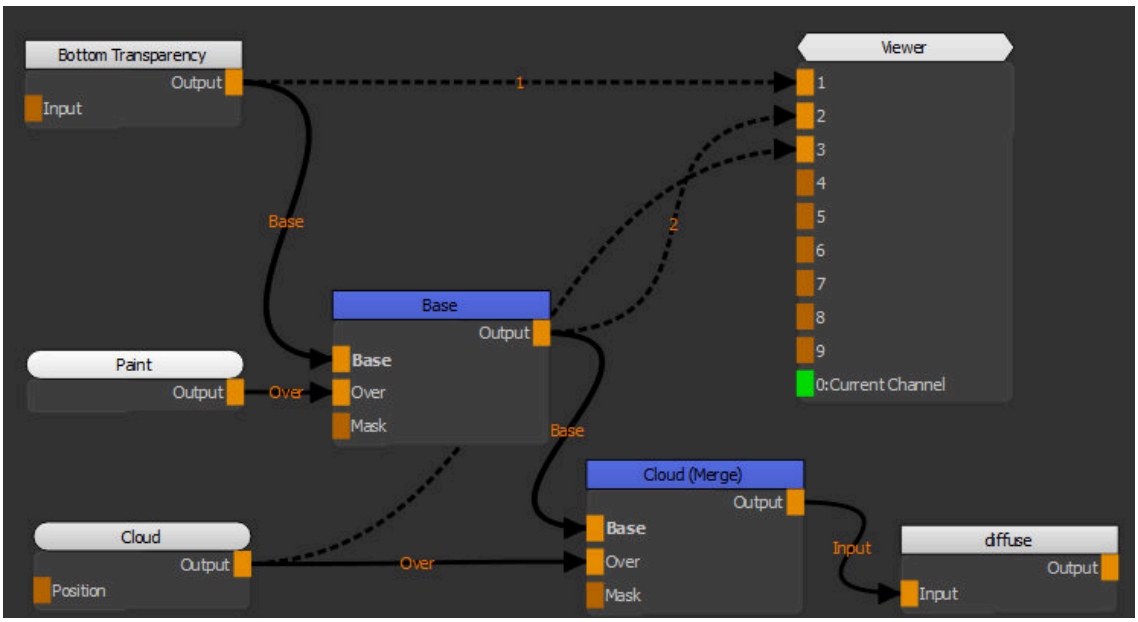
Shader, Description, and Settings	Example
Normal **Occlusion** **Bump** **Vector** **Displacement**	
Layered (See *Layered* section in the *Mari Reference Guide*) Creates a flat shader with a special channel that allows other shaders to be added as layers. Shader layers can be masked, grouped, and largely treated as layers in the **Layers** palette. The **Layered** shader lets you apply different shading models to create different material effects. **Settings**: Set the channel used for the following inputs: **Vector** **Displacement** The amount of displacement can be adjusted with the **Displacement** sliders.	

Node Graph

The Node Graph is essentially another way of viewing and managing channels, layers, and shaders. You can set up a Node Graph for ordering different sets of paint and combine effects in a non-destructive manner,

just like layers, but the Node Graph allows you to micro-manage the various components as individual nodes.

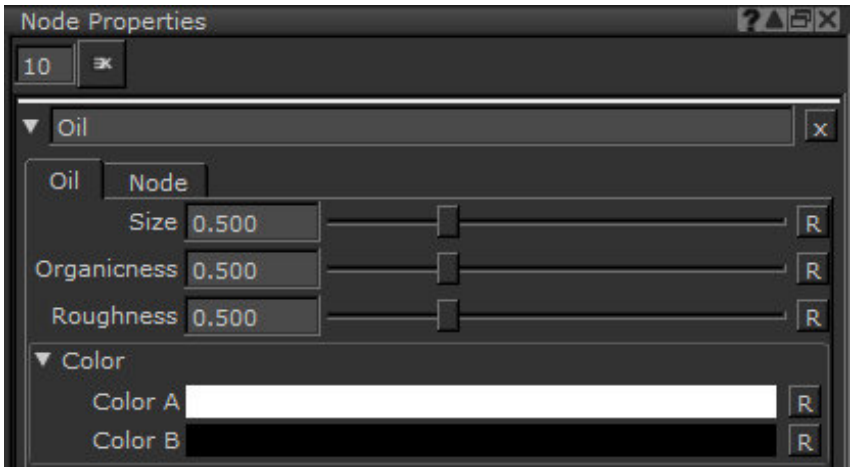
To view the Node Graph, navigate to **View** > **Palettes** > **Node Graph** or in the **Palettes Toolbar** (see the *Palettes Toolbar* section in the *Mari Reference Guide*), click the Node Graph.



A simple node tree.

The Node Graph contains nodes and each node has properties specific to the node's purpose located in the **Node Properties** palette. The **Node Properties** palette is also located in **View** > **Palettes**, but the fastest way to open a node's properties is to double-click it in the Node Graph.

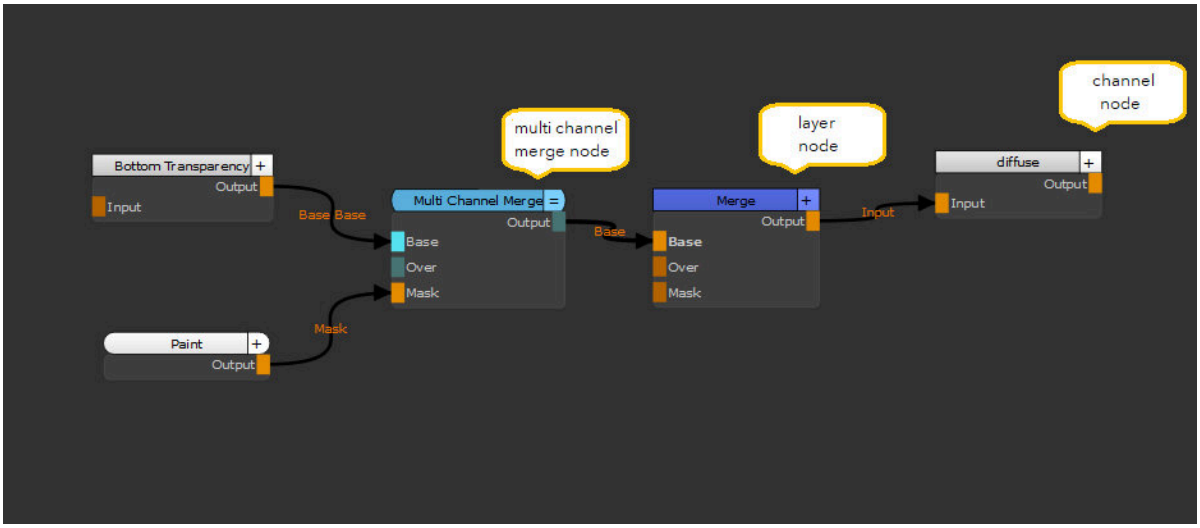
For example, the Oil node has controls for **Size** and **Color** to determine the look of the oil effect. See the *Mari Reference Guide* under *Layers Palette*, *Layers Palette Fields* for more information.



If you prefer working with layers, you can build a layer stack as normal, using channels and shaders, and Mari automatically replicates it in the Node Graph. See Layers and Adding Graph Layers for more information.

Objects, Channels, Layers, and Shaders Within the Node Graph

At first glance, the Node Graph that Mari creates automatically from layers can seem a little chaotic, but each element corresponds to a familiar concept.



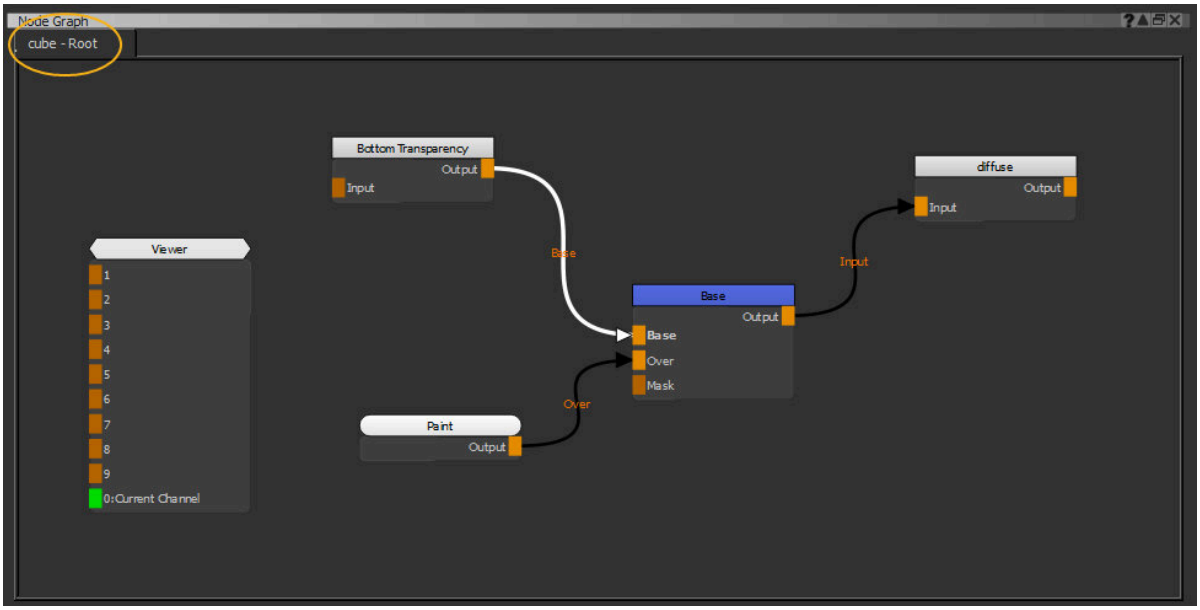
The types of nodes in the Node Graph in Mari 8.0v1.

Each Node Graph represents a single object and, just like in the layer system, each channel contains layers, including a Bottom Transparency layer. In the example, a Base layer is also included. If you add more layers to the channel, they appear under the first layer.

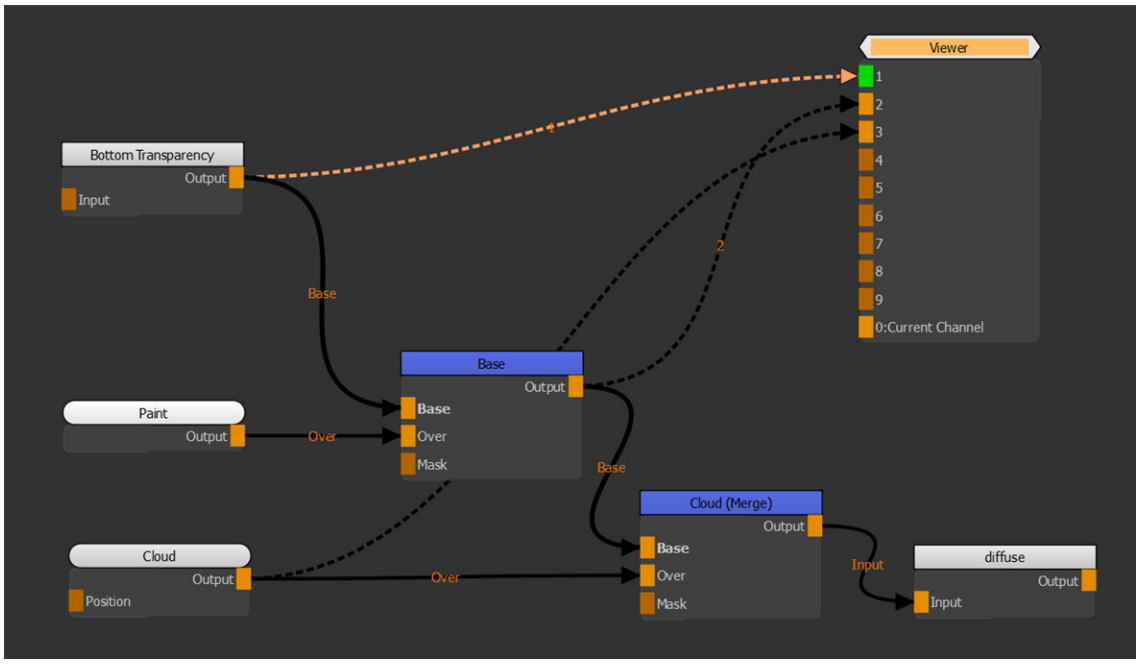
Tip: In Mari's Node Graph, you can either navigate from left-to-right (default), also called **No Port List**, or top-to-bottom, also called **Show Port List**, similar to the **Layers** palette. You can set the navigation type in the Mari Preferences Dialog (see the *Mari Preferences Dialog* section in the *Mari Reference Guide*). You can also switch the navigation directly in the Node Graph, either use **Shift**+**S** or right-click and select **Node Graph** > **Edit** > **Toggle Port List**.

Tip: You can collapse and uncollapse nodes using the button to the right of the node's title bar, and set a default collapse style in the Mari Preferences Dialog to make your Node Graph tidier. Also see Working with Nodes.

If a project contains multiple objects, the Node Graph Root name changes to represent the currently-selected object. For example, selecting an object called **cube** changes the Node Graph label to **cube - Root**.

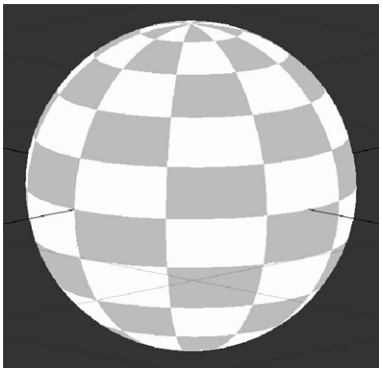


You can use the Viewer node to examine your node tree at any point to help you get to grips with how the Node Graph represents the Layers palette. Select the node you want to view and press **1** on the keyboard to connect the Viewer node. You can connect up to 9 nodes to the Viewer node. The **0** input of the Viewer node is automatically connected to what is selected in the **Shaders** palette.

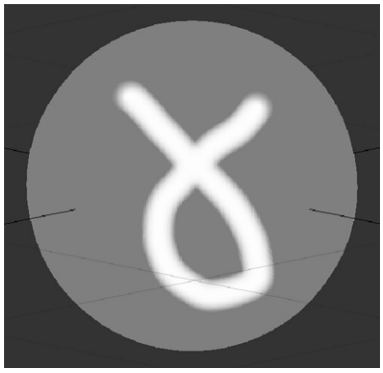


Tip: When you select a Multi-Channel node such as a Multi-Channel Merge or Material Node, press **0** (zero) to attach the 0th input of the Viewer node to the current material. Note, the current material is the only thing that can be connected to the 0th input, which defaults to the current material shader as determined by the Multi-Channel Merge or Material node that is selected.

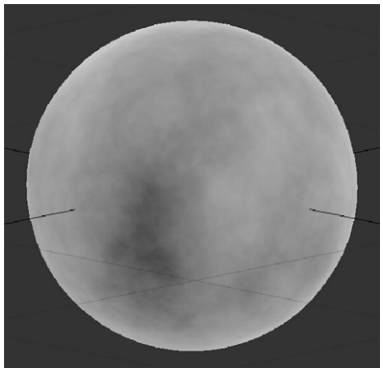
In the example, attaching the Viewer to the Bottom Transparency, Base (including paint in the Paint node), and Cloud nodes in succession produces the following results:



Bottom Transparency



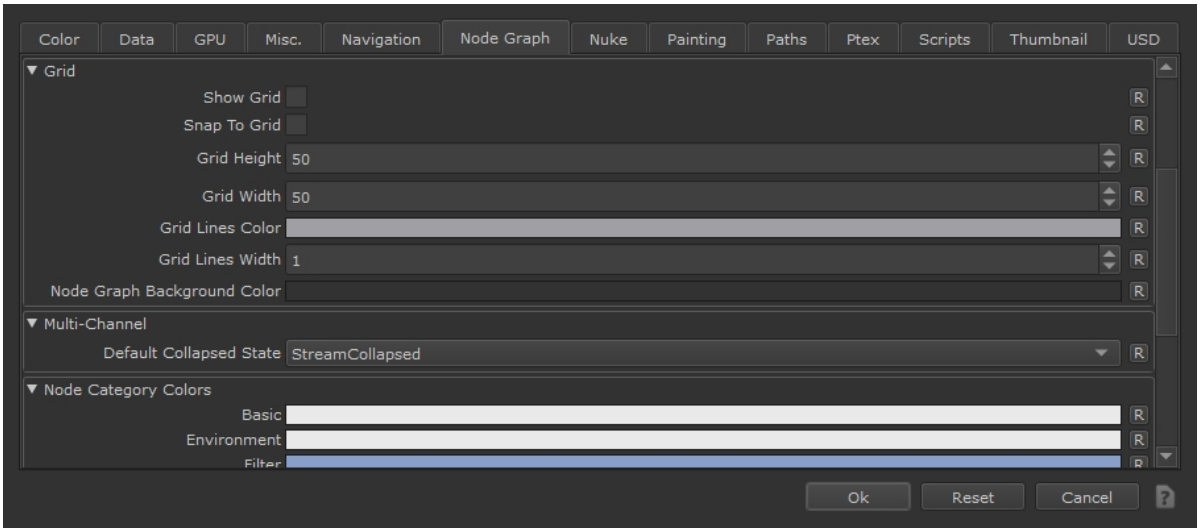
Base Layer, with Paint



Cloud Procedural

Snapping Nodes in the Node Graph

To aid the organization of your node network in the node graph, nodes, groups, and backdrops can be aligned and snapped to a grid.



To enable the grid and grid snapping:

1. Under the **Edit** menu, select **Preferences**.
2. Navigate to the **Node Graph** section of the **Preferences** menu and scroll down to the **Grid** section.
3. Toggle the **Snap To Grid** checkbox to enable grid snapping.
4. Toggle the **Show Grid** checkbox to enable the grid's visibility.
5. Adjust the height, width, color and line weight properties of the grid as required in the parameters available in the grid settings menu.

Working with Nodes

Nodes are the basic building blocks used to represent layers, channels, and shaders in Mari. Nodes can exist individually, within Backdrop nodes, or as part of a Group (including graph layers) in the Node Graph.

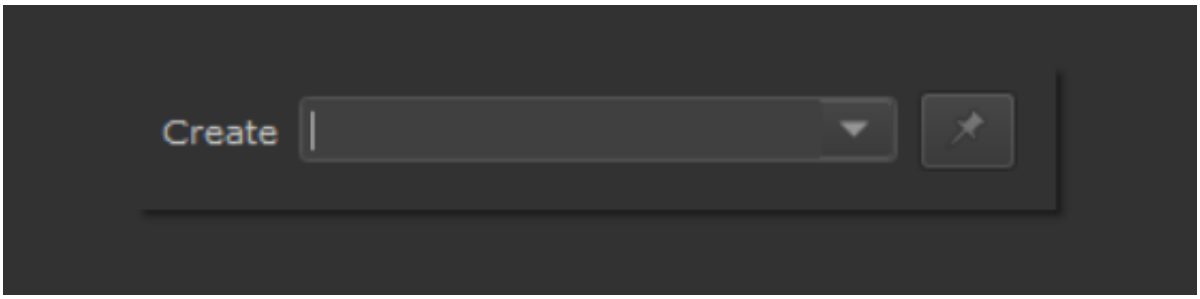
Adding, Deleting, and Bypassing Nodes

You can add nodes using the **Tab** menu or the right-click menu. When you add a node, Mari automatically connects it to the currently-selected node.

Adding and Searching for Nodes Using the Tab Menu

You can quickly add a node to the Node Graph by pressing **Tab** to open the **Create** dialog. If you want to add the new node downstream of an existing node, select the existing node before hitting **Tab**.

Once selected, hit **Enter** to add the node or press **Esc** to abandon the node insertion.



Hit **Tab** to open the node creation menu.

The Tab menu's features include:

- **Free text, fuzzy, scrollable search**: To find matches, type in the field, and then use the scrollbar to browse the results. The search is fuzzy, so you don't have to match the text exactly. For example, type **MCBP** to retrieve the Multi-Channel Bake Point node.

> **Tip:** You can scroll through search results with the cursor keys.

- **Recently Used**: Your most recently used nodes automatically appear at the top of the list. Recently used nodes are shown alongside a small time icon.
- **Favorites**: Click on the pin button to choose node(s) to be added to or removed from your favorites list. Click the pin again to close the list. Once added, nodes appear at the top of the list under the recently used nodes, together with a pin icon.

Adding Using the Right-Click Menu

1. Right-click on an existing node that you want the new node to follow.
2. From the menu that opens, select the node you want to add. For example, **Filter** > **Clamp**.

Tip: If you don't select an existing node, the new node is added at the node selector's location.

Adding Images through the Image Manager Palette

You can add images, as Tiled or Triplanar Projection nodes, to the Node Graph by dragging-and-dropping them from the **Image Manager** palette.

To add a Tiled node:

1. In the **Image Manager** palette, right-click and navigate to **Add to Node Graph** > **As Tiled**.

   OR

   Click and hold an image preview.
2. Drag-and-drop it in the **Node Graph** palette.

   The image is added as a Tiled node.

Note: You can add multiple images as Tiled nodes at once. In the **Image Manager** palette, select multiple images and drag-and-drop them in the **Node Graph** palette. The images are added as Tiled nodes.

To add a Triplanar Projection node:

1. In the **Image Manager** palette, right-click and navigate to **Add to Node Graph** > **As TriPlanar**.

   OR

   Press **Shift** then click and hold an image preview.
2. Drag-and-drop it in the **Node Graph** palette.

   The image is added as a Triplanar Projection node.

Note: You can add multiple images as Triplanar Projection nodes at once. In the **Image Manager** palette, select multiple images then press **Shift** and drag-and-drop them in the **Node Graph** palette. The images are added as Triplanar Projection nodes.

Deleting Nodes

To delete a node from the Node Graph, either:

• Select the node and press **Delete** (**Fn**+**Backspace** on Mac keyboards with no dedicated **Delete** key), or

• Select the node, right-click in the Node Graph and select **Edit** > **Delete**.

Bypassing and Disabling Nodes

Mari allows you to bypass a single node, or its inputs, and disable a selection of nodes to alter the image output as if the node(s) were removed from the node tree. Bypassing and disabling are similar, but work slightly differently.

To **Bypass** a single node:

1. Select the target node in the Node Graph.
2. Press **Ctrl**/**Cmd**+**D** on the keyboard or right-click in the Node Graph and then select **Edit** > **Bypass**.

   All inputs but one are disabled in order as you repeat the bypass action, negating their effect on the node tree without removing them.

   For example, cycling through the bypass modes for a Merge node with three inputs results in:

   • First bypass - all inputs except **Base** are disabled.

   • Second bypass - all inputs except **Over** are disabled.

   • Third bypass - all inputs except **Mask** are disabled.

   • Fourth bypass - all inputs are disabled, effectively terminating the upstream node tree.

   • Fifth bypass - all inputs are re-enabled.

To **Disable** a node or a selection of nodes:

1. Select the target node or nodes in the Node Graph.
2. Press **D** on the keyboard or right-click in the Node Graph and then select **Edit** > **Disable**.

   The selected nodes are disabled, negating all but their preferred input's effect on the node tree. For example, disabling a shader node only passes the diffuse input down the node tree.

> **Note:** You can set the preferred input for a node by editing the node's **.xml** file. For example, to set **Base** as the preferred input: **<Input Name="Base" PreferredInput='1'></Input>**
> Node **.xml** resides in the following locations, by platform:
> • Windows: <install_directory>\Bundle\Media\Nodes\
> • Mac OS X: <install_directory>/Content/Media/Nodes/
> • Linux: <install_directory>/Media/Nodes/

3. Press **D** or select **Disable** again to re-enable the selection.

Selecting Nodes

Mari offers a number of options for selecting nodes. Selected nodes are highlighted in yellow.

Type of Selection	How to Make the Selection
Select a single node	To select a single node, simply click on it.
Select multiple nodes	To select multiple nodes, you can either press **Shift** while clicking on each node you want to select, or click and drag in the workspace to draw a marquee around the nodes you want to select.
Select all nodes	Press **Ctrl**/**Cmd**+**A**.

Renaming Nodes

If you need to rename a node, you can either:

1. Click on the node and press **N** or right-click on the node and select **Edit** > **Rename Node.**

   The node name becomes editable.
2. Enter a new name and press **Enter**.

OR

1. Double-click on the node to open its properties.

   In the **Name** field at the top of the **Node Properties** panel, you should see the current name of the node.
2. Delete the current name and enter a new name in its place.

Editing Nodes

To copy, paste, and perform other editing functions in the Node Graph, you can use the standard editing keys (for example, **Ctrl**/**Cmd**+**C** to copy, and **Ctrl**/**Cmd**+**V** to paste). Copied nodes inherit the values of their parent, but these values are not actively linked. Copied nodes allow you to assign different values to the original and the copy.

When you paste nodes, Mari automatically connects them to the node that is selected before the paste operation. If you don't want to connect to anything, click on a blank area of the workspace to deselect any selected nodes before pasting.

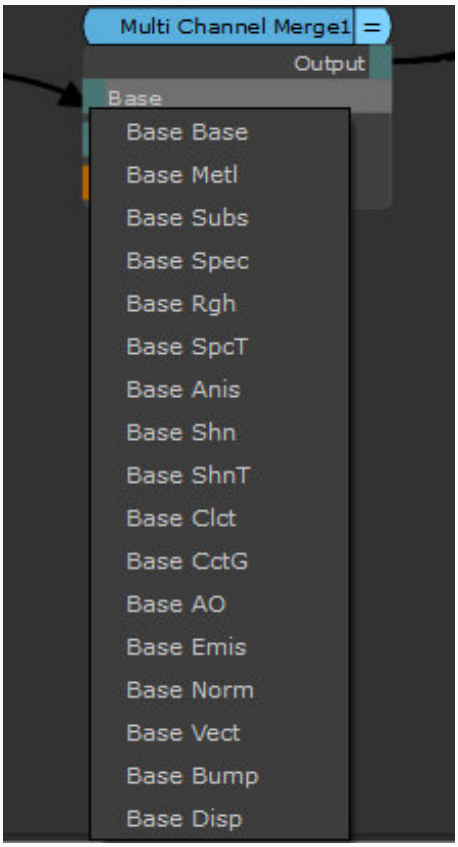
Type of Edit	How to Perform It
Copy nodes to the clipboard	To copy nodes to the clipboard: 1. Select the node or nodes you want to copy. 2. Right-click in the Node Graph and select **Edit** > **Copy** (or press **Ctrl**/**Cmd**+**C**).
Cut nodes	To cut nodes: 1. Select the node or nodes you want to cut. 2. Right-click and select **Edit** > **Cut** (or press **Ctrl**/**Cmd**+**X**). Mari removes the node(s) from the Node Graph and writes the node(s) to the clipboard.
Paste nodes from the clipboard	To paste nodes from the clipboard: 1. Select the node that you want the pasted node(s) to follow. 2. Right-click in the Node Graph and select **Edit** > **Paste** (or press **Ctrl**/**Cmd**+**V**). Mari adds the nodes to the Node Graph, connecting them to the selected node.

Collapsing Nodes

Click the button on the right of the title bar of your node to cycle through port list modes. Changing the collapsed state of the node can help to reduce the visual complexity of your Node Graph.

Note: When the Node Graph is zoomed out, the collapsed state button can't be clicked.

State	Icon	Description
Expanded		The node is fully expanded in the Node Graph, including all connected and unconnected streams.
Collapsed		The node is fully collapsed in the Node Graph and only displays the title bar, and

State	Icon	Description
		the input and output pipes.
Connected Only		The node is fully expanded but only displays input and output channels that are connected to other nodes.
Stream Collapsed		Partially collapses Multi-Channel node inputs to display a collapsed version of their input and output ports. For example, when a Multi-Channel Merge node is stream collapsed, you can only see **Base**, **Over**, and **Mask** inputs, and one **Output**. Port list appears when you connect a pipe to the stream collapsed port, allowing you to connect a pipe up to a port without fully expanding the node.  This state affects Multi-Channel nodes only.

Tip: Use **Shift** + ~ on selected nodes to cycle through their port list modes and set all selected nodes to the same collapsed state. The Stream Collapsed mode is skipped if you have a mix of nodes selected.

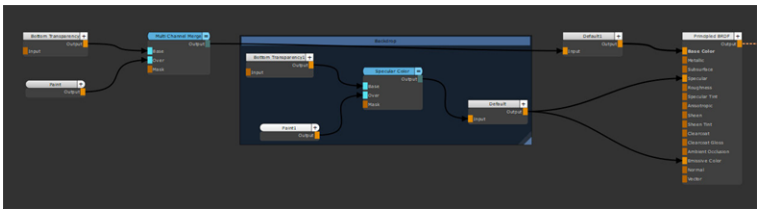
Tip: Click drag the title bar of the node to move the node around in the Node Graph.

Autoplacing Nodes

The Node Graph's autoplace functionality allows you to layout upstream nodes into tidy, structured columns. If Multi-Channel Merge nodes exist in your complex Node Graph, they act as a visual backbone. Activate Autoplace functionality in the Node Graph with the **L** key, or by right-clicking + selecting **Autoplace**. As a result, your upstream nodes are laid out into columns from left to right, minimizing as many crossed connections as possible.

Tip: Autoplacing does not occur for nodes within a Backdrop unless one is explicitly selected. See Backdrop Node for more information.

Tip: If any nodes within a column outside a Backdrop node intersect a Backdrop node, the entire column is moved left until it no longer overlaps the Backdrop node.



Multi-Channel Merge, Backdrop, and other nodes using Autoplace functionality in the Node Graph to restructure the nodes.

Baking to a Bake Point Node

The **Bake Point** node allows you to:

- Control the workflow, making the export of your scene or parts of your scene easier.

- Export baked data to Geo-Channels to instantly sync the result of the upstream network to all paired Geo-Channel layers and nodes across your Node Graph.
- Increase the performance of the Canvas by baking the result of the upstream network at a specific node, making the render process faster.
- Configure the data definition of the textures.

To create a Bake Point node:

1. In the Node Graph, right-click on a node where you want to bake to a Bake Point node and select **Nodes** > **Misc** > **Bake Point**.

   The **Add Bake Point** dialog opens.
2. Set the Image data. See the *Add Paint/Bake Point Dialog* section in the *Mari Reference Guide*.
3. Click **Ok**.

   The Bake Point node is created and connected to the selected node.
4. Double-click on the Bake Point node.

   The Bake point node panel is highlighted in the **Node Properties** palette, see Bake Point Node Properties for more information.

Warning: Changing colorspace settings of a channel removes any cache from all nodes of that channel.

Note: The Bake Point node is used for caching nodes, you can't paint in a Bake Point node.

Tip: Geo-Channels can be used as an intermediate caching point with the Bake Point node to allow complex node sharing without the difficulty of navigating complex shared connections in the Node Graph, by exporting the current bake data to a Geo-Channel on the object. See Adding Geo-Channels to Objects and Syncing Bake Points to Geo-Channel Nodes.

To update Bake Point nodes:

Note: The Bake Point nodes that need updating are highlighted in red.

1. Double-click on a Bake Point node.

   The Bake Point node panel is highlighted in red the **Node Properties** palette.
2. Click **Bake.**

   The Bake Point node is updated and turns green.

OR

1. In the Node Graph, select the required Bake Point node(s).
2. Right-click and select **Edit** > **Bake Points** > **Update Selected** or **Update Downstream**.

   The Bake Point nodes are updated and turn green.

Exporting and Importing Paint and Nodes

Mari allows you to export and import Paint image data or a selection of nodes for use in other projects or layers. Paint is saved as named files or sequences and node selections are saved as **.mng** files.

Action	How to Perform It
Export	To export Paint data: 1. Select the Paint node you want to export. 2. Right-click in the Node Graph and select **File** > **Export**. 3. See Exporting Textures from Layers for more information. **Tip:** To export Paint data from multiple Paint/Bake Point nodes: Select the required Paint and/or Bake Point nodes, right-click and select **File** > **Export**. Mari opens the **Export from Paint** dialog for each selected node first. See Exporting Textures from Layers for more information. Mari starts exporting.
Import	To import Paint data: 1. Right-click in the Node Graph and select **File** > **Import**. 2. See Importing Textures to Layers for more information.
Export nodes to file	To export nodes to file: 1. Select the node or nodes you want to export. 2. Right-click in the Node Graph and select **File** > **Export Nodes**. 3. Navigate to the directory where you want to store the node(s). 4. Type a name for the export at the end of the pathway, followed by the **.mng** extension.
Import nodes from file	To load nodes from file: 1. Select the node that you want the loaded node(s) to follow.

Action	How to Perform It
	2. Right-click in the Node Graph and select **File** > **Import Nodes**. 3. Navigate to the directory that stores the **.mng** node file. 4. Select the file, and click **Open**. Mari adds the node(s) described by the file to the selected node.

Customizing Node Colors and Behavior

You can customize node colors by type, such as all Filter nodes, or make changes to the appearance of the Node Graph and default node size. In the case of certain nodes, such as Dot and Backdrop, you can also customize the node's size when added to the Node Graph.

All default node parameters are stored in a file called **NodeGraphStyleSheet.ini** (on Windows and Mac OS X) or **NodeGraphStyleSheet.conf** (on Linux) which resides in your **.mari** or **.config** file depending on your operating system:

- Mac OS X: <home directory>**/.config/TheFoundry/NodeGraphStyleSheet.ini**
- Linux: <home directory>**/.config/TheFoundry/NodeGraphStyleSheet.conf**
- Windows: <home directory>**\.mari\TheFoundry\NodeGraphStyleSheet.in**i

Within this file there are several broad categories, which you can edit to customize the appearance of the Node Graph and the nodes it contains.

Category	Function	Example
Scene	Sets the Node Graph background color.	SceneBackgroundColor="(50,50,50,255)"
NodeStateColors	Sets the color used to denote various node states and the general appearance of nodes.	NodeTextColor="(0,0,0,255)"
Category Colors	Sets the color used to denote a particular category of nodes.	FilterNodeColor="(234,234,234,255),(125,125,125,255)"
Edge	Sets the appearance and behavior of node edges when you connect pipes to nodes.	EdgeLabelColor="(234,111,0,255)"
NodeDimensions	Sets the size of nodes that don't have	NodeHeight=20

Category	Function	Example
	specialized controls of their own, such as Dot and Backdrop.	
AutoInsert	Sets the size of the activation zone when you drag a node over an existing pipe to auto-insert between nodes.	AutoInsertItemWidth=4
Backdrops	Sets the size and appearance of Backdrop nodes.	MinimumBackdropNodeHeight=150
StickyNoteNode	Sets the minimum height of StickyNote nodes.	MinimumStickyNoteNodeHeight=30
DotNode	Sets the size of Dot nodes.	DotNodeSize=16

Connecting Nodes

To connect nodes, click on a node's output port to start drawing the connection, then click on the destination node's input port to connect the two nodes. Connecting also works from input to output.

When you add or paste nodes into the Node Graph, Mari automatically generates noodles between the currently selected node and the new nodes. As you build up a node tree, you may need to move these pipes or run new pipes between nodes. Simply drag either end of the pipe to the new location.

You can insert a node into an existing pipe by dropping the node onto the connection. You can toggle this behavior in the **Mari Preferences** dialog under **Node Graph** > **General** > **Auto Insert Node**.

When **Auto Insert** node is enabled, you can define an auto insert hitbox, which will create an area in which new nodes can be auto-inserted. In the **Preferences** dialog, navigate to **Node Graph > General > Auto Insert Height** and **Auto Insert Width** to define a desired target area for your nodes to be dropped into. You can hold **Shift** to temporarily disable auto-insertion while dragging a node.

If you are connecting multi-channel nodes, such as a Material node to a Multi-Channel Merge node, you can connect just one output to the corresponding input and the remaining outputs automatically connect to the remaining inputs.

Note: For more information on connecting nodes, see Material Node Workflow Example and Applying a Material to an Asset.

Disconnecting and Reconnecting Nodes

To disconnect, drag the head or tail of the noodle to an empty area of the node graph. To reconnect, drag and drop the arrow over the node's input or output.

Other ways to disconnect:

- **Shake the node**: Simply selectand shake the node to disconnect it. **Note**: You can toggle shaking and adjust shake settings under **Edit** > **Preferences** > **Shake to Extract**. See Mari Preferences Dialog.
- **Keyboard and menu**: To disconnect a node or group of nodes, select the node(s) and press **Ctrl/Cmd**+**Shift**+**X** or right-click and select **Edit** > **Extract**
- **Sever connections with the Razor**: Press **Y** to turn the cursor into a razor. Drag the razor over the connections you want to cut. The connections turn orange and disappear. Press **Y** to return to pointer mode.

Note: You can detach a node from all of its connections. To do so, select the node and either press **Ctrl/Cmd**+**Shift**+**X** or right-click and select **Edit** > **Extract**.

Note: You can switch the first two inputs of a node by pressing **Shift**+**X**. For this to work, the node must have at least one connection.

Connecting Distant Nodes

It can be tricky to drag connections between nodes that are far apart in the node graph. Instead, you can search for the destination node once you've selected an output on the source node:

1. Click on the output of the node you want to connect from.
2. Hit **J** to search for the destination node and select the one you need.
3. Click on the destination node input to make the connection.

Note: Instead of clicking, you can also use the ‘ (backtick) key (which is usually located next to the 1 key) to select an output or input.

You can also connect distant nodes using the Teleport Broadcaster and Teleport Receiver nodes. These make a noodle-less, hidden connection between nodes, helping you construct a tidier node graph. See Teleport Broadcaster and Teleport Receiver Nodes.

Node Input Types

Nodes in Mari have different inputs depending on their function. Some are simply labeled **A** and **B**, as in the Add and Divide nodes. Some nodes, such as Ambient Occlusion and Color, require no inputs at all. The following table lists some common inputs and example nodes that employ them:

Input	Function	Example Nodes
Input	A standard input with no particular requirements.	Brightness
Mask	Passes masking information into the associated node.	Merge
Normal	Passes surface normals into the associated node.	Beckman Specular, Displacement
Position	Passes positional coordinates into the associated node.	Cube Map Projector, Oil
ToView	Connect this input to any point in the node tree to view the output up to that point.	Viewer
UV	Passes UV coordinates into the associated node.	Tiled Texture
Vector	Passes vector information into the associated node.	Set Vector

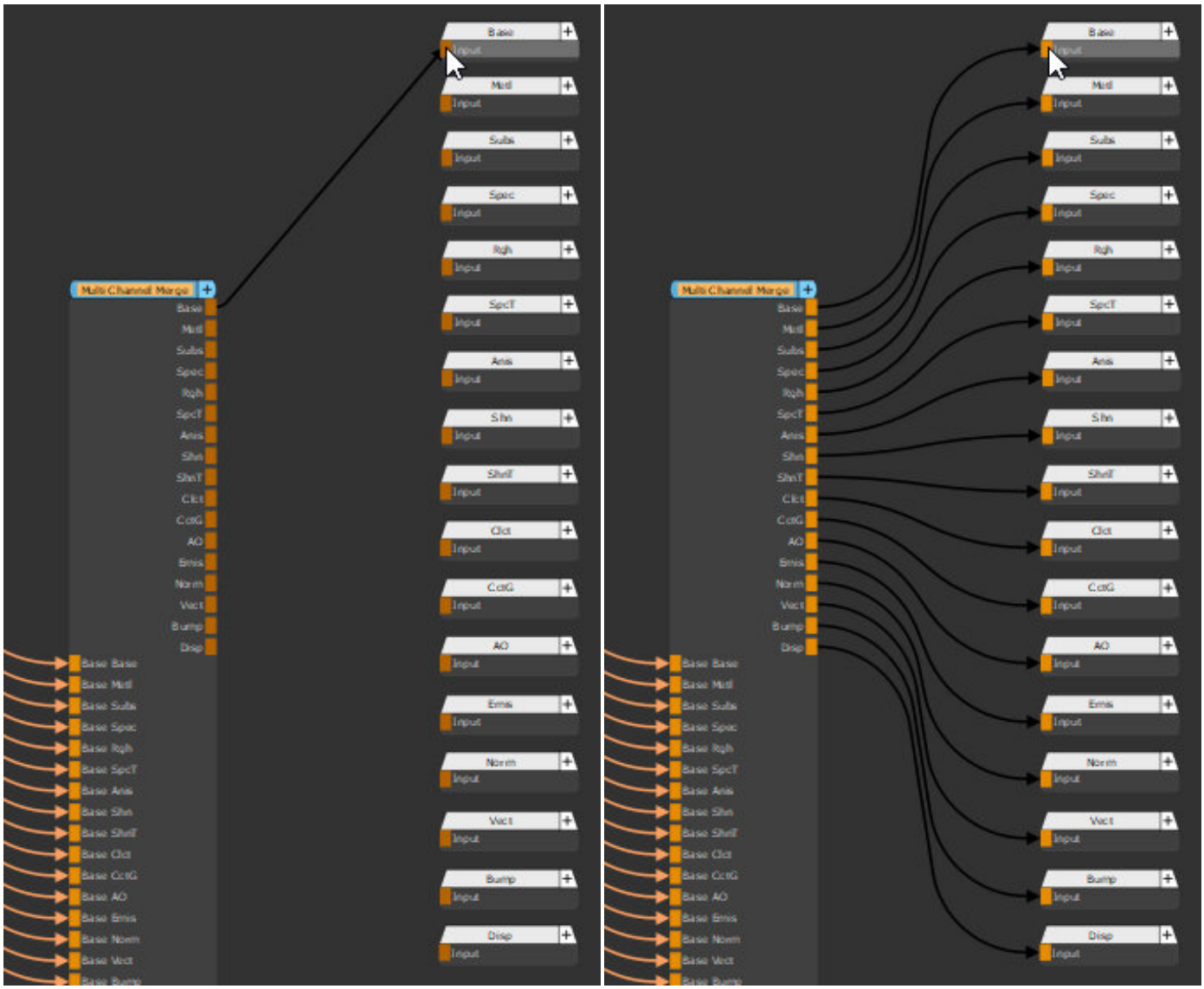
Note: You can also connect a node by dragging it over an existing pipe, Mari then auto-inserts the node between nodes. You can enable or disable this option in the **Mari Preferences** dialog under **Node Graph** > **General** > **Auto Insert Node**.

Connecting Multi-Channel Nodes

If you are connecting a source node with shader model driven output ports, such as a Multi-Channel Merge node, to another Multi-Channel node or the Output nodes within a Material node, you can connect just one stream of the source node's output ports to its corresponding input, and all sibling output ports automatically connect to the sibling stream. If an output port has an existing connection, its stream is skipped.

In the same way, when disconnecting a Multi-Channel node or the Output nodes within a Material node from a source node with shader model driven output ports, all sibling input ports also disconnect from their corresponding sibling output ports. If an output port is connected to a different node, then its connection is maintained.

Note: For more information on Materials and Output nodes, see Material Node Workflow Example



Batch connecting a Multi-Channel Merge node to Output nodes within a Material node

If one or both of the Multi-Channel nodes are Stream Collapsed , you can still connect all the channels at once by dragging the connection from the output port to the input port on the other Multi-Channel node. Once connected, you can expand the nodes to see the full connections.

> **Note:** For information about collapsing Multi-Channel nodes and making connections, see Collapsing Nodes.

Hold **Shift** to connect just one output port without the others connecting automatically.

Hold **Shift** to disconnect just one output port without the others disconnecting automatically.

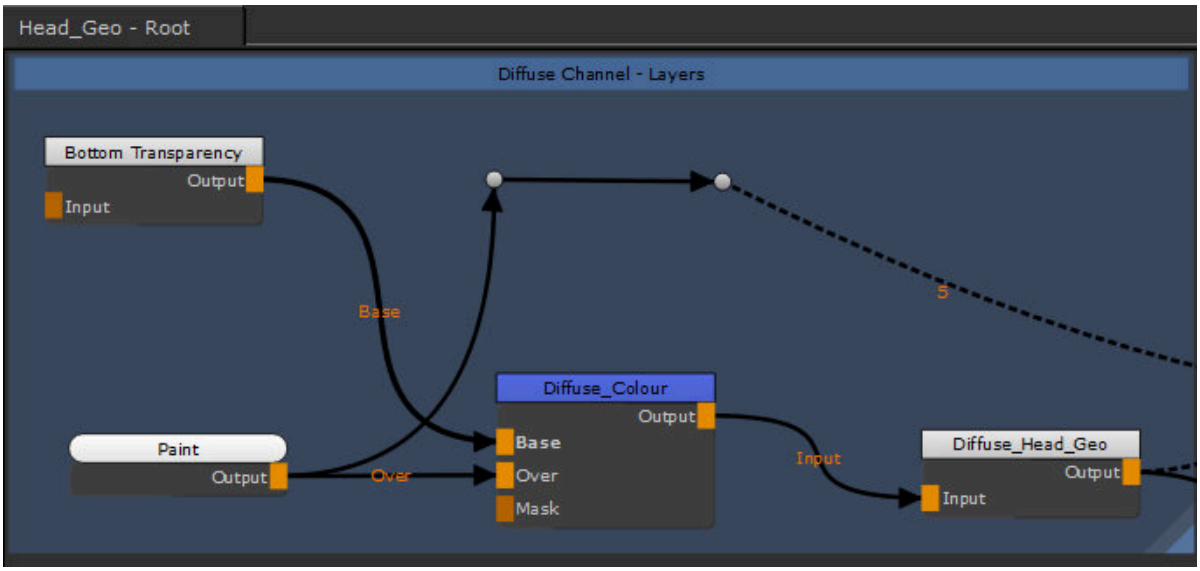
Bending Connecting Arrows

Complex Node Graphs can become difficult to read with pipe overlaps, but Mari allows you to bend pipes using a specialized node called Dot. The Dot node doesn't have any special controls like regular nodes, but it can be used to direct pipes away from congested areas of the Node Graph.

To add a Dot node, select the node before the connector you want to bend and then, either:

- press **Tab** and type **Dot** to add a Dot node,
- right-click and select **Nodes** > **Misc** > **Dot**, or
- hold **Ctrl**/**Cmd** and click the yellow highlight on the required pipe.

A Dot is placed after the selected node, allowing you to bend the pipe as required by dragging the Dot to a new position in the Node Graph.



Navigating Inside the Node Graph

As node trees grow in complexity, you need to be able to pan to or frame a particular cluster of nodes quickly. The Node Graph offers a couple of methods for doing so.

In Mari's Node Graph, you can either navigate from left-to-right (default), also called **No Port List**, or top-to-bottom, also called **Show Port List**. You can set the navigation type in the Mari Preferences dialog, see the *Mari Preferences Dialog* section in the *Mari Reference Guide*.

You can also switch the navigation of selected nodes directly in the Node Graph, either use **Shift**+**S** or right-click and select **Node Graph** > **Edit** > **Toggle Port List**.

Panning

Panning with the Mouse

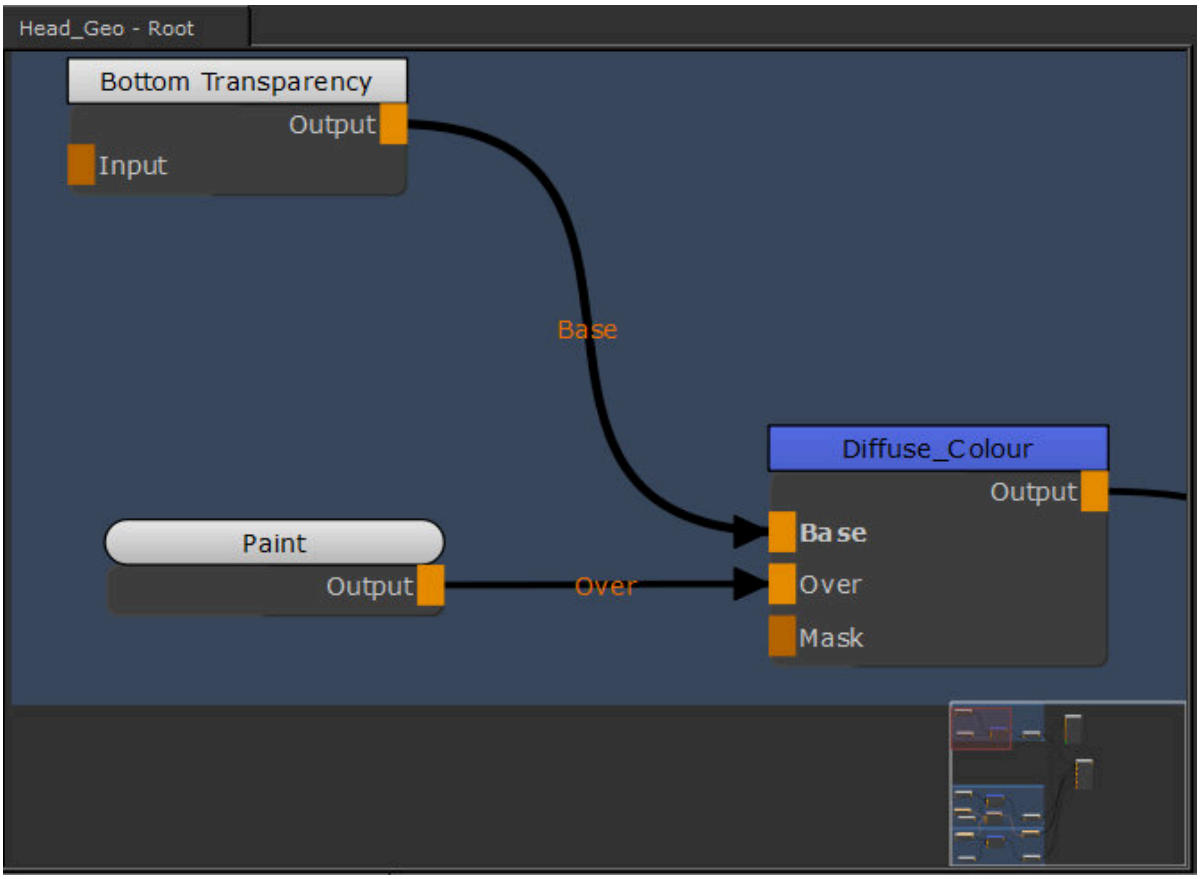
To pan with the mouse, press the middle mouse button and drag the pointer over the Node Graph or hold **Alt** and drag. The node tree moves with your pointer.

Panning with the Arrow Keys

You can pan in the Node Graph by holding **Alt** and pressing the Up, Down, Right and Left arrow keys on the keyboard. The Node Graph pans in the direction of the arrow you press.

Panning with the Map

If your Node Graph is larger than the visible workspace, a navigator map automatically appears in the bottom-right corner.



The map shows you a miniature view of the entire Node Graph and the red rectangle shows the portion of the Node Graph that you see within the workspace borders.

To pan with the map, drag the red rectangle to pan to a different view of the Node Graph.

When the whole Node Graph is contained within the window border, then the map automatically disappears.

Zooming and Fitting in the Node Graph

To zoom with the mouse, either:

- use the mouse scroll wheel, or
- hold **Alt** and middle-mouse drag left and right.

Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.
You can use key mapping to assign the mouse modifier to another key, such as the (**Super** or **Meta**) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

To zoom with the keyboard:

- Press **Shift** + **Alt** + Up arrow to zoom out.
- Press **Shift** + **Alt** + Down arrow to zoom in.

You can set whether to zoom according to the cursor's position or the center of the Node Graph. To set the zoom preference:

1. In the menu bar, navigate to the **Edit** > **Preferences** > **Node Graph** tab.
2. In the **General** section, from the **Zoom To** dropdown menu, select either **Center** or **Cursor**.

To center the node tree, press **A** with the Node Graph in focus. To center certain selected nodes, press **F** with the Node Graph in focus.

Note: Pressing **F** without selecting any nodes produces the same result as pressing **A**.

Searching for Nodes

Mari allows you to look for nodes that are currently used in the Node Graph. To look for a node:

1. Press **J.**
   OR
   Right-click the Node Graph and select **Edit** > **Search Node**.
   The **Search** box pops up in the Node Graph.
2. Enter the node name or part of the node name you're looking for.
   The nodes containing the letters you've entered display in the dropdown menu.
3. Select the required node.
   Mari takes you directly to the selected node in the Node Graph and highlights it.

Note: You can also do a wildcard search, which is based on character pattern, using the character *.
For instance, if you're looking for nodes that contain the letters b and d in that order, enter **b*d** in the **Search** box, this displays any Principled BRDF, Backdrop, or Object Space Linear Gradient nodes in existence, providing they have not been renamed. Any nodes that you have renamed, which contain the letters b and d in the specified order, also appear in the dropdown menu.

Cleaning up the Node Graph

Sometimes you may find that your node tree becomes disorganized as you're creating and removing nodes. There's a quick fix you can use to tidy up your tree:

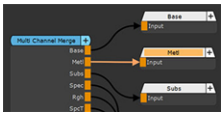
1. Right-click in the Node Graph.
2. Select **Autoplace**.
   The nodes in the tree are re-organized for clarity.

Tip: If you make a selection of nodes before using **Autoplace**, only the selected nodes are re-arranged.

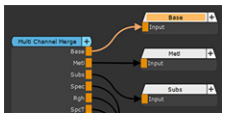
Selecting Nodes

When working inside the Node Graph, you can use keyboard shortcuts to quickly select nodes depending on their position to a selected node or nodes.

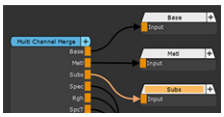
- Up/Down arrow keys - Select previous/next sibling node.



Original node selection

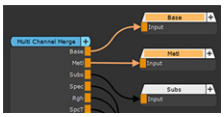


Up arrow key selects previous sibling node

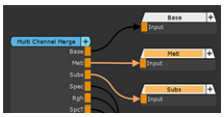


Down arrow key selects next sibling node

Hold **Shift** to add to the selection rather than replace it.



Shift + Up arrow key adds previous sibling node to selection



Shift + Down arrow key adds next sibling node to selection

- Left/Right arrow keys - Select the next upstream/downstream node(s).

Original node selection

Left arrow key selects the next upstream node (s)

Right arrow key selects the next downstream node(s)

Hold **Shift** to add the next upstream/downstream node(s) to the selection, rather than replace it.

Shift + Left arrow add the next upstream node(s) to the selection

Shift + Right arrow adds the next downstream node(s) to the selection

Hold **Ctrl** to select all upstream/downstream nodes.

Hold **Ctrl** + **Shift** to select all upstream/downstream nodes which are part of the current stream.

Original node selection

Ctrl + Left arrow selects all upstream nodes

Ctrl + **Shift** + Left arrow selects all upstream nodes that directly affect the selected node

Original node selection

Ctrl + Right arrow selects all downstream nodes

Ctrl + **Shift** + Right arrow selects all downstream nodes that are directly affected by the selected node

- **Ctrl** + **Shift** + Space bar - Select all of the nodes between the current node selection.

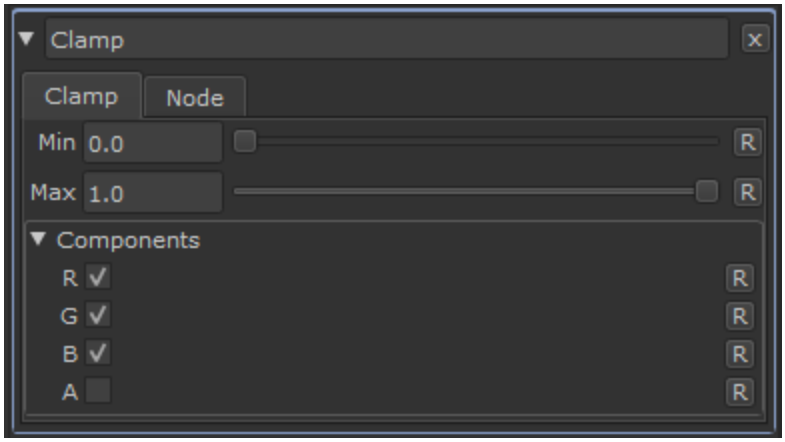
Selected nodes

Ctrl + **Shift** + Space bar adds all nodes in-between the selected nodes to the selection

- **Ctrl** + **P** - Open the **Node Properties** panel for the selected nodes.

Working with Node Properties

Node properties allow you to control how a node affects the node tree. To display a node's properties, double-click it in the Node Graph. The properties appear at the top of the **Node Properties** panel, above any recent node properties. For example, double-clicking a Clamp node in the Node Graph displays the following controls:



Note: For a full list of available nodes and controls, see Layer Types and Examples.

Mari helps you see which properties belong to which node by highlighting the node properties when you roll over the node, and vice versa.

Node Properties: Node Tab Controls

In addition to the controls unique to the various nodes, all nodes share certain universal controls within the **Node** dropdown:

- **Delete Cache** - click to delete cached data for the current node or Group.
- **Cache** - click to cache the current node or Group data to the location specified in the **Preferences** under **Data** > **Project** > **Project Location**.

> **Note:** Node caching is now being deprecated and will be removed in a future release to introduce a new mechanism for caching portions of your project.

> **Note:** You can set, using the **Use Cache for Bake** preference, whether to use the cached result of a node/layer for baking, see the *Mari Preferences Dialog* section in the *Mari Reference Guide*.

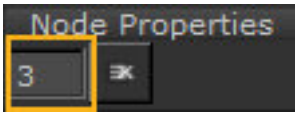
> **Note:** If you modify the controls of a cached node or Group, the cache is deleted.

> **Note:** The **Cache** option caches material nodes across all siblings channels. See Working with Materials in the Node Graph for more information on materials.

- **Export as Gizmo** - for Group nodes only, you can export the Group as a gizmo to share your work with other artists. See Exporting Groups as Gizmos for more information.

Managing the Node Properties Panes

You can limit the number of panes that are open in the **Node Properties** palette. To do so, enter the maximum number of node properties panes in the field in the **Node Properties** palette.



To empty the **Node Properties** palette and close all the node properties panes in it, click the remove all panels button.

Using Backdrops, StickyNotes, Groups, and Gizmos

You can organize and label nodes in the Node Graph using the Backdrop and StickyNotes nodes or use the Group node. The Backdrop node adds a background box behind the nodes, separating the nodes visually from the rest of the node tree. The StickyNote node allows you to add annotations to the elements in the node tree.

A Group node combines a set of nodes into a single node, acting as a nesting container for them. Group nodes can be published as gizmos, allowing you to share parts of your node tree that perform specific functions with other artists.

Note: Groups and gizmos support the usual range of Mari metadata types as Name=Value pairs.

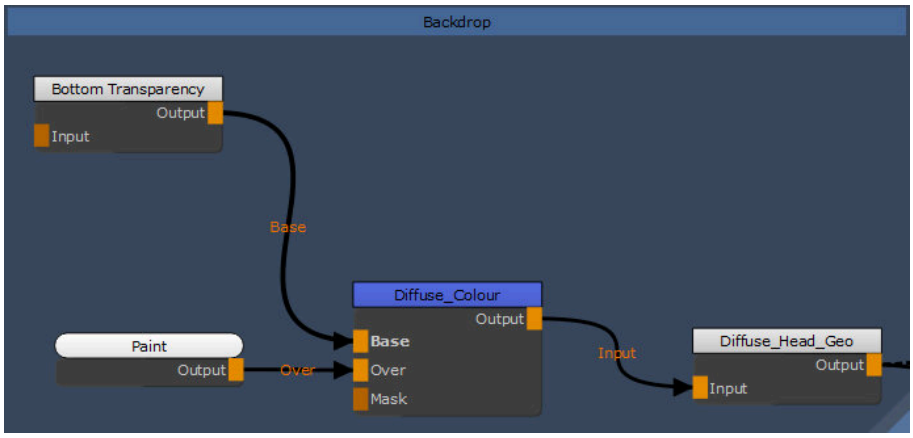
Adding Backdrops

Backdrop nodes allow you to organize parts of your node tree into named areas performing a specific function.

To add a Backdrop, do the following:

1. Select the nodes you want to place within the Backdrop node.
2. To add a Backdrop, either:
   - press **Tab** to display the node selector, type **Backdrop**, and press **Return**, or
   - right-click in the Node Graph and select **Node** > **Misc** > **Backdrop**.

   A Backdrop is added containing the selected nodes.



3. Double-click the Backdrop node's header to display its properties.

> **Note:** You can use the **Z Order** control in the **Node Properties** palette to layer-up Backdrop nodes. Backdrops with lower **Z Order** values appear underneath those with a higher value.

4. Click the **Background Color** and select a new color, if required.
5. Click the **Label** field to add a text description to the Backdrop's background.
6. Enter the required Backdrop name in the **Name** field.

Adding StickyNotes

StickyNote nodes allow you to add annotations to elements in the node tree. For example, you might add text describing the function of a certain part of the node tree.

To add a StickyNote:

1. Press **Tab** to display the node selector, type **StickyNote**, and press **Return**, or right-click in the Node Graph and select **Nodes** > **Misc** > **StickyNote**.
2. Double-click the node to open its properties.
3. Enter the note text in the **Label** field.

   The text is added to the StickyNote in the Node Graph.

Managing Groups

Group nodes allow you to condense parts of your node tree into groups performing a specific function.

Creating Groups

Mari creates groups automatically in some cases, such as when you add a new graph layer from the **Layers** palette, but you can create groups manually from a selection of nodes.

To create a Group:

1. Select the nodes you want to place within the Group node.
2. To create the Group node, do one of the following:
   - press **Ctrl**/**Cmd**+**G**, the Group keyboard shortcut,
   - press **Tab** to display the node selector, type **Group**, and press **Return**,
   - right-click in the Node Graph and select **Group**.

   A Group node with the required number of Input pipes and one Output pipe is added. For example, if you have three nodes that are connected to nodes within the Group, Mari creates three inputs for the Group node.

> **Tip:** If you didn't make any node selections, an empty Group is created.

3. Double-click the Group node to display its contents and properties in the **Node Properties** palette.
4. In the **Node Properties** palette, click the **Name** field to give the Group a meaningful name.

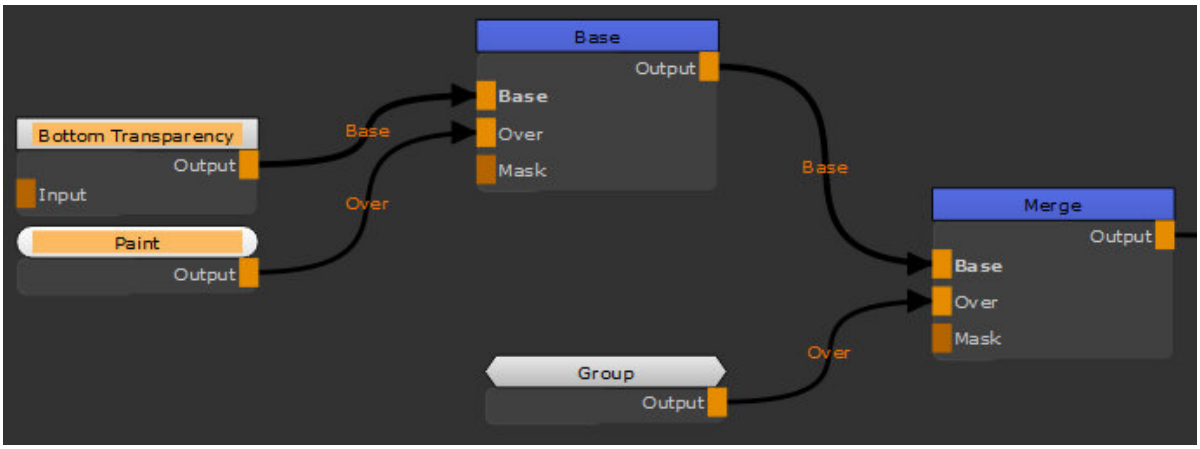
> **Note:** To enter a Group node **Ctrl** +double-click the Group node to open its subgraph. Alternatively, select the Group node and press **Ctrl**+**Enter** or in the **Node Properties** palette, click the **s** button.

Exposing Group Properties

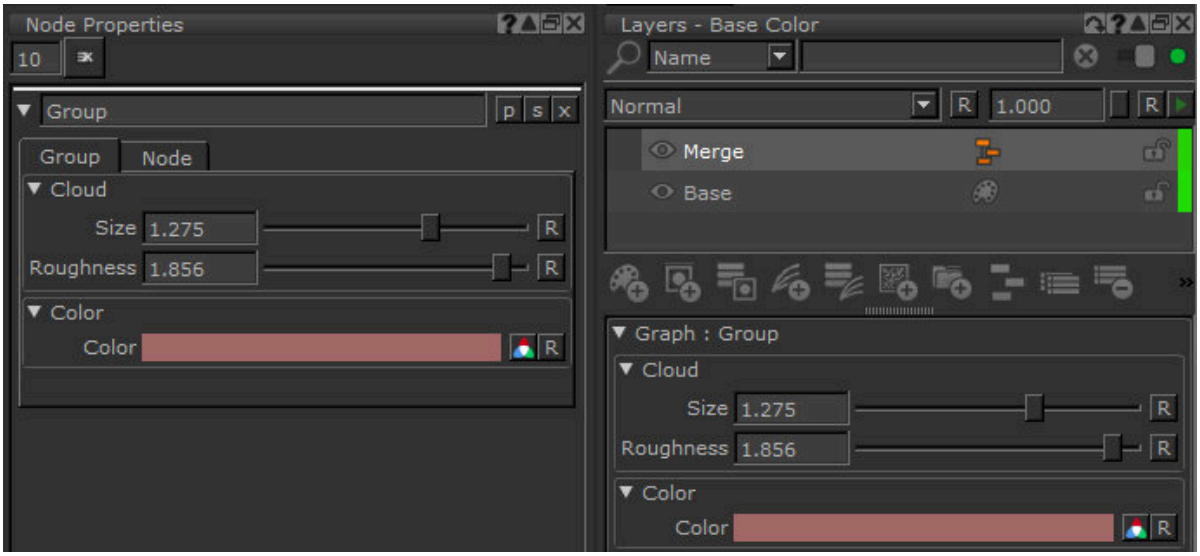
You can expose node properties of a Group node using two methods. One is entering the nodes within the Group node and exposing their properties by clicking the Promote button. The other is exposing node properties at the Group node level using the **Group Node Knobs** dialog.

Group nodes don't display any properties by default, other than those shared by all nodes. See Working with Node Properties for more information.

Exposed properties are displayed in the Group's Node Properties palette and in the associated Graph layer in the Layers palette. For example, given a merged Group in the Node Graph containing a Cloud and a Color node:



You could expose **Size**, **Roughness**, and **Color** properties:



Exposing Controls Using the Promote Button

1. Enter a Group node and then double-click the required nodes within this Group node.

> **Note:** To enter a Group node, **Ctrl** +double-click the Group node to open its subgraph. Alternatively, select the Group node and press **Ctrl**+**Enter** or in the **Node Properties** palette, click the **s** button.

The properties of the nodes display in the **Node Properties** palette.

2. Click the Promote button.

   The Promote button turns orange and the node properties are exposed at the Group node level.
3. Click the Promote button again to unexpose the node properties.

> **Note:** You can link exposed node properties together under one control (see Linking Exposed Properties) but if you unexpose one or more node properties that are part of this control, this deletes the control at the Group node level as the control becomes irrelevant.

Exposing Controls Using the Group Node Knobs Dialog

1. Double-click the Group node's header to display its properties, if they're not already open.
2. Click the **p** button at the top-right of the **Node Properties** palette.
   OR
   Right-click on the Group node and select **Edit** > **Groups** > **Manage User Knobs**.

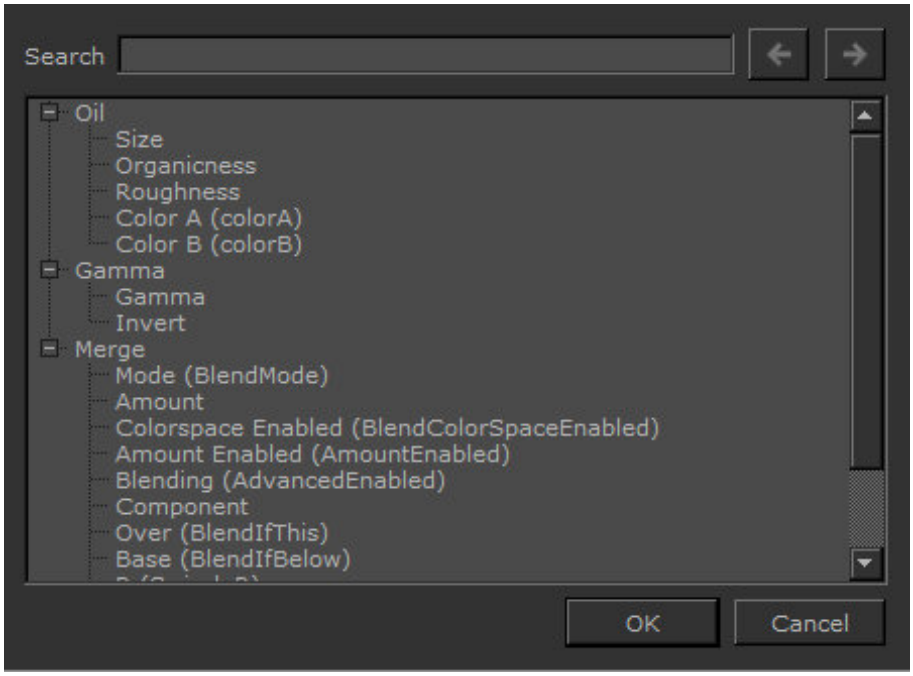
   The **Group Node Knobs** dialog displays.

3. Click **Pick** to begin exposing controls in the **Pick Knobs to Add** dialog.

Note: Mari allows you to search for controls to expose in the **Pick Knobs to Add** dialog. To do so, enter the name or part of the name of a control in the **Search** field and click the left or right arrow. Mari finds all the controls containing the letters you've entered.

Note: You can also do a wildcard search, which is based on character pattern, using the character *.
For instance, if you're looking for Principled BRDF controls that contain the letters s and f in that order, enter **s*f** in the **Search** field, and as a result, the **Subsurface** control displays in the dropdown menu.

4. Select the required control from the list available. You can only select controls that are associated with nodes contained within the Group.

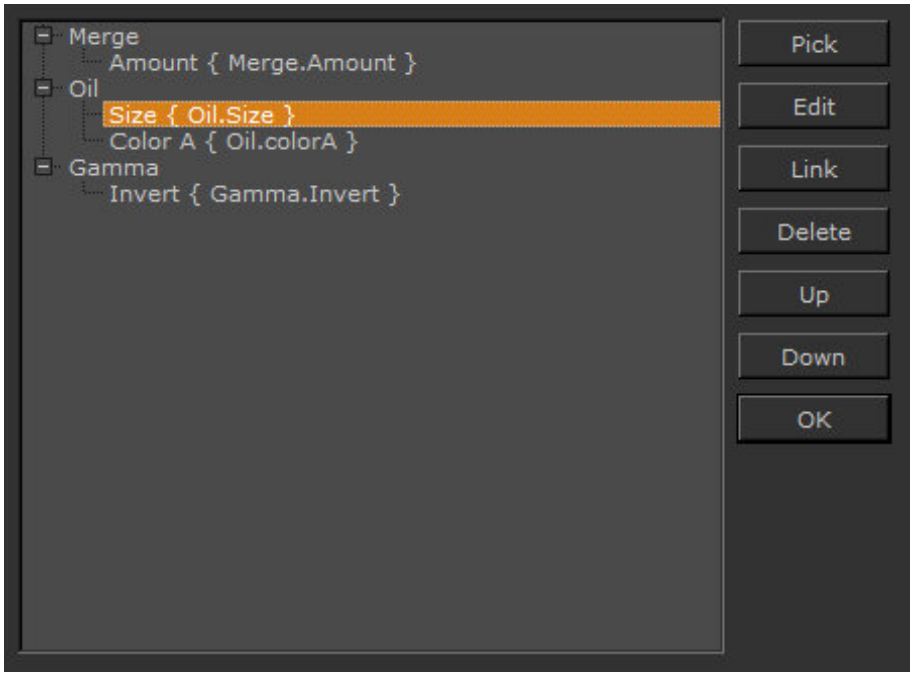


5. Click **OK** to expose the control(s) in the **Node Properties** palette.
   You can expose as many controls as you like in a Group's properties by repeating this process.

Tip: You can expose all the properties of a node at once. To do so, in the Group node, right-click on the required node and select **Edit** > **Groups** > **Expose Node Properties**.

Note: The gizmo values defaults to the ones set in the Group node properties at the time you export a Group as a gizmo. See Exporting Groups as Gizmos.

6. The controls editor has some additional functions to help you organize how controls appear in the **Node Properties** palette.



- **Edit** - click to give the selected control a custom name, though the control itself retains its original function.
- **Link** - click to link the exposed controls you've selected. For more on the **Link** functions, see Linking Exposed Properties.
- **Delete** - click to remove the control from the Group's properties.
- **Up**/**Down** - click the **Up** and **Down** buttons to re-arrange the order in which controls appear in the Group's properties.

Linking Exposed Properties

The **Link** function allows you to link a number of exposed properties together under one control. For example, if you have a **Cloud** procedural inside a group with a number of offsets **X**,**Y**, and **Z**, the **Link** function allows you to scale these offsets uniformly under one exposed control, instead of having to set each of these properties individually.

To link exposed properties:

1. Make sure you have exposed properties in the **Group Node Knobs** dialog. See Exposing Group Properties
2. **Ctrl**/**Cmd**+click to select as many properties as you need and click **Link**.

   The **Link Knobs** dialog opens.
3. Enter a name for the control you want to create, and click **OK**.

   The new control is created.
4. Click **OK** to close the **Group Node Knobs** dialog.

   In the **Node Properties** palette, the exposed properties are linked under one control in the associated group.

Note: If you link exposed properties from the same node, the control displays under the node menu in the **Node Properties** palette. If you link exposed properties from separate nodes, the control displays under a **Links** menu in the **Node Properties** palette.

Note: If you unexpose one or more node properties that are part of a control, this deletes the control at the Group node level as the control becomes irrelevant. See Exposing Group Properties.

Exporting Groups as Gizmos

You can export a Group as a gizmo so that other artists can use parts of your node tree to perform similar tasks. Gizmos are exported using the **.mng** file extension and can be imported into any project in Mari 3.0, or above.

To export a gizmo:

1. Double-click the Group node you want to export to open its properties.
2. Click the **Node** dropdown to expand the view.
3. Click **Export as Gizmo**.

   The **Export as Gizmo** dialog displays.
4. Navigate to the required directory, enter a **File name**, and then click **Save**.

   Mari saves your gizmo (**.mng**) to the specified location.

Note: The gizmo values defaults to the ones set in the Group node properties at the time of export.

Importing Gizmos

To import a gizmo, do the following:

1. Create a **Gizmo** folder in the following location, depending on the platform in use:
   - Windows: `C:\Users\<username>\Documents\Mari\`
   - Linux and Mac OS X: `/home/<username>/Mari/`
2. Copy the gizmo to the **Gizmo** folder, depending on the platform in use:
   - Windows: `C:\Users\<username>\Documents\Mari\Gizmos\`
   - Linux and Mac OS X: `/home/<username>/Mari/Gizmos/`
3. Restart Mari to make the gizmo available.
4. Right-click inside the Node Graph and select **Nodes** > <gizmo_name>.

> **Tip:** You can create a sub-menu in the **Nodes** dropdown, to better organize your gizmos, simply by creating another folder in the **Gizmos** folder. For example on Windows: `C:\Users\<username>\Documents\Mari\Gizmos\Gizmos\`

OR

Right-click inside the Node Graph, select **File** > **Import nodes**, then navigate to the gizmo's location, select the required gizmo and click **Open**.

Mari inserts the gizmo into the Node Graph.

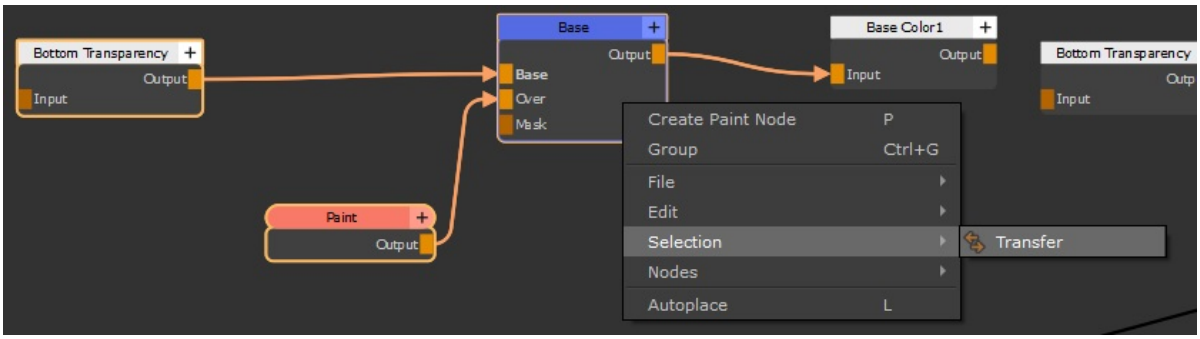
> **Tip:** You can use a MARI_GIZMO_PATH environment variable to set custom locations for loading gizmos at launch.

Transferring Nodes to a Different Object

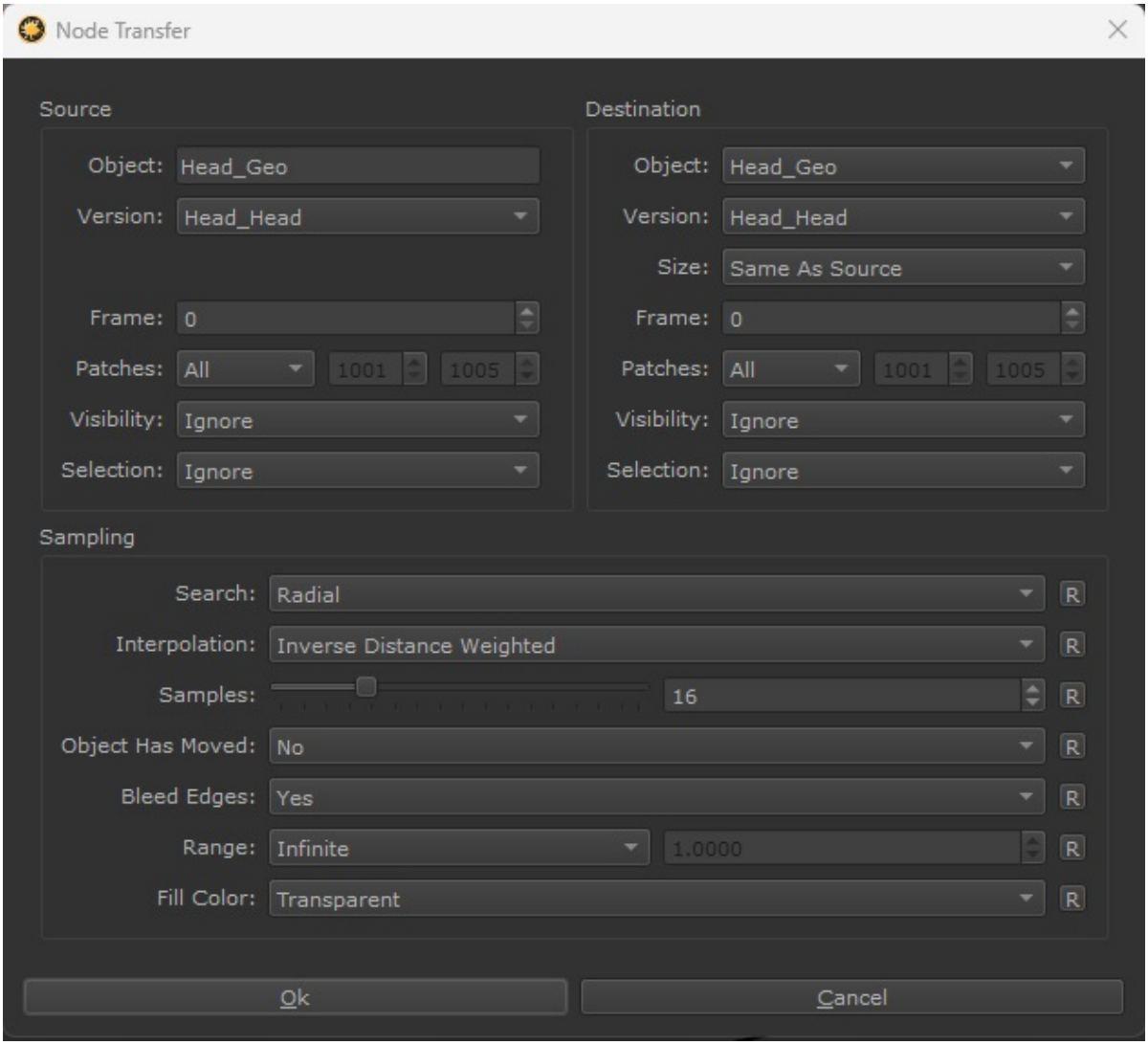
There may be cases when you want to transfer part of the node graph to a node graph that belongs to a different object, so that you can make use of your existing work without the need to start from scratch. This is particularly useful for recreating work on an object that has different topology to the source object, or is identical geometry with a different UV layout.

Mari uses the Bakery to transfers the nodes and paint data to the target object, avoiding the need to manually redo existing work on a new version of the geometry

To transfer nodes, select them in the node graph, right-click and select **Selection** > **Transfer**.



This opens the Node Transfer dialog.



1. Adjust the Node Transfer settings:

Source	Select the **Object**, **Version**, and **Patches** that the node can currently be found on.
Destination	Select the **Object**, **Version**, and **Patches** that the node is being transferred to. **Tip:** For more information on versions, see Adding and Removing Object Versions. **Tip:** For more information on objects, see Objects and Object Hierarchies.
Version	The source and destination versions of the objects for the transfer. For more information on versions, see Adding and Removing Object Versions.
Transfer Method	Specify the method for transferring the texture onto the destination UV map. Choose between: • **UV Remap From Bakery** - Use the Bakery to generate the UV coordinate mappings from the source object to the destination object. • **UV Remap From File** - Use an import-like template to remap pre-generated textures based on UV coordinates. For example, if you have generated the UV map beforehand, you can load it here.
Size	Defines the resolution of each UDIM of the resulting destination paint node image sets. Use the same sizes as the source, or any other supported channel size.
Bleed Edges	Enable to generate edge bleed on the resulting image set.
Fill Color	Define the initial color to fill the resulting image set with. If an object has extra patches that aren't covered by the transfer, then the fill color specifies whether these extra patches are filled with **Transparent**, **Black**, **White**, **Mid-gray**, **Magenta**, **Foreground**, or **Background**.

2. Click **Ok**.
3. The transfers will then be added to the node graph of the object you are transferring to.

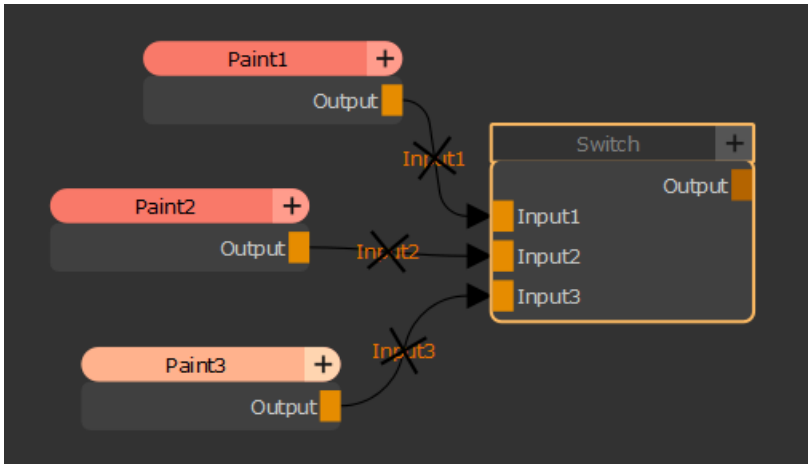
Variations with the Switch node

The Switch Node enables you to create multiple variations of any part of your node graph, and quickly change which upward stream is used with a simple dropdown menu. For example, you could choose between variations such as 'Clean' and 'Dirty' by choosing which input should be actively compiled by the shader.

Connecting nodes

- First, add a Switch Node and add input ports to it:
  You can create inputs in the Switch node's **Node Properties**, finding the **Inputs** list and using the button.
- Set up your node graph so that multiple nodes' outputs are connected to the Switch node's input ports.

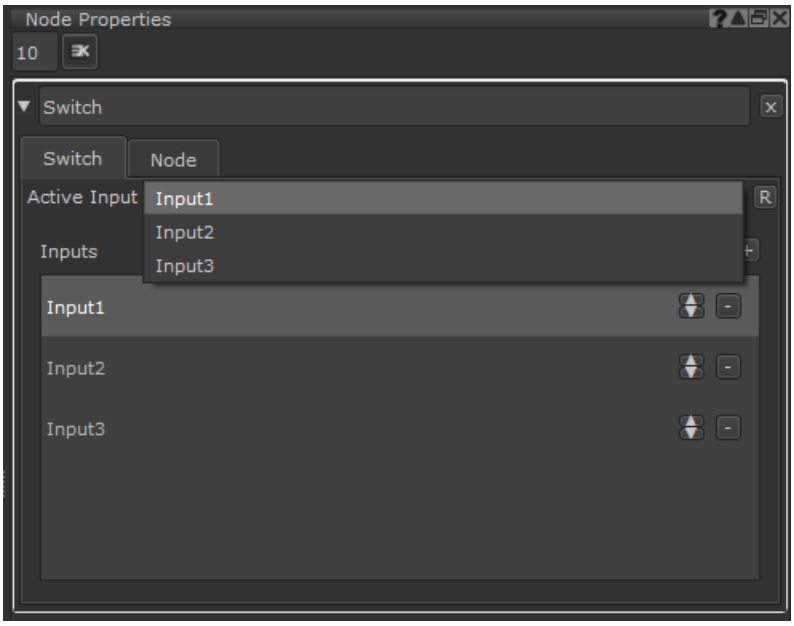
Node connections example (with all inputs inactive):



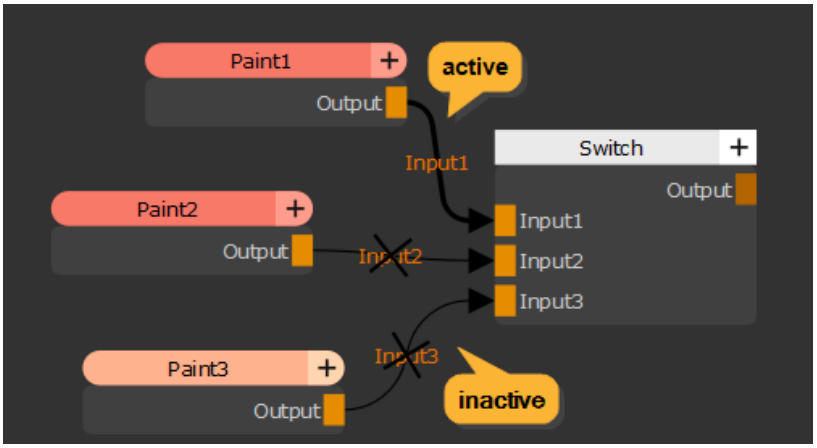
Tip: You can rename any input port in the Node Properties by double clicking on the input names in the Inputs list.

Selecting which input to use

In the Switch node's **Node Properties**, the **Active Input** dropdown menu lists all available inputs and allows you to select the desired input by name.



For example, if you select Input 1, Input 1 will be marked as active and Inputs 2 and 3 will be marked as inactive. Active input connection lines are shown in bold. Inactive connections show a cross on the node connection.



You can use this tool to create variations of materials and quickly switch between them, making your Mari project more versatile and efficient. To more about how to apply this to Materials, see Authoring Material variations using the Switch node.

Tip: Inactive inputs and streams are not evaluated by the shader which can help to improve performance.

Note: Inactive streams may also be referred to as 'disabled' streams. For more information about how to disable streams, go to Working with Nodes.

Baking in Mari With the Bakery

Baking is an essential part of the texture artist's toolbox. It enables the display of detail from high-res (high poly) models on low-res models, thereby requiring fewer computational resources to create. You can also streamline creating certain types of materials and effects without having to repaint each time. Let's say you need an edge wear mask on an object, instead of painting a new mask from scratch each time, you can generate a curvature map as a a basis.

In essence, baking refers to the process of converting and flattening dynamically computed 3D geometry into a 2D texture map. The baked texture map is essentially a frozen snapshot of the surface properties, which can then be applied to a lower poly mesh to generate detail within the shader.

High-resolution 3D source data is often too computationally expensive to use when rendering, texturing, or animating. Instead, a 2D texture map can efficiently draw the same details without loss of fidelity.

Example: Imagine the surface of a game character's skin is cracked and creviced. You can add realism by using ambient occlusion to calculate the shadows that may lie within the cracks. However, calculating these shadows for every frame in a render or animation would require a lot of computational resources. Instead, you can convert this shadow detail into a 2D map known as an Ambient Occlusion map that can be used on the surface of the character to replicate the crack details. The use of a map means that no frame-by-frame recalculation is needed as the map detail is fixed ahead of time.

Capturing Details With Baked Texture Maps

There are many types of mesh maps that can be captured by baking including curvatures, ambient occlusion (AO), and normal maps. This provides a way to collect valuable details such as where shadows lie within ambient lighting conditions, the position of geometry edges, and the thickness of areas within the mesh.

Once generated, the texture map can add surface information and depth to textures and materials, allowing simpler models to display complex visual details without processing heavy geometry. These baked maps can then be used to drive advanced materials, influencing how the textures interact with light and capturing information that may otherwise need to be uniquely painted for each asset.

Note: To find out how to add texture maps to layers, see Importing Textures to Layers.

Mari offers default recipes for the following bake item types:

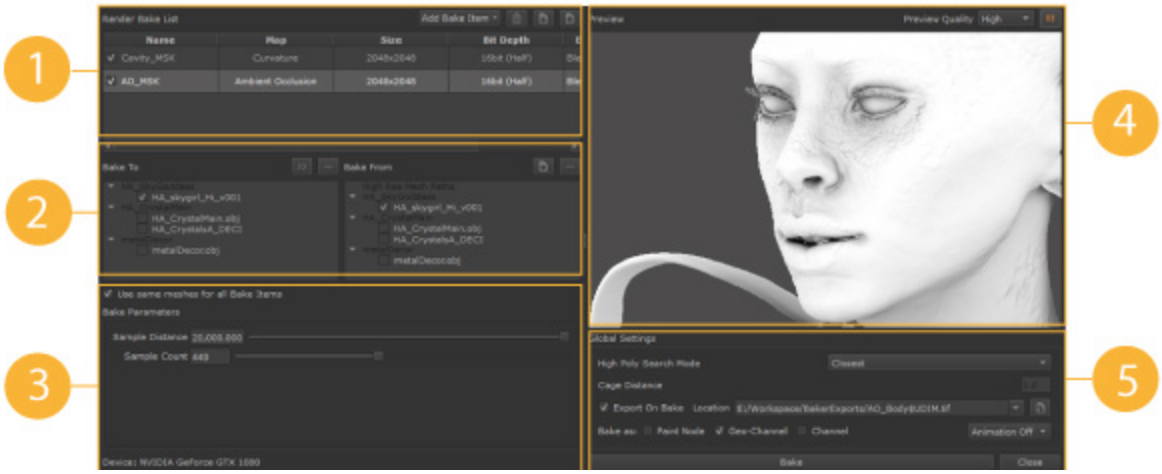
- **Ambient Occlusion** - Simulates how an ambient light source affects an asset, especially in crevices and in areas where the asset will cast a shadow upon itself. This creates soft areas of shadow and light.

- **Displacement** - Captures height information from a high-poly model's surface details, enabling realistic texture and geometry effects on a lower-poly model.
- **Normal** - Encodes the angles of the surface normals from the high-poly model, which gives the low-poly model the appearance of more complex surface detail.
- **Position** - Records the 3D position of every point on the model's surface, providing valuable information for directional shading and material effects.
- **Thickness** - Measures the distance between the top and bottom sides of a surface. This type can be used in subsurface scattering or to simulate various materials such as cloth, skin, or wax.
- **Curvature** - Highlights the edges and curves of a model by analyzing the bending of its surface. This is useful for highlighting details and can be employed to create edge wear effects in texturing.

The Bakery

Mari provides a dedicated tool for baking mesh maps called the Bakery. Click the Bakery button in the top toolbar to open the **Bakery** window.

Bakery Workflow Overview



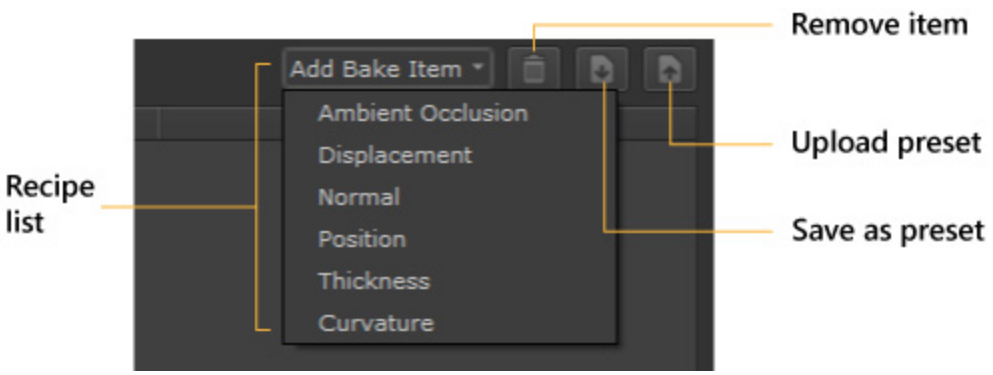
The Bakery offers a simple workflow for baking maps. The numbered steps below correspond to the numbered panels in the UI screenshot above.

1. Populate the **Render Bake List** - Add a list of item types to bake, such as ambient occlusion, and set the texture map's properties, such as size and bit depth.

2. Configure **Bake From** and **Bake To** meshes - Choose your high-res mesh to bake from and your lower-res mesh to bake to.
3. Adjust the bake item's **Bake Parameters**.
4. **Preview** the result for any of the bake items.
5. Choose **Global Settings** - Adjust the general settings and start the bake.

Adding and Editing Bake Item Maps

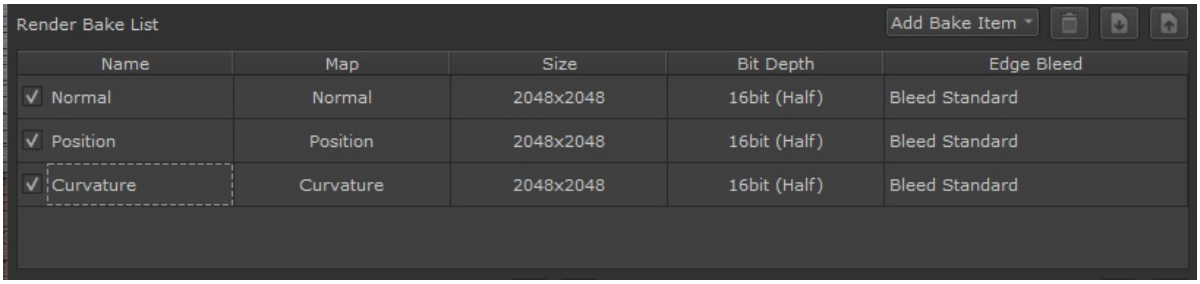
To add an item to bake, select the texture map recipe from the **Add Bake Item** dropdown in the top toolbar of the **Bakery** window.



- To remove an item, highlight it in the list and use the trashcan.
- You can also store and upload presets for later use.

See Capturing Details With Baked Texture Maps for more details on each type.

Adding bake recipes populates the **Render Bake List**.



Every item type shares properties that determine how the texture is generated.

Bake Texture Map Properties

For each item in the list, you can edit the following texture properties:

Note: Map is not editable as it is a fixed property that describes the map type.

Name	**Name** is the map type by default, but you can edit this as needed, particularly if you have multiple maps of the same type.
Size	Choose the resolution of the texture map. There are eight options ranging from 256x256 to 32768x32768 (32K).
Bit Depth	Choose from three possible bit depths to define the range of colors in each channel for the texture map: • **8bit (Byte)** - for 256 shades per channel • **16bit (Half)** - for 65,536 shades per channel • **32bit (Float)** - for HDR
Edge Bleed	Choose the bleed behavior for the edges around the bake textures. • **Bleed Off** - no bleed applied to the output textures • **Bleed Fast** - add bleed but use the fastest and lowest quality algorithm • **Bleed Standard** - add bleed using the slower but better quality algorithm Note: For more information, see Bleeding Across Patch Edges.

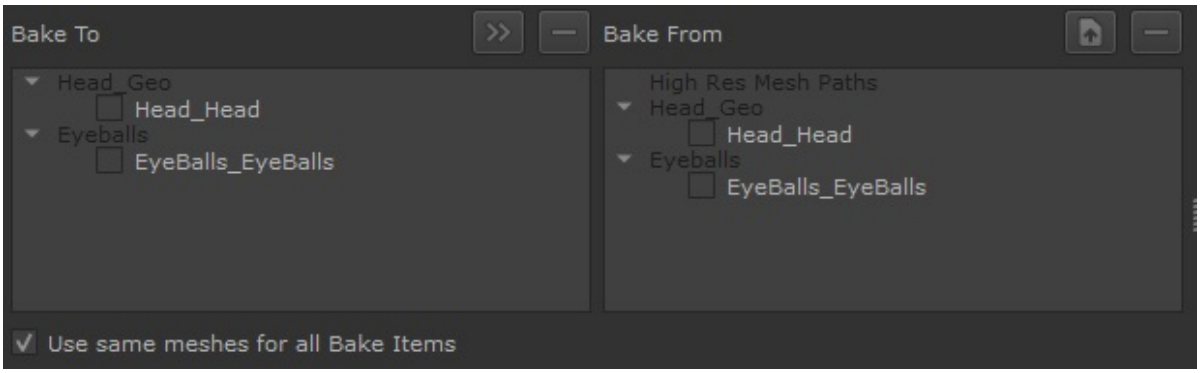
Each item type offers a set of parameters that affect the appearance of the bake result. To see these, click on the item to populate the **Bake Parameters** section.

Choosing Meshes for Baking

When baking, you typically take the 3D data from a mesh, which is baked onto a texture that is applied to a different, usually lower poly mesh.

The two meshes are built with equivalent parts so the bake can work smoothly.

The mesh to bake from and the mesh to receive the texture are show in two adjacent panels:



- The **Bake To** panel displays the lower poly meshes to receive the baked texture.
- The **Bake From** panel lists the higher poly meshes that provide the data for the bake.

Note: There may be a case when the source and the target meshes are the same model. In certain cases, the mesh may contain enough information to generate a useful map. For example, a simple position map.

The meshes in each panel have a checkbox so that you can select which meshes are involved in the baking process.

A typical workflow would follow these steps:

1. By default, the **Bake From** panel lists the same meshes as your project meshes. You can upload a different high-res mesh using the upload button .
2. Select the meshes in the **Bake To** panel to receive the baked items.
3. Propagate the selection from the **Bake To** meshes to the equivalent meshes in the **Bake From** panel using the Copy button .This ensures a match between the selections.

You can select individually, or clear all selections in each panel using the Clear Selection button .

You also have the option to decide which meshes are used in the bake for each item in the **Render Bake List** . Toggle **Use same meshes for all Bake Items** as follows:

- On **-** The same selection of meshes is used for all the recipes in the list. Select this when you don't want to customize the mesh selection on a per-item basis.
- Off **-** Turn off if you want a different mesh selection for each item. This lets you customize the bake per-bake item. For example, when baking an environment, say a kitchen, you many not want a fruit bowl to cast an occlusive shadow on a tabletop. Simply deselect the fruit bowl in the mesh hierarchy to eliminate it from the bake.

Setting Bake Item Parameters

Each bake item type offers key parameters to customize the look of the texture map. When editing the parameters, you can use the **Preview** window to check the results. See Previewing the Bake Item.

Note: A position map has no parameters as the source mesh topology derives the position data.

Ambient Occlusion Bake Parameters

Ambient Occlusion (or AO) is a shading technique that adds depth and realism by simulating how diffused light interactss with objects. It introduces soft shadows in areas with limited light exposure, like creases, holes, and closely spaced parts, which can be beneficial in making a material seem more believable..

Sample Distance	**Sample Distance** s the maximum distance the rays will travel in order to search for occluding surfaces.
Sample Count	**Sample Count** is the number of rays used to determine shading. More samples mean smoother, more accurate shading but take longer to compute, while fewer samples are faster but can look noisier.

Displacement Map Bake Parameters

Displacement in texture painting is a technique that uses a texture map to change the geometry of a surface. Unlike Normal or Bump maps that only simulate depth, displacement maps physically move the vertices to add genuine depth and detail to the model without increasing the polygon count.

Bias	**Bias** helps avoid unrealistic shading near object edges. It prevents too-dark areas by adjusting how close an obstruction needs to be to affect the shading. This makes the shading look more natural.

Normal Map Bake Parameters

Normal maps are used to imitate physical depth on a flat surface without modifying its geometry. It fakes recessions and bumps on the surface, creating the illusion of depth and complexity without adding extra resolution to your model.

Bitangent	Bitangent in normal maps is a vector that works with tangent and normal vectors to form a 3D surface coordinate system. This setting is crucial for properly aligning textures and lighting effects.Choose from: • **Per-Pixel Cross Product** - Calculates the bitangent for each pixel, providing detailed surface information, making it ideal for high-quality renders but at a higher computational cost. • **Per-Vertex Cross Product** - Calculates the bitangent is based on mesh vertices, providing a moderate level of detail and improved performance. This method works well for models that are not primarily detail-focused. • **Mesh** - Uses the mesh's own bitangent data, making it ideal for specific requirements where custom bitangent information is embedded in the mesh.
Invert Y	Check this to reverse the direction of the Y-axis in the bitangent vector. Enabling this

	option helps fix mismatches in normal map orientation between various software or rendering engines. It can help ensure that the textures are displayed correctly.
Normalization	Normalization in baking normal maps ensures all normal vectors are the same uniform length - standardized to a unit vector. This corrects vector lengths for consistent shading, accurate surface detail, and uniform lighting effects.Choose from: • **Per-Pixel** - Adjust the length of each normal vector individually for every pixel. This is more computationally expensive but is better at expressing fine detail. • **Per-Vertex** - Adjust the length of each normal vector individually for every vertex. This results in lower detail but might be sufficient for simpler models or when computational resources are limited.
Space	Choose the coordinate system to use to interpret the normals: • **Tangent** - Use a local coordinate system at each model's vertex, making normals relative to the model's surface. This allows the normal map to adapt to different shapes and movements. **Note:** This option is often used in games for its flexibility with dynamic lighting and object movement. • **World** - use a fixed, overall coordinate system for the entire scene. This option is ideal for static scenes where objects and lights don't move.

Thickness Map Bake Parameters

A Thickness map in capturess the thickness of a 3D model's surface. This data is idealfor simulating subsurface scattering and other realistic lighting effects in materials such as skin or marble that have varying densities.

Sample Distance	The sample distance is the depth range used to measure thickness from the model's surface. This setting determines how deep Mari looks inside the model, affecting the light scattering effects in the final render.
Sample Count	**Sample count** is the number of samples taken within the defined sample distance to determine the thickness at each point. A higher sample count means more points are evaluated, leading to a more accurate and detailed thickness map. However, higher sample counts are more computationally expensive.

Curvature Map Bake Parameters

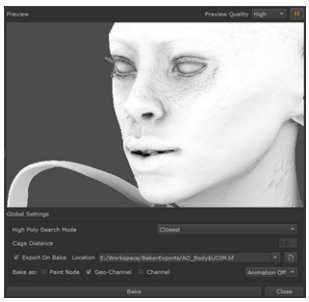
A Curvature map creates a gradient where lighter areas represent convex regions of the mesh and darker areas are more concave. Curvature maps are commonly used as a starting point for creating wear along material edges or for accumulating dirt in mesh cavities.

Metric	Choose a different method (or metric) for calculating and representing the curvature of a surface. • **Tangent Dot** - Calculates curvature by measuring the dot product between surface tangents and a fixed direction, emphasizing curvature changes relative to this direction. • **Angle** - Measures the angle between surface normals of adjacent polygons, highlighting sharp edges and distinct changes in surface direction. • **Normal Dot** - Uses the dot product between normals of adjacent surface points to emphasize broader, smoother curvature changes across the surface.
Multiplier	**Multiplier** adjusts the intensity and contrast of the curvature details to make the effects either more pronounced or subtler. A higher multiplier enhances the highlights and shadows on edges and details for a more dramatic effect. A lower multiplier gives a more subtle enhancement.
Output	**Output** is the type of curvature information to highlight in the baked map. Choosing the right output depends on which aspects of the model you want to emphasize in your texture. • **Curvature** - Generate a map that combines both convex and concave features. The resulting map shows all curvature, highlighting edges and details regardless of whether they curve outward(convex) or inward (concave). • **Convexity** - Focus on highlighting the convex parts of the model, which are the areas that curve outward. This is particularly useful for emphasizing raised details, like bumps, ridges, or outward-curved edges. • **Concavity** - Focus on highlighting the concave parts, which are the inward-curving areas or indentations. This helps in accentuating recessed details, grooves, or inward-curved edges.
Sample Offset	**Sample Offset** controls how far the baking process samples geometry from the model's surface. A larger offset captures broader features, whereas a small offset captures the finer details of the model's curvature.

Sample Count	**Sample Count** refers to the number of samples the baking algorithm uses to calculate the curvature. A higher sample count generally leads to a more accurate and smoother curvature map, as the algorithm has more data points to analyze the surface's curvature. A higher sample count will increase baking time as it is more computationally expensive.

Previewing the Bake Item

The **Preview** panel in the Bakery lets you visually preview the outcome of your recipes with their current parameters to ensure the final bake output is as expected.



When you adjust the **Bake Parameters**, the preview auto-updates.

To show a preview click the play button at the top-right of the Bakery.

You can set the **Preview Quality** to **Low**, **Medium**, or **High**, which in turn affects the preview speed.

Baking and Global Bakery Settings

High Poly Search Mode	When sampling data, choose how to locate the high poly mesh in relation to the low poly mesh. • **Closest** - rays are cast perpendicularly from the low-poly mesh to find the nearest points on the high-poly mesh. This is more suited for simpler models and may result in artifacts for complex shapes. • **Cage** - this looks for the topmost high-poly mesh by casting rays perpendicularly to a 'cage' that encapsulates both meshes. This provides more accuracy when dealing with complex models and when there is a significant difference between the high and low poly meshes.
Cage	The **Cage Distance** is the distance by which the cage extends outwards from the surface of

Distance	the low-poly mesh. It's essential to get this right as too small, and details can be missed. Too large, and this may add artifacts from neighboring geometry. This setting is only active if **Cage** is selected for the **High Poly Search Mode**.
Export On Bake and **Location**	Check **Export on Bake** if you want to save your baked textures to disk. If checked, choose a **Location** using the browse button , or from a list of recent locations.
Bakes as:	Choose how to import the baked result into Mari. You can choose from one or more of the following: • **Paint Node** - Import the result as a Mari Paint Node. • **Geo-Channel** - Import as a Mari Geo-Channel, see Adding Geo-Channels to Objects. • **Channel** - Import as a new Mari channel, see Channels.
Animation	If you have an animated object and want to include multiple frames, select whether to include them in the bake: • **Animation Off** - Don't include additional frame data due to animation. • **Animation All** - Use all of the animated geometry frames in the bake.

Texture Transfers Between Projects

You can transfer existing paint data from one object or object version to another. You can do this by transferring Layers, Channels, or Nodes.

This is useful for re-creating work on an object with minimal changes and similar topology or an object with the same topology but a different UV layout. By using the old textures transferred across onto the new object, you can save time instead of starting again from scratch.

Using the **Transfer** dialog for your chosen texture, you can specify which object, version, and patches to use as a source, and which of these to use as a destination.

In some cases, where the topology of the object might have changed between versions, there may be inconsistencies from the transfer. You can specify a patch Fill Color to highlight where the textures didn't transfer to a patch, and then fix only these patches separately after the transfer.

Mari's texture transfer uses the Bakery engine for a fast and high-performance texture transfer.

Note: Mari's old particle-based transfer system (used before Mari 7.5) is still available through the Python API. Documentation for the old particle system can be found here. It is recommended to use the Bakery transfer system as it is much faster.

You can transfer Nodes, Layers, and Channels. To see specific workflows for each of these methods, go to the following pages:

Transferring Nodes to a Different Object

Transferring Layers

Transferring Channels

Previewing and Rendering

Learn how to preview and render your work in Modo, and how projectors behave in Mari to store specific camera details. Familiarize yourself with the functionality of projectors, and to access baking and rendering features within Modo, including how to save out these renders.

Projectors

In Mari, **projectors** store the specific camera details of a particular view - the rotation, zoom, and orientation of the view, plus the current paintable area, the painting mode, and mask settings. Think of this as being like a bookmark for the exact view you currently have in Mari.

You can use a projector to take “snapshots” of your paint buffer, work on them externally, and then project the changes back onto your model. For example, you can use projectors to easily flip back and forth between Mari and Photoshop®, editing a texture in Photoshop®, then projecting it back onto your model and checking it in Mari.

Once you have created a projector, you can use it to project and unproject on and off the image:

- **Unproject** takes everything visible from the paint buffer and saves it as a file.
- **Project** reads a file and projects it back onto the model.

Unprojecting takes the surface currently visible in Mari and exports it to a file. This includes everything on the surface, just as it appears onscreen - it's like taking a snapshot of the model as you can see it right now. So a particular unprojected file could include parts of several meshes. Once you have unprojected to a file, you can edit the file in your paint editor of choice.

When you're happy with the file, you can then flip back to Mari and project it back onto the view. By restoring the original projector, you can be sure that you're looking at the exact same view as the snapshot was originally taken from. When you select the projector and click **Import** (to the buffer) or **Project** (import and bake), Mari projects the file onto the model in the correct place.

So, once you've set up a particular projector, you can quickly flip between Mari and another editor. You only need to set the target and source files for unproject/project once, and then it's just a click to move back and forth between Mari and your external editor.

To make it easier when you're working on projects with multiple channels, there is also a batch mode for both unproject and project. This lets you unproject multiple channels at once, or project a set of files back on to multiple channels at the same time.

And to make it even easier to quickly project and unproject, there are Quick modes for both. These project and unproject from the current view, without you having to create a specific projector.

There is also support for unprojecting to a layered **.psd** file. As with the standard project and unproject features, you can quickly unproject to a layered **.psd** file without having to create a projector.

In addition, you can also project on to models using **.fbx** files created by 3rd party software such as Maya. Importing the model, cameras, and textures enables you to quickly create textured models using Mari projectors.

Tip: If you only require a single camera view point, use the **Camera** > **Load Camera** option from the Mari menubar.

Projectors also allow you to export a **turntable** view of your model. The turntable takes the model as you can see it through the current shader, and creates a series of images showing the model rotating through an axis. You can include custom text or thumbnails of reference images in the turntable. You can create:

- **Render turntables** - this exports a single channel, and lets you pick the shader and lighting to use.
- **Diagnostic turntables** - these export a set of channels, using the default shader and flat lighting, to help you check the current look.

The **Projectors** palette shows the projectors currently defined for the project.

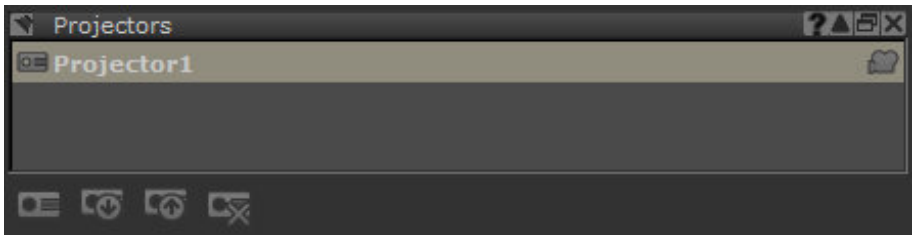
Setting the View

1. Start Mari and open your project.
2. Adjust the view how you'd like it. The projector takes snapshots of the contents of the paint buffer, exactly as they appear onscreen, so:
   - Switch to your preferred camera view (UV, orthographic, or perspective).
   - Orbit, roll, and zoom the model so you can see the area you need to work on.
   - Resize the paint buffer (if necessary) to focus in on the specific area you want to work on.
   - Set the paint blending mode you want.
   - Turn on any masking (edge masking, channel masking, etc.) that you need.

Creating the Projector

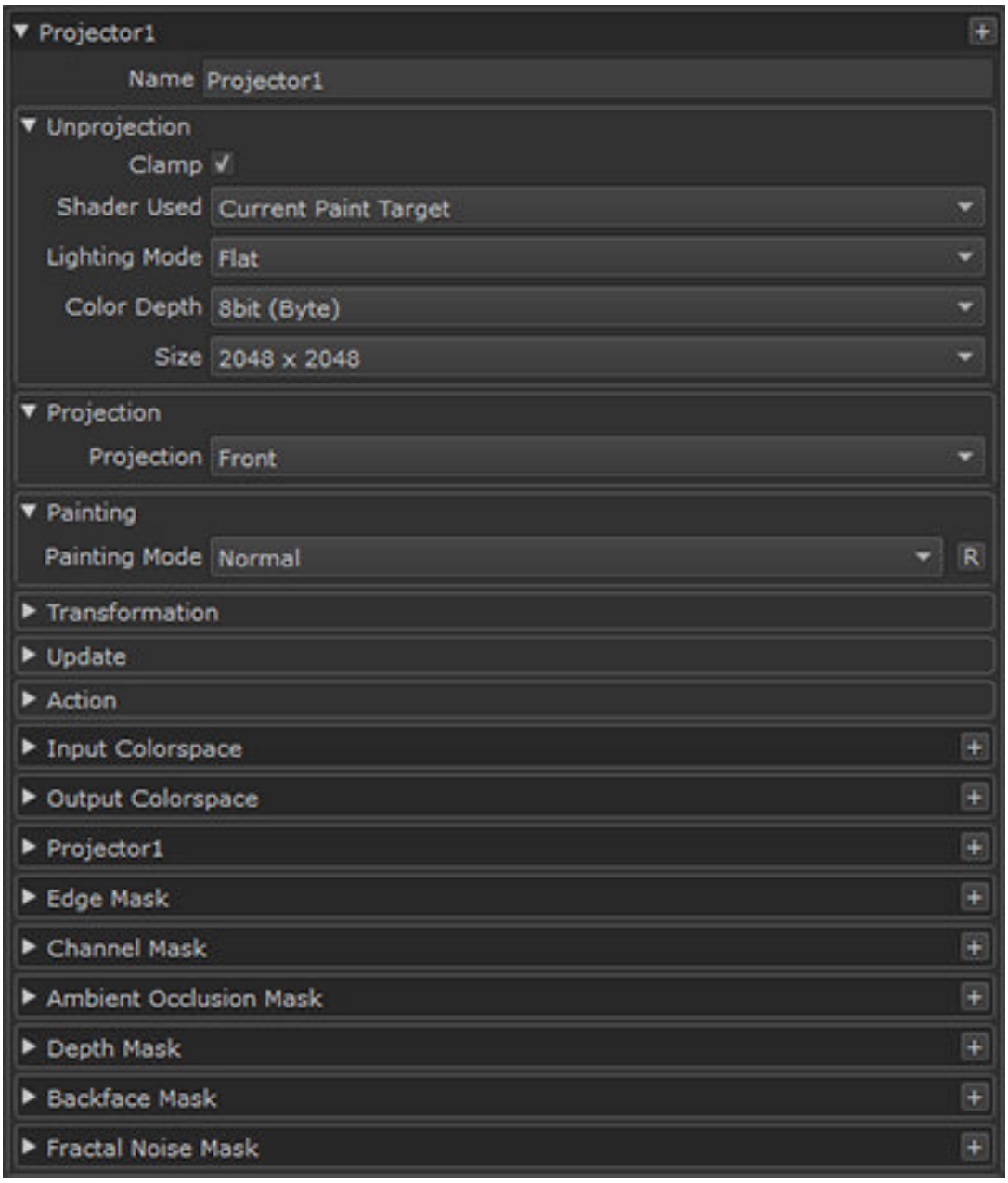
1. In the **Projectors** palette, either:
   - right-click and select **Create Projector** from the dropdown menu, or
   - click the button at the bottom of the projectors list.

   The new projector displays.



2. To rename the projector, double-click on its name in the list and enter the new name.
3. Click to select your new projector.

   The bottom half of the **Projectors** palette displays details of the projector.



4. The settings in the **Unprojection** section control how the projector takes screen snapshots.
   - If **Clamp** is checked, Mari restricts the range of the image to values between 0 and 1.
   - The **Shader Used** defaults to **Current Paint Target**, but you can change this to determine which shader you want to unproject.
   - The **Lighting Mode** defaults to **Flat**, but you can change this.
   - You can set the **Color Depth** and the image **Size** - these default to the settings of your paint buffer when you create the projector.
5. Set the **Output File Path**, to specify where the projector saves the image. You can either:
   - type this directly, or

- click the button.

The **Save File** dialog box opens, allowing you to set the location for the saved files.

6. Navigate to where you want to save the projector, type in the filename, and click **Save**.

> **Tip:** If you want to take shots of multiple channels using this projector, include **$CHANNEL** in the filename. When Mari saves the files the projector produces, it replaces this variable with the channel name for each.
>
> Mari sets the file type (from the listed possibilities) by the file extension you include.

7. Set the **Input File Path**, to specify where the projector looks for updated images to load. As with the output file path, you can type this directly or click to select from the **Pick Path** dialog box.
   You can also include **$CHANNEL** in the input file path.

> **Tip:** Decide whether you want to use the same file for output and input, or use separate files. This depends on how you prefer to work. For example, you may prefer to export to **.tif**, edit the file and save the changes as a **.png** for re-import. Or you may prefer to use the same **.tif** file for both output and input.

> **Tip:** Don't change the settings for **Transformation** or **Perspective**. These show the details of the paint buffer and perspective camera settings when the projector was created (including, for perspective cameras, field of vision and clipping planes).
>
> If absolutely necessary, you can edit these, but it's usually easier to just set the main view to your desired settings and create a new camera.

8. To switch to the view from another projector, either:
   - right-click on the projector in the **Projectors** palette and select **Make Projector Current**, or
   - double-click on the projector in the **Projectors** palette.

The view switches back to how it was when you created the projector, including any masking you had set and the Painting **Mode** that was in use. The name of the projector in use appears on the current tab in the canvas.

Tip: You can also adjust projection mask properties in the **Projectors** palette. Note that if you adjust these properties, you need to click the **Update Only Masks From Projector** button to see the effect of your changes in the canvas. When you do, the equivalent projection mask properties in the **Painting** palette are also updated.

Similarly, you can adjust other global properties such as the Painting **Mode** or **Projection** behavior in the **Projectors** palette. If you do, click **Update Global Settings From Projector** to update the equivalent global properties (including masks) in the **Painting** palette.

Tip: Masks on projectors are hardwired to use the red color component. If you want to use another color component for your mask value, the **Shuffle** adjustment layer allows you to change the RGBA components so that any can be replaced by another. Additionally, the **Luminosity** adjustment layer allows you to convert an RGB input to a grayscale output.

Unprojecting from Your Projector

1. To save a copy of the model as it appears onscreen, either:
   - right-click the projector in the **Projectors** palette, and select **Unproject**, or
   - click the **Unproject** button in the projector details, or
   - click the button to the right of the projector.

   The snapshot saves to the filename specified as the **Output File Path**. If you have not specified an output file path, the **Pick Path** dialog displays, so you can set where the file should go.
2. Mari projects your image in the background. You cannot edit the files until the projection is complete.

   While the projection is still taking place, Mari shows a running man icon at the bottom right of the main window. Click this icon to see a progress box.

Editing the Files Externally

In your preferred editor, open the file and edit it as required.

If you edit the whole snapshot and then project it all back onto your model in Mari, you may get some artifacts.
Typically, you may see a black line at the “edges” where the projected image wraps around the surface.

The original view in Mari, as the projector grabs it.

We edit the whole snapshot, even though we only want to change one small section.

Projected back onto the model, and rotated to show the blurring where the projected image “wraps” around the surface at the nose and cheek.

To avoid this:

- Set your projectors up so they face straight onto the area you're working on.
- When editing the projected image, work on a different layer, only using the areas on the snapshot that you want to change.

This time, we've created a new layer in the snapshot. We're only making changes on the new layer.

Now we're happy with our changes, we remove the background layer and save the file.

Now when we project it back onto the model, Mari only adds the “changes” layer. There are no artifacts, and it all looks good.

Projecting the Files Back onto the Model

Back in Mari, click either:

- **Import** - to load the image from the projector file and put it into the paint buffer, or
- **Project** - to load the image and bake it onto the surface.

Mari loads the updated image from the location set in the **Input File Path** and projects it back onto the model. If you have not specified an input file path, the **Pick Path** dialog displays so you can select the file.

Experiment: Using a sample project, try creating a projector, unprojecting, editing the unprojected file in an external program in some obvious way, and then projecting it back onto your model.

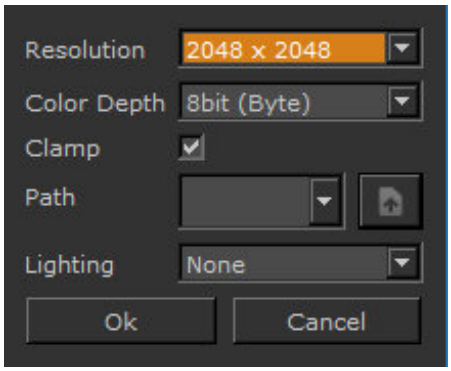
Using the Quick Project Functions

The Quick Project functions provide a fast and simple way to project and unproject from the current view without having to create a projector first.

Tip: When using quick unproject and quick project, be careful not to change your view! If you need to change your view, create a projector as normal, so you can go back to the view and project correctly.

1. To unproject from the current view (that is, the current camera settings and visible shader), from the **Camera** menu, select **Quick Unproject**.

   The **Quick Projection** dialog box displays, allowing you to set a resolution and file path for the unprojected image.



2. Set the details:
   - **Path** - sets the filename and type. You set the image type by typing the image extension (for example, **.tif**, **.exr**, **.psd**, **.png**, **.tga**, and so on).
   - **Lighting** - sets whether the lighting information is output along with the paint when you unproject. If it is, whether it's stored as a **Separate File** or - in the case of **.psd** files - an individual **Layer**.

   If you have already set the quick projection details, Mari just unprojects and updates the file on disk.
3. You can now edit the unprojected image as normal, see Editing the Files Externally.

> **Note:** Due to the nature of **.psd** files, the lighting displayed in your external image editor is an approximation of what you see in Mari.

> **Tip:** When unprojecting a separate lighting layer it appears brighter, but when this is multiplied over an unlit pass in an external application, such as Nuke, the lighting appears as expected.

When projecting, you can either:

- project the image only onto the front patches visible in the current view, or
- project through all patches in the current view.

1. To project onto the front of the patches, from the **Camera** menu, select **Quick Project** Front.

To project...	From the Camera menu, select...
onto the front of the patches	**Quick Project Front**
through all the patches	**Quick Project Through**

Mari loads the updated image from the location set in the **Quick Projection** dialog box and projects it back onto the model.

2. If you want to change the quick projection settings, you can open the **Quick Projection** dialog box - from the **Camera** menu, select **Quick Projection Settings**.

Projecting onto a Model Using .fbx Files

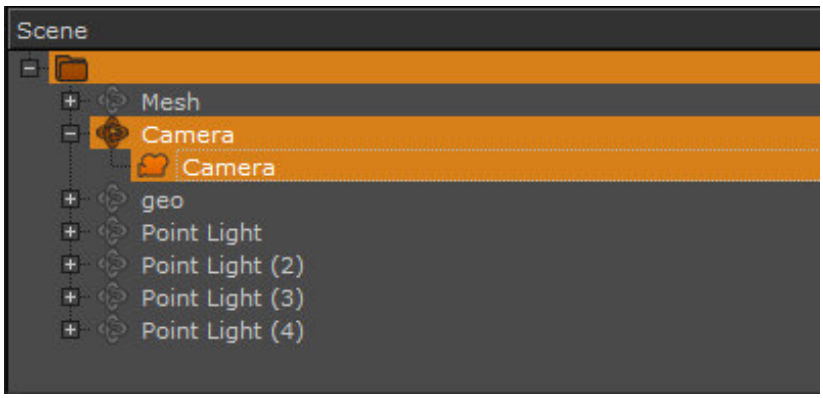
To project onto a model using an **.fbx** file, you need the following files created using a 3rd party application:

- a 3D model (**.obj** format)
- a camera file (**.fbx** format)
- texture files (**.jpg** format for example)

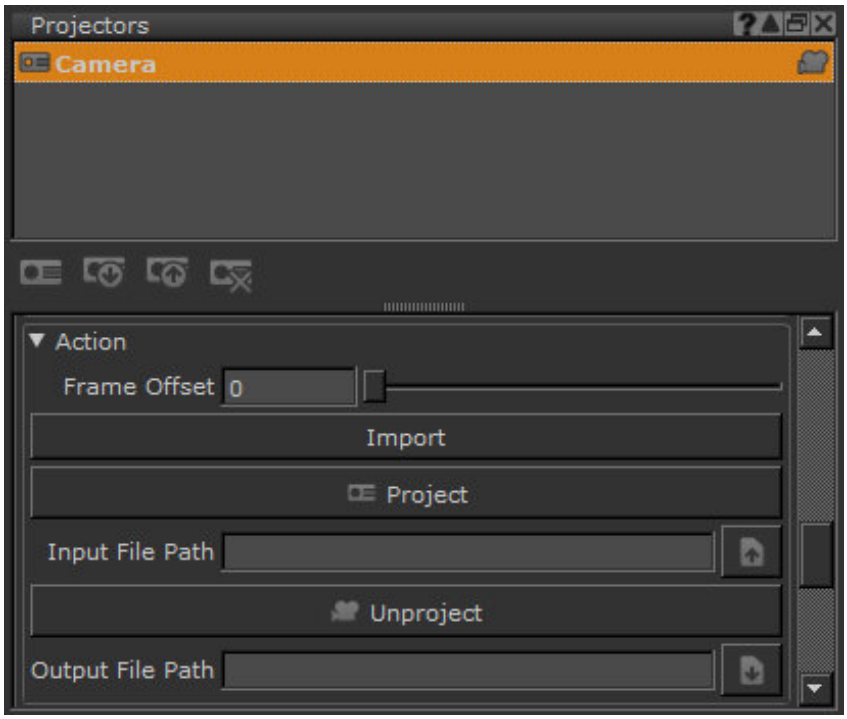
> **Tip:** To make things easier later on, you should use the same naming convention for the camera and texture files when you create them.

1. Start Mari and import your model file.
2. In the **Projectors** palette either:
   - right-click and select **Load Projector**, or
   - click .

   The **Load Projector** dialog box displays.
3. Locate the **.fbx** file associated with your model and click **OK**.

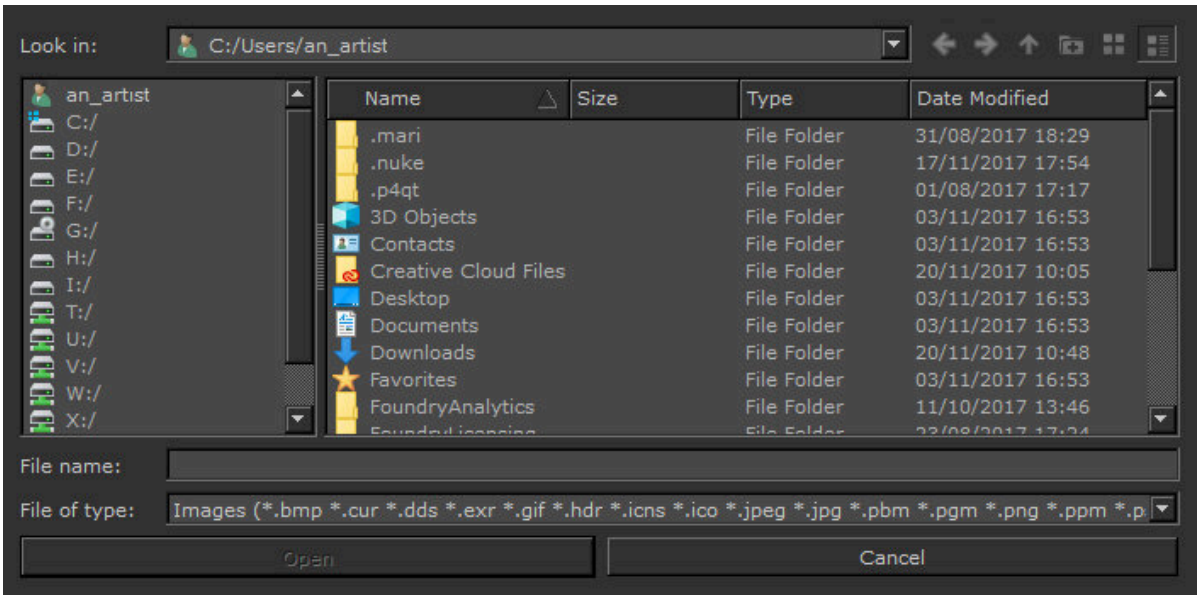


Mari imports the camera data from the **.fbx** file and creates a corresponding number of projectors in the **Projectors** palette.



You can click on the individual projectors in the **Projectors** palette to view the camera angles that you imported.

4. Select a projector in the **Projectors** palette and either:
   - enter the location of the associated texture file in the **Path** > **Input File Path** field, or
   - click to browse for the file location.
5. Select the required file and click **Open**.



You can set the other Projector attributes in the same way as described in Creating the Projector.

6. In **Projectors** > **Action**, select either:

- **Import** - to load the texture from the projector into the paint buffer, or
- **Project** - to load the texture and bake it onto the surface.

Mari projects the texture onto the model as specified.

Once the texture is loaded, you should be able to tell immediately if you've loaded the wrong texture - simply change the file location in **Projectors** > **Input File Path** if this is the case.

7. Repeat steps 5 and 6 for all the imported projectors to complete the process.
8. Once you're happy with the results, you can export your work as described in Exporting Textures from Channels, Generating a Turntable from a Projector, or Saving, Loading, and Deleting Projectors in the same way as regular projectors.

Note: It's worth mentioning that projection using **.fbx** files is not fool-proof and some extra texturing inevitably is required. You can reduce the extra work needed by using more camera angles when creating the **.fbx** file.

Experiment: Try: Loading some textures through the wrong projector. Can you see that the image is incorrect? Loading textures in a different order. Notice how the textures overlay each other?

Generating a Turntable from a Projector

Mari lets you generate two kinds of turntables:

- **Render turntables** - this exports the current channel, using a selected shader.
- **Diagnostic turntables** - these export a set of channels, with flat lighting for each one, using Mari's default shader.

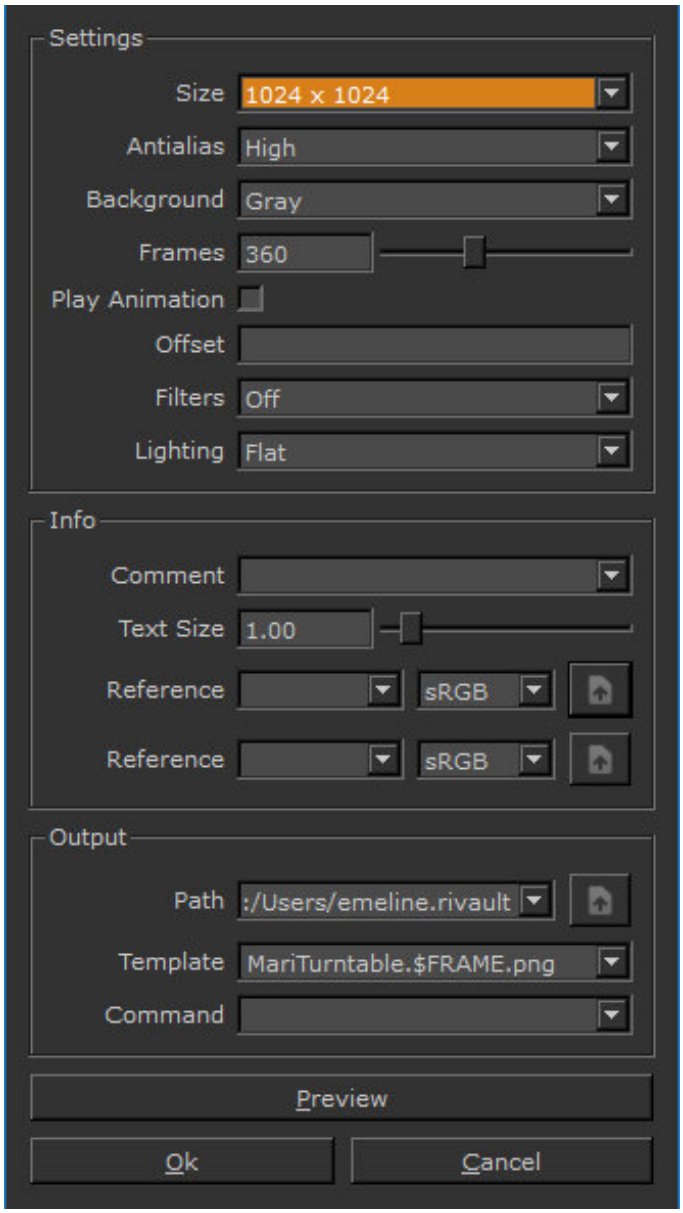
Tip: Turntables always rotate around the center of your current view. That is, the axis for a turntable is a line straight up the middle of the screen in the current view. This is in the center of the visible portion of the model.

You can zoom in on a particular part of a model and create a turntable that focuses on just that part of the model.

Generating a Render Turntable

1. Right-click on the projector and select **Render Turntable** from the dropdown menu.

   The **Render Turntable** dialog box displays.



2. Set the basic image details:
   - **size** of the generated images
   - the amount to **Antialias** the images.
   - **background** for the images - either gray, transparent or the current background in Mari.
   - the number of **Frames** to generate - that is, how many images should be in the sequence.
3. If you are using animated models or textures and want to render the turntable with the animation, check **Play Animation**. This disables the **Frames** control above because the frame range is taken from the Play Controls Toolbar (see the *Play Controls Toolbar* section in the *Mari Reference Guide*).
4. If you are combining multiple turntables, select an **Offset**. This lets you chain together sequences, by offsetting the image numbers.

For example, with an offset of 0, the first image is MariTurntable_1.jpg. With an offset of 360, the first image is MariTurntable_361.jpg. You can use this to generate one sequence of frames for one axis, then move your model around and make another set of frames for another axis.

5. By default, the turntable images use the Example LUT (and any other visual filters you have applied to the view). If you want to create the images without a LUT, set the **Filters** to **Off**.
6. Pick the **Lighting** for the turntable images. You can pick **Flat**, **Basic**, or **Full** lighting.
7. You can also include reference images and text in the turntable images. If you want to include any other information in the turntable files, enter the details:
   - You can set a **Comment** (and change the **Text Size**) - Mari includes this at the bottom of the turntable.
   - You can also select up to two **Reference** images - Mari includes these to the left of your model. Specify the colorspace for each image (sRGB or Linear).
8. Set the **Path** where Mari should create the images, and set the **Template** for the filenames.

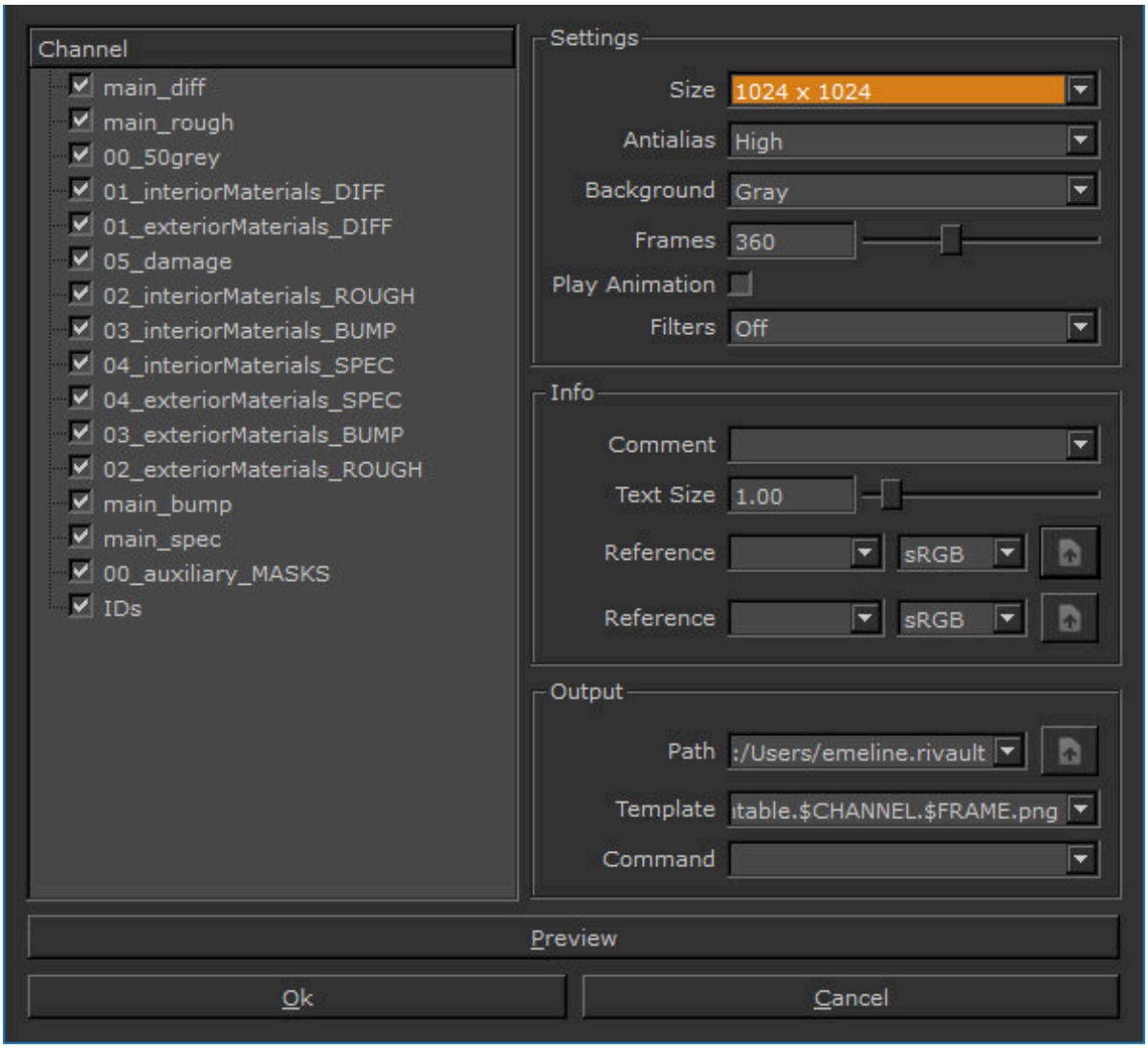
   This must include the $FRAME variable, so Mari assigns the frame numbers correctly to the files. Mari can export turntables as either **.tif**, **.png**, **.jpg,** or **.tga** files - change the file extension in the **Template** to set the file type.
9. Finally, you can enter any shell **Command** that you want Mari to run on the files once it finishes creating them.
10. If you want to check the placement of your reference images and comment text, click **Preview**. This generates the first image of the turntable so you can check the appearance.
11. Click **OK**.

    Mari generates a series of images for the turntable.

Generating a Diagnostic Turntable

1. Right-click on the projector and select **Diagnostic Turntable** from the dropdown menu.

   The **Diagnostic Turntable** dialog box displays.



2. In the **Channel** section of the dialog box, click to uncheck any channels you do not want to generate turntables for.
3. If you are using animated models or textures and want to render the turntable with the animation, check **Play Animation**. This disables the **Frames** control above because the frame range is taken from the **Play Control** toolbar, see the *Play Control Toolbar* section in the *Mari Reference Guide*.
4. Set the other details for the turntable(s) to generate. The properties for diagnostic turntables are the same as for Generating a Render Turntable, except:
   - You cannot set an offset.
   - You cannot set the shader or lighting to use – all images use the default shader and flat lighting.

   In addition, you should ensure that the **Template** includes:
   - The **$FRAME** (**%FRAME** on Windows) variable, so Mari assigns the frame numbers correctly to the files.
   - The **$CHANNEL** (**%CHANNE**L on Windows) variable, so Mari includes the channel in the filename. If this is not present, and you export multiple channels, the later channels overwrite the earlier ones.
5. Click **OK**.

Mari generates a series of images for the turntable, once per channel. Generating turntables for a large number of channels takes a while. Mari shows the progress in the status bar at the bottom right:

Saving, Loading, and Deleting Projectors

1. To save a projector to disk, in the **Projectors** palette, either:
   - right-click on an existing projector and select **Save Projector** from the dropdown menu, or
   - click on the button.

   Mari displays a **Save** dialog box, for you to select a location on disk and set the projector's name.
2. To save all the projectors in the project as a single file, right-click in the **Projectors** palette and select **Save All Projectors**.

   Mari displays a **Save** dialog box, for you to select a location on disk and set the file name.
3. To load a saved projector file, in the **Projectors** palette, either:
   - right-click and select **Load Projector** from the dropdown menu, or
   - click on the button.

   Mari displays an **Open** dialog box, for you to find the projector file. This can be for a single or multiple projectors.
4. To delete a projector, either:
   - in the **Projectors** palette, right-click on the projector and select **Remove Projector** from the dropdown menu, or
   - click on the button.

   Mari removes the selected projector.

Previewing and Rendering

The Modo Render script provides the ability to retrieve renders of your model from Modo, and to display them in the Modo Render palette. It also serves as an example for how you can integrate renderers inside

Mari for a look development workflow. The palette enables you to link your Mari channels to the Modo Effects, and to even bake data from Modo into your channels.

The Modo Render script and the related features that rely on Modo to function do not require you to previously have Modo installed. A bundled command line version of Modo ships with Mari for the purposes of these features and is the default version of Modo selected, unless you specify otherwise.

If you do have a copy of Modo already installed, you can tell Mari where to find this copy and specify whether you want it to use this version or the shipped version of Modo.

Video: To learn how to send a Mari project over to a locally installed copy of Modo for rendering, watch Sending a Mari Project to Modo for Rendering.

Warning: Both the previewing and rendering functions allowed with the Modo Render script do not work with Ptex projects. Ensure that you are using Alembic, FBX, or OBJ geometry with UV coordinates before attempting to use these features.

Rendering with Modo

The Modo render script provides you with options so that you can link Mari channels to specific Modo Effects, preview the results, and/or render your textures by sending them to Modo in a flattened state or as a whole.

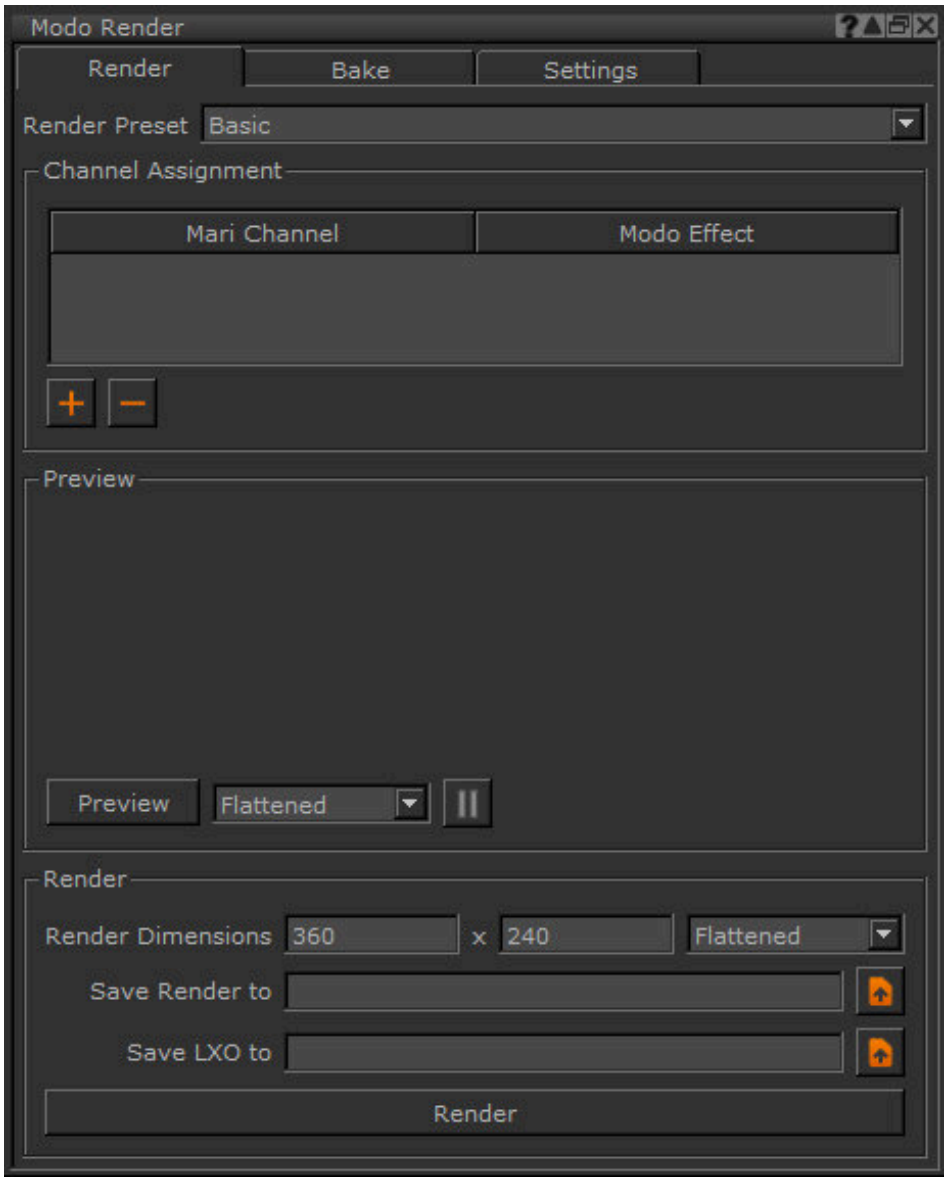
The preview function sends geometry, textures, and other data to Modo, and tells Modo to preview-render your Mari scene with channels linked to Modo shader effects. If you've specified a file location, Modo saves an **.lxo** file representing the scene. There are two modes for previewing your model: previewing in a **Flattened** or **Not Flattened** state. Previewing, either in a flattened state or as is, produces a full preview window in the palette, which shows you what the texture looks like in Modo. At the same time, Modo is opened and the texture is sent to the application.

The render function sends geometry, textures, and other data to Modo, and tells Modo to render your Mari scene with channels linked to Modo shader effects. If you've specified file locations, Modo saves the rendered image in a **.png** or **.exr** file, and the scene in an **.lxo** file. There are also two modes for rendering your model: rendering in a **Flattened** or **Not Flattened** state. Rendering, either in a flattened state or as is,

does the same thing as the preview functions do, but also allows you to save the render to a specific location or save the scene as an **.lxo** file to a specific location.

To send your textures and other data to Modo for preview in the **Modo Render** palette:

1. Open the **Modo Render** palette and select the **Render** tab, if it's not already selected.

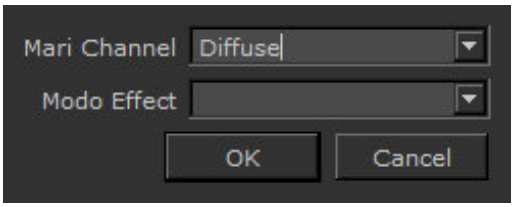


2. Choose a preset from the **Render Preset** dropdown. The preset allows you to set the render settings you wish to use for previewing and rendering.

   By default, the **Render Preset** dropdown is set to **Basic**, which allows Mari to tell Modo to render using Modo's default settings. However, you can set this to **None**, if you prefer not to use a preset.

3. Under the **Channel Assignment** section, click the **Add** button to specify the Mari channels and Modo effects to link.

   The **Assign Channel** dialog appears.



Tip: You can only link one channel and one effect at a time, however, you can add as many channel assignments as you want by repeating step 3.

4. Choose the **Mari Channel** and **Modo Effect** you want to link to one another and click **OK**.

   The channel assignment is shown in the table and, if you have multiple channel assignments, you can scroll though the table to see which assignments are in the list.

5. Specify **Flattened** or **Not Flattened** for your preview settings and then click **Preview**.

   If your project has a lot of UDIMs or high-resolution textures, flattening can take longer than previewing as is. However, previewing **Flattened** saves time when sending lots of commands to Modo, so it may be faster for lighter projects.

   Either option produces a full preview window in the **Modo Render** palette and sends the texture to Modo. Previewing is nearly real-time, so camera movements made on the canvas also occur in the preview window automatically. If you have clicked **Preview** already and proceed to make changes to your channels or channel assignments, you need to click **Preview** again in order for Mari to send new channel data and assignments to Modo.

   The camera settings and the field of view for the preview are taken from Mari.

6. Click the pause button if you need to stop camera updates being sent to Modo.

   When you click this, any camera changes you make in Mari, especially those stemming from navigation in the canvas, are not automatically updated in the Preview pane. These updates won't be taken into account and sent for preview until you toggle the pause button again.

Note: When a preview is active (not paused) only camera updates are continually sent to Modo. If you are navigating around the model and continuing to paint, the paint isn't updated in the preview. To see these changes, you need to **Preview** the scene again.

Previewing doesn't save out the final rendered image, though it does still send textures and data to Modo. If you have finished with your channel assignments, and you are ready to render your final scene:

1. Follow steps 1-4 in the previewing steplist above.
2. Specify the **Render Dimensions** in pixels.
3. Specify whether you want to render **Flattened** or **Not Flattened**, depending on which option is better for your project. If your project has a lot of UDIMs or high-resolution textures, flattening can take longer

than rendering as is. However, flattening the channels saves time when sending lots of commands to Modo, so it may be faster for lighter projects.

4. Specify the location to which the final rendered images is saved on render in the **Save Render to** field or browse to it with the file browser button. This is only the rendered image file. Modo saves the rendered image in a **.png** or **.exr** file format.
5. Specify the location to which the Modo scene is saved as an **.lxo** file on render in the **Save LXO to** field or browse to it with the file browser button. This allows you to open the **.lxo** file in Modo at a later point after the render has been completed.
6. Once you've set your render settings, click **Render**.

   Both **Flattened** or **Not Flattened** options produce a full preview window in the **Modo Render** palette and sends the textures and other data to Modo. It also saves the rendered image file and the Modo scene in the locations specified in **Save render to** and **Save LXO to**.

Video: To learn how to use the Modo renderer inside of Mari, watch Using the Modo Renderer.

Baking from Modo

Note: You can now use the Bakery to bake texture maps, eliminating dependency on external apps. For more information, seeBaking in Mari With the Bakery .

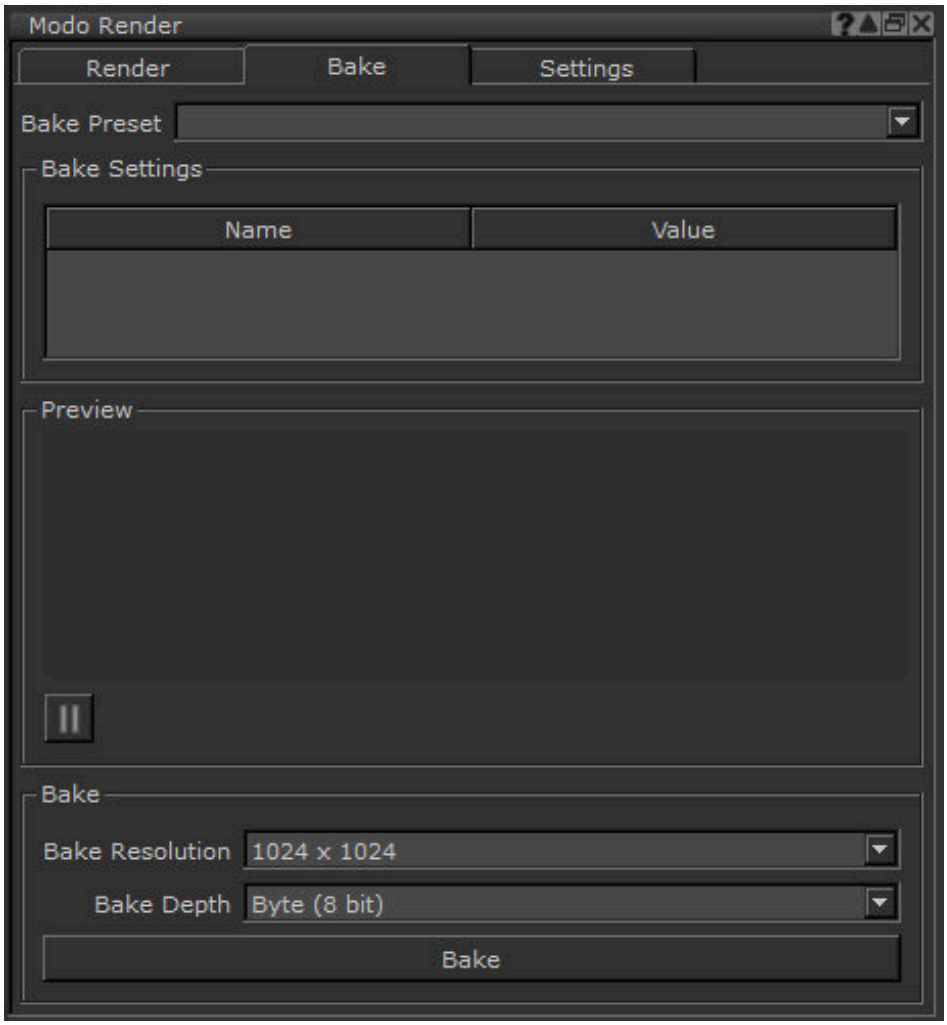
Where rendering is producing a final image of the scene you've been working on, baking allows you to bake data from Modo into your Mari channels. For example, you can bake a specific component, such as ambient occlusion produced in Modo, into a channel in Mari and continue working with it as a baked texture.

Article: Modo's Command Line Render Engine is not bundled with Mari 4.6v3 on macOS. Please consult the following Knowledge Base Article on how to connect Mari with a separate Modo installation.

Before you bake the Modo data, use the preview functionality in the **Bake** tab of the **Modo Render** palette to check your work and ensure you are happy with the results before baking the data.

To bake a texture in Modo and bring it into Mari:

1. Open the **Modo Render** palette and select the **Bake** tab, if it's not already selected.



2. Choose a preset for bake from the options in the **Bake Preset** dropdown. There are presets shipped with Mari listed in the dropdown. You can also create your own presets in Modo.

   The preset parameters are listed in the table, and you can edit them directly within the palette. When you edit a parameter, this change is sent back to Modo and is reflected in the preview shown in the palette as well.
3. Set the **Bake Resolution** for the texture quality.
4. Set the **Bake Depth** for the color depth of the channel into which Modo is baking.
5. Click **Bake** once you are happy with all the parameter settings and have previewed the texture in the **Bake** tab.
6. If you have previewed the scene in the **Render** tab, you can pause the updates in the **Bake** tab as well by clicking the pause button. For more information on the pause button, refer to the Rendering with Modo section.

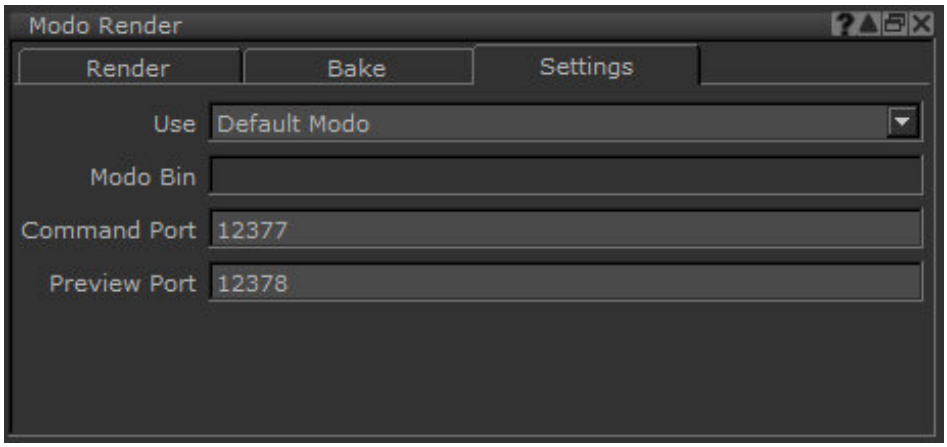
Video: To learn how to use the Modo Bakers that have been added to Mari, watch Using Modo Bakers.

Modo Render Settings

In order to perform the previewing, rendering, and baking functionality that relies on Modo, you need to specify a few settings. You can do this directly within the **Modo Render** palette, instead of needing to check the **Preference** dialog.

To adjust your settings:

1. Open the **Modo Render** palette and select the **Settings** tab, if it's not already selected.



2. Set **Use** to tell Mari whether to use the **Default Modo** that ships with Mari (the command line version) or a **Locally Installed Modo** to which you already have access.

Note: If using a locally installed version, Modo 901 and above is required for rendering and previewing to work correctly.

3. If you chose to use a locally installed version of Modo, specify where Mari can find the application by entering the file path into the **Modo Bin** field.
4. Enter the **Command Port**, on which Mari sends commands to Modo.
5. Enter the **Preview Port**, on which Mari receives previews from Modo.

Tip: By default, Mari looks for preset files in **Mari8.0v1/Bundle/Media/**. You can change the path where presets are searched by using the MARI_MODO_BAKE_PRESETS and MARI_MODO_RENDER_PRESETS environment variables. For more information on environment variables, refer to the Environment Variables That Mari Recognizes section.

Exporting, Importing, and Session Scripts

Learn how to export textures from and import textures into your project, as well as utilizing session scripts to enable you to share your session (project) with other users and to allow them to import the data into another session. Familiarize yourself with the workflows for import and exporting textures, be they layer or channel texture data, or session scripts.

Video: To learn how to export a Mari project as a session and then import that session information into a new project, watch Working with Mari Sessions.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Exporting and Importing Textures

You can export and import both channels and layers in your project, both in 'flat' formats, such as **.png**, and in layered formats, such as **.psd** or **.psb**. This is especially useful when exporting multiple layers or channels that you don't want to take up a lot of space, or when you want to use the final output of a channel in another application.

Importing is how you can pick up an existing set of textures to work on. Similarly, exporting is how your work moves on into the next step in the pipeline:

- Importing loads images into the current channel or layer, or as new channels and layers on your model.
- Exporting saves out the paint on your model as a set of image files (one file per UV patch).

Exporting and importing can be done for both a single channel or layer, as well as multiple channels, layers, even objects in a project. In essence, you can:

- export or import the entire channel (including all layers),
- export or import selected layers, and
- export all channels or objects in a project, or all layers in a layer stack.

Mari supports standard image formats for export and import (such as **.tif**, **.exr**, **.psd**, **.png**, **.jpg**, **.tga**, and **.dds**). You can find a full list of supported file formats in the import and export dialogs.

Note: When exporting **.dds** files, they should be encoded as 8.8.8.8 ARGB files to avoid any complications when bringing them into Mari.

When you create a set of new channels, either when creating your project or when using the **Channel Presets** button, you can import a set of existing textures. See Channels for more information.

Note: As Mari supports **.ptx** files, it is also possible to import and export Ptex files in Mari. This can only be done for a Ptex project. However, unlike the steps described in this section, importing and exporting Ptex files behaves a little differently. For more information on how to import and export Ptex files, see Exporting a Ptex Channel or Layer or Import a Ptex Channel or Layer in this guide.

Importing Textures to Channels

To import into your current channel's layer stack:

1. Open your project.
2. Select a channel from the **Channels** palette and do one of the following:
   - from the **Channels** menu, select **Import** > **Import into Current Channel** to import textures into the current layer stack of the selected channel.

     OR
   - from the **Channels** menu, select **Import** > **Import into New Channel** from the **Channels** menu to import textures into a newly created channel.

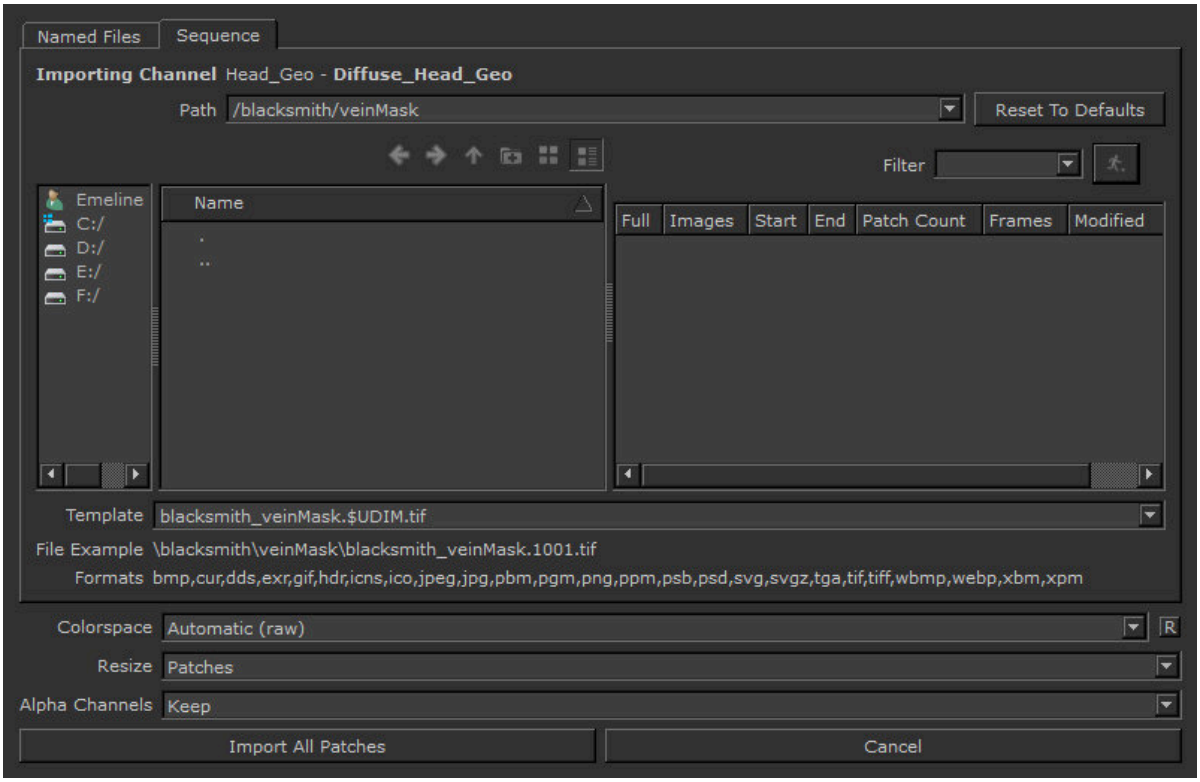
     The **Add Channel** dialog box displays. Follow the instructions for setting up a new channel in Creating and Deleting Individual Channels, and when you've finished, click **Ok**.

Tip: The **Import** menu options are also available from the right-click dropdown menu on the canvas and in the **Channels** palette.

Note: By default, if your model has more than 20 patches, the **Named Files** tab does not display.

To change this, set the **Patch Cutoff** preference to a number greater than the number of patches in the file. Alternatively, you can use the **MARI_NAMEDFILEIMPORT_SEQUENCELIMIT** environment variable to override the preference setting. Please see the section on Environment Variables That Mari Recognizes for more information.

The **Import** dialog box displays.



Note: The Walk button is toggled off by default. When enabled, Mari looks in all folders of the selected location (refer to the **Path** field in the **Import** dialog) and displays all folders and files in the right pane.

Importing Individual Files

To individually pick which files to load for each patch:

1. Click the **NamedFiles** tab.

The tab shows all the patches on the object.

2. For each patch, set the file you wish to import.

   You can either:

   • type the file path and name directly, or

   • use the open button to open a navigation window to find the file.

You can now set resize details and import the textures.

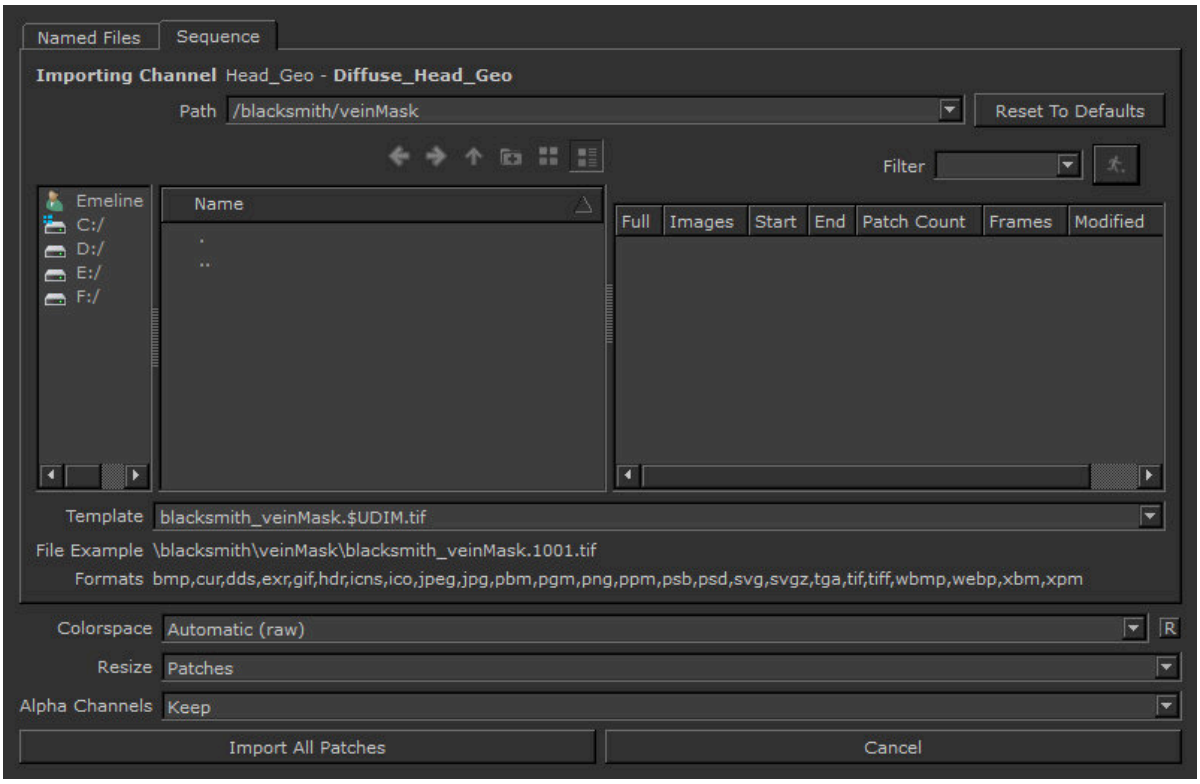
> **Note:** When the image source resolution is non-square, or unsupported by Mari, the image is resized to the nearest power-of-two square size bigger than the original. You are informed of the resize change in the **Image Import Summary**, and is further detailed when **View Summary** is turned on.

Importing a Sequence of Files

If your patches have a consistent naming/numbering system, you can import them as a sequence.

1. Click the **Sequence** tab.

   The tab shows all the patches on the object.



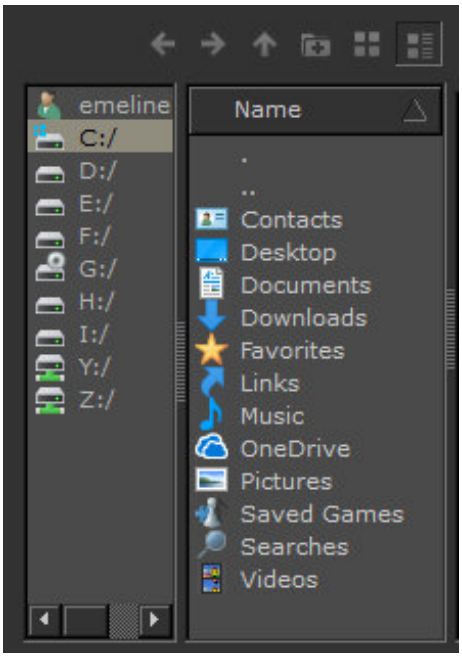
2. Enter the location of the files to import in the **Path** field.

Note: This isn't the name of the files themselves. It's the name of the directory in which the sequence of files exists.

Warning: Make sure you check the path and name carefully, especially if you have been swapping between assets - Mari defaults to the last place you've exported. Don't risk overwriting your textures!

You can either:

- enter the full path to the directory containing the files (or pick from the dropdown), or
- use the navigation controls to navigate to the directory:



When you've selected the path, the file area shows all the existing image sequences under that path.

3. If you are looking for a specific folder or file, you can use the **Filter** field to specify a name or file format to search for.

> **Note:** The Walk button is toggled off by default. When enabled, Mari looks in all folders of the selected location (refer to the **Path** field in the **Import** dialog) and displays all folders and files in the right pane.

4. Either use the default filename template or set a **Template** for Mari to use when gathering the names of images in the selected sequence.

   You can either:

   - Edit this line manually. As you edit, the **FileExample** updates with the name of an example file in the sequence.
   - Click on an existing set of images in the main list. The **Template** automatically populates with the corresponding values.

   If you edit the line manually, you can use any of the following variables, which Mari replaces with the appropriate values when it imports the images:

   - **$ENTITY** - the name of the object.
   - **$CHANNEL** - the name of the channel.
   - **$LAYER** - the name of the layer.
   - **$NODE** - the name of the node.
   - **$UDIM** - the UDIM patch value.

- **$FRAME** - the animation frame number.
- **$NUMBER** - the number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28").
- **$COUNT** - the total number of files in the channel.
- **$COLORSPACE** - the chosen colorspace for the selected images.
- **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").

Note: The steps above assume you are importing a sequence of patches with the UDIM number (1001, 1002, and so on.) in the file names. For more information on how the UDIM patch values are allocated, please see the section on Using UDIM Values.

Note: If you choose to import a layered **.psd** file, layers are imported directly into the selected channel. If attempting to import layers with the same name, Mari asks whether to update the layers or create new ones. You can now set resize details and import the textures.

5. Set the **Colorspace** for data export. By default, Mari selects the colorspace determined by the value set in **Default Import Colorspace** in the **Mari Preferences** dialog under the **Color** tab.

   See Mari Preferences Dialog for more information.
6. Set how Mari handles patch/image size mismatches in the **Resize** field. You can select either to resize the **Images** to match the patches, or to resize the **Patches** to match the images.
7. In the **Alpha Channels** field, choose to either **Keep** or **Remove** the alpha when importing channels. This can help decrease the resulting file size when exporting.
8. Click **Import All Patches** to import the textures or, if at least once patch is selected, this appears as **Import Selected Patches**.

   Mari imports the images and loads them into the layer stack of the selected channel.

Note: For more information on setting the options for a newly-created channel, please see Creating and Deleting Individual Channels.

Importing Textures to Layers

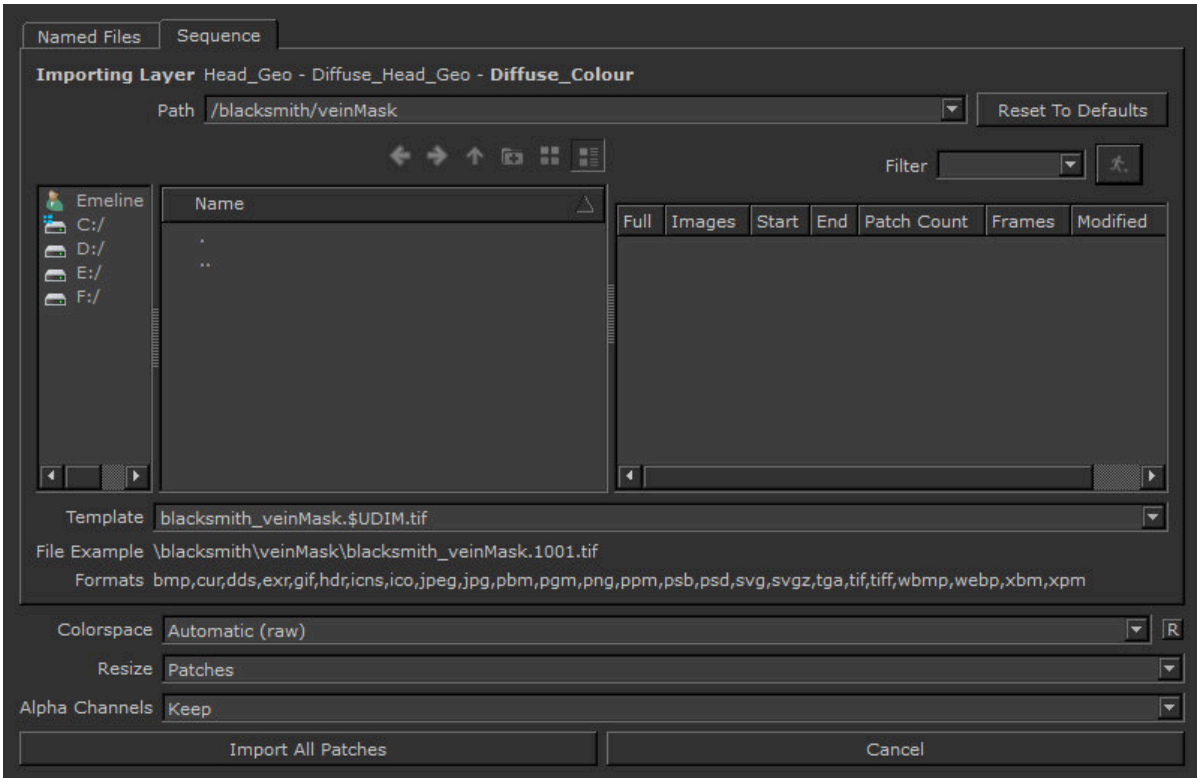
It is possible to import your own existing texture maps to layers in Mari. This is useful if you have baked out texture maps, such as normal or displacement maps, from a sculpting application like ZBrush, before reducing the poly-count or resolution of your model. By importing the texture maps you are able to keep the same level of detail from your original sculpt, but instead of communicating the detail through polygons, you are using texture maps.

To import a texture map to a layer:

1. Start Mari and open your project.
2. To import, select a layer from the **Layers** palette and do one of the following:
   - from the **Layers** menu, select **Import** > **Import into Current Layer** to import textures into the currently selected layer, or
   - from the **Layers** menu, select **Import** > **Import into New Layer** to import textures into the layer stack as a newly created layer.

Tip: The **Import** menu options are also available from the right-click dropdown menu on the main canvas and in the **Layers** palette.

The **Import** dialog displays.



Note: The **Import into Current Channel/Layer** options occur in both the **Channels** and **Layers** menus, and the right-click menus in either palette. The result of importing textures into the selected channel's layer stack occurs regardless of which menu option you use.

Video: Have a quick look at the Importing texture maps into Mari video.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Importing Individual Files

To individually pick which files to load for each patch:

1. Click the **NamedFiles** tab.

   The tab shows all the patches on the object.

2. For each patch, set the file you wish to import.

   You can either:

   • type the file path and name directly, or

   • use the open button to open a navigation window to find the file.

You can now set resize details and import the textures.

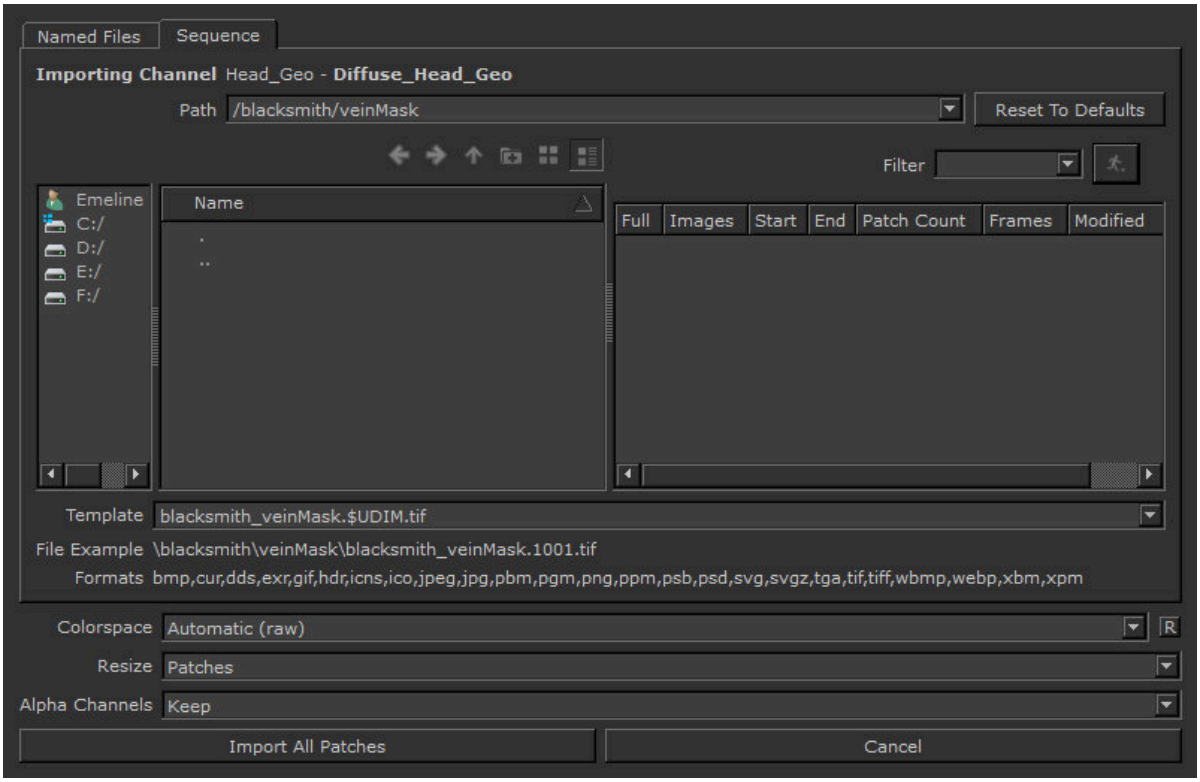
Note: When the image source resolution is non-square, or unsupported by Mari, the image is resized to the nearest power-of-two square size bigger than the original. You are informed of the resize change in the **Image Import Summary**, and is further detailed when **View Summary** is turned on.

Importing a Sequence of Files

If your patches have a consistent naming/numbering system, you can import them as a sequence.

1. Click the **Sequence** tab.

   The tab shows all the patches on the object.



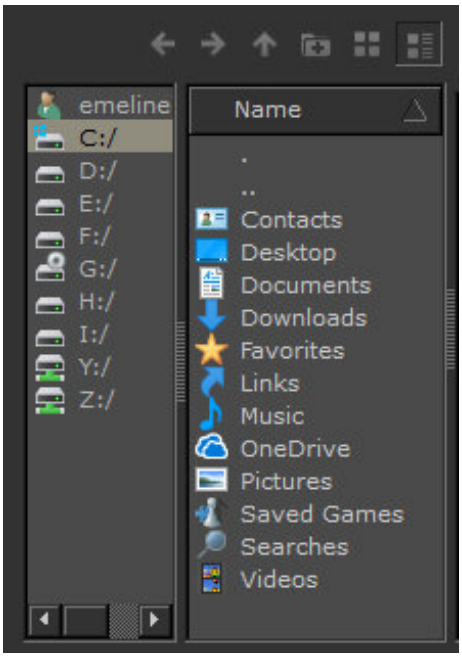
2. Enter the location of the files to import in the **Path** field.

Note: This isn't the name of the files themselves. It's the name of the directory in which the sequence of files exists.

Tip: Make sure you check the path and name carefully, especially if you have been swapping between assets - Mari defaults to the last place you've exported. Don't risk overwriting your textures!

You can either:

- enter the full path to the directory containing the files (or pick from the dropdown), or
- use the navigation controls to navigate to the directory:



When you've selected the path, the file area shows all the existing image sequences under that path.

3. If you are looking for a specific folder or file, you can use the **Filter** field to specify a name or file format to search for.

Note: The Walk button is toggled off by default. When enabled, Mari looks in all folders of the selected location (refer to the **Path** field in the **Import** dialog) and displays all folders and files in the right pane.

4. Either use the default filename template or set a **Template** for Mari to use when gathering the names of images in the selected sequence.

   You can either:

   - Edit this line manually. As you edit, the **FileExample** updates with the name of an example file in the sequence.

- Click on an existing set of images in the main list. The **Template** automatically populates with the corresponding values.

If you edit the line manually, you can use any of the following variables, which Mari replaces with the appropriate values when it imports the images:

- **$ENTITY** - the name of the object.
- **$CHANNEL** - the name of the channel.
- **$LAYER** - the name of the layer.
- **$NODE** - the name of the node.
- **$UDIM** - the UDIM patch value.
- **$FRAME** - the animation frame number.
- **$NUMBER** - the number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28").
- **$COUNT** - the total number of files in the channel.
- **$COLORSPACE** - the chosen colorspace for the selected images.
- **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").

Note: The steps above assume you are importing a sequence of patches with the UDIM number (1001, 1002, and so on.) in the file names. For more information on how the UDIM patch values are allocated, please see the section on Using UDIM Values.

Note: If you choose to import a layered **.psd** file, layers are imported directly into the selected channel. If attempting to import layers with the same name, Mari asks whether to update the layers or create new ones. You can now set resize details and import the textures.

5. Set the **Colorspace** for data export. By default, Mari selects the colorspace determined by the value set in **Default Import Colorspace** in the **Mari Preferences** dialog under the **Color** tab.

   See Mari Preferences Dialog for more information.
6. Set how Mari handles patch/image size mismatches in the **Resize** field. You can select either to resize the **Images** to match the patches, or to resize the **Patches** to match the images.
7. In the **Alpha Channels** field, choose to either **Keep** or **Remove** the alpha when importing channels. This can help decrease the resulting file size when exporting.
8. Click **Import All Patches** to import the textures or, if at least once patch is selected, this appears as **Import Selected Patches**.

   Mari imports the images and loads them into the layer stack of the selected channel.

Exporting Textures from Channels

Once you've finished working on your textures, to export the files (for example, for use in the next stage of the pipeline) select the channel(s) you want to export.

Video: To learn how to export textures out of Mari once you are done painting them, watch Exporting Textures.

Note: As the options below only export channels one dialog at a time, to export multiple channels at once, see the **Export Manager Dialog**.

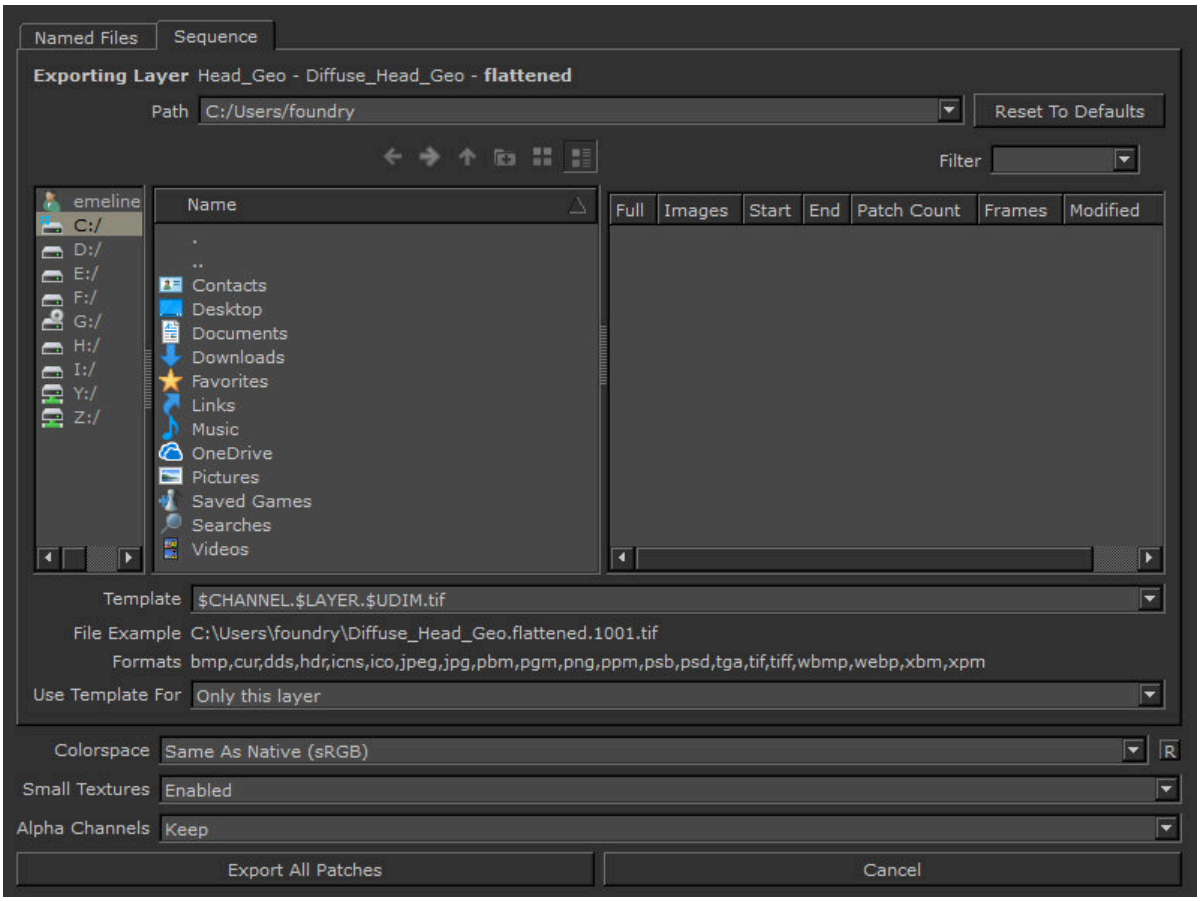
From the **Channels** menu, select **Export** or **Export Flattened**.

1. To export the channels without flattening them, do one of the following:
   - from the **Channels** menu, select **Export** > **Export Current Channel** to export the currently-selected channel, or
   - from the **Channels** menu, select **Export** > **Export All Channels** to export all channels in the **Channels** palette, or
   - from the **Channels** menu, select **Export** > **Export Everything** to export all channels for all objects in the project.
2. To flatten the channels before exporting them, do one of the following:
   - from the **Channels** menu, select **Export Flattened** > **Export Current Channel Flattened** to flatten and export the currently-selected channel, or
   - from the **Channels** menu, select **Export Flattened** > **Export All Channels Flattened** to flatten and export all channels in the **Channels** palette, or
   - from the **Channels** menu, select **Export Flattened** > **Export Everything Flattened** to flatten and export all channels for all objects in the project.

Tip: The **Export** and **Export Flattened** menu options are also available from the right-click dropdown menu on the main canvas and in the **Channels** palette.

Note: All of the methods for exporting textures shown above follow the same instructions once they reach the **Export Channels** dialog.

The **Export Channels** dialog displays.



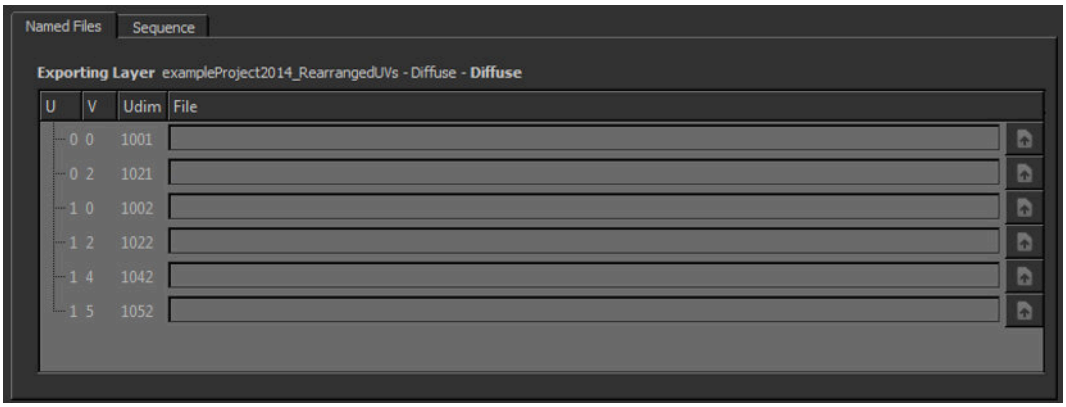
Exporting Individual Files

Exporting individual files lets you individually pick the patches to export, and where to export them.

Note: By default the **Named Files** tab doesn't display if you have more than 20 patches in your object.
To change this number, set the **Patch Cutoff** preference to a number greater than the number of patches in the file. Alternatively, you can use the **MARI_NAMEDFILEIMPORT_SEQUENCELIMIT** environment variable to override the preference setting. Please see the section on Environment Variables That Mari Recognizes for more information on environment variables.

1. Click the **Named Files** tab.

   The tab shows all the patches on the object.



2. For each patch, set the file to export it to.

   You can either:

   - type the file path and name directly, or
   - use the open button to open a navigation window to create a new file in a directory, or select an existing one to overwrite.

3. Select whether to enable **Small Textures**.

   With the small textures dropdown set to enabled, Mari exports any patches filled with a constant color as 8x8 pixel textures, no matter what resolution the patches are in Mari.

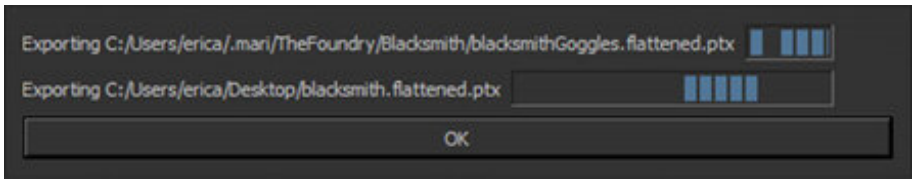
   - Enabling **Small Textures** saves disk space if you're exporting a channel that hasn't been fully painted.
   - Disabling **Small Textures** lets Mari always export patches at their full size (even if they haven't been painted).

4. Click **Export All Patches** to export the textures.

   Mari exports your textures to disk in the background.

5. To check the progress of the export click the icon at the bottom right of the main window.

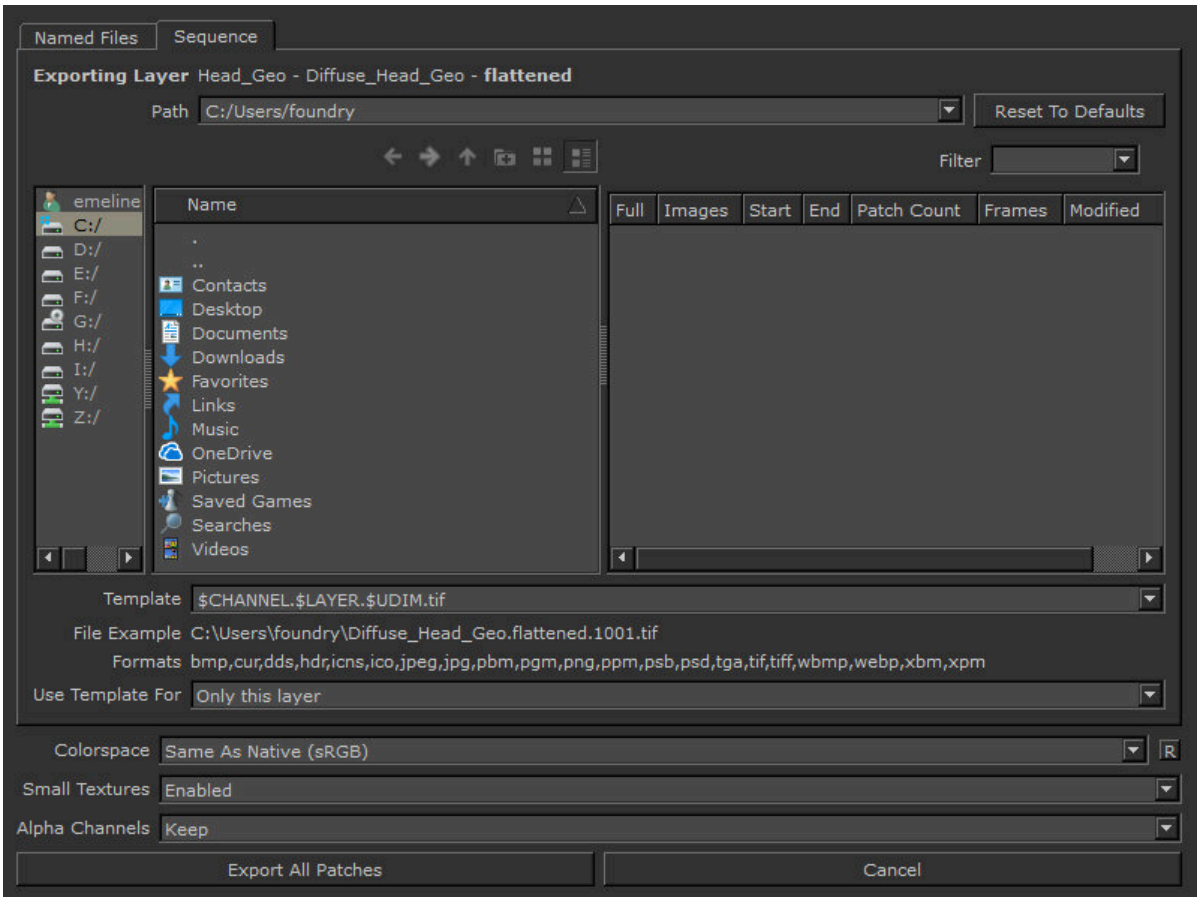
   Mari shows the export progress.



Exporting a Sequence of Files

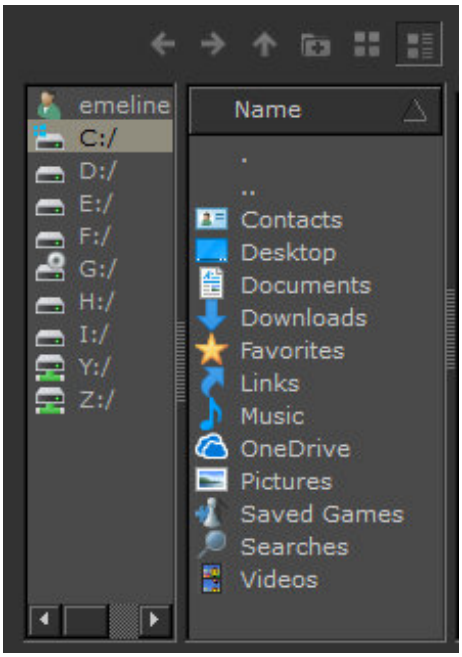
If your patches have a consistent naming/numbering system, you can import them as a sequence.

1. Click the **Sequence** tab.



2. Enter the location to export the files to in the **Path** field. You can either:
   - enter the full path to the directory (or pick from the dropdown), or
   - use the navigation controls to navigate to the directory.

> **Tip:** Make sure you check the path and name carefully, especially if you have been swapping between assets - Mari defaults to the last place you've exported. Don't risk overwriting your textures!



3. When you've selected the path, the file area shows all the existing image sequences under that path.
4. If you are looking for a specific folder or file, you can use the **Filter** field to specify a name or file format to search for.
5. Either use the default filename template or set a **Template** for Mari to use when generating the names of images in the selected sequence.

   You can either:

   - Edit this line manually. As you edit, the **File Example** updates with the name of an example file in the sequence.
   - Click on an existing set of images in the main list. The **Template** automatically populates with the corresponding values.

   If you edit the line manually, you can use any of the following variables, which Mari replaces with the appropriate values when it exports the images:

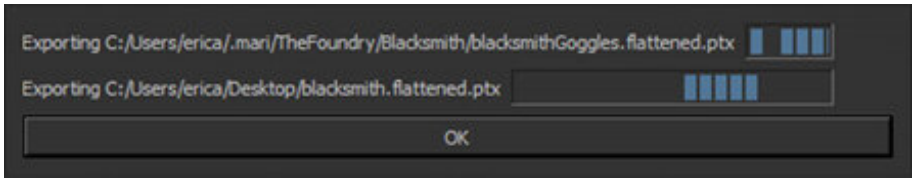
   - **$ENTITY** - the name of the object exported.
   - **$CHANNEL** - the name of the channel exported.
   - **$LAYER** - the name of the layer exported.
   - **$NODE** - the name of the node.
   - **$UDIM** - the UDIM patch value.
   - **$FRAME** - the animation frame number.
   - **$NUMBER** - the number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28").
   - **$COUNT** - the total number of files in the channel.
   - **$COLORSPACE** - the chosen colorspace for the selected images.

- **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").

6. Set the **Use Template For** field to determine if the template above is used for **Only this layer** (the selected layer in the channel's layer stack) or **Everything** (to include all layers in the stack).
7. Set the **Colorspace** in which you want to export the data. By default, Mari selects **Automatic**, which is determined by the file name, size, and type of data in the channel.
8. Decide whether **Small Textures** is enabled or disabled.
   - Enabling **Small Textures** saves disk space and exports any patches that you haven't painted yet, or have painted as a single color, as 8x8-pixel textures.
   - Disabling **Small Textures** lets Mari always export patches at their full size (even if they haven't been painted).
9. In the **Alpha Channels** field, choose to either **Keep** or **Remove** the alpha when importing channels. This can help decrease the resulting file size when exporting.
10. Click **Export All Patches** to export the textures.

    Mari exports your textures to disk in the background.
11. To check the progress of the export click the icon at the bottom right of the main window.

    Mari shows the export progress.



Note: The steps above assume you are exporting a sequence of patches with the udim number (1001, 1002, etc.) in the file names. For more information on how UDIM numbers are allocated to patches, please see Using UDIM Values.

Exporting Textures from Layers

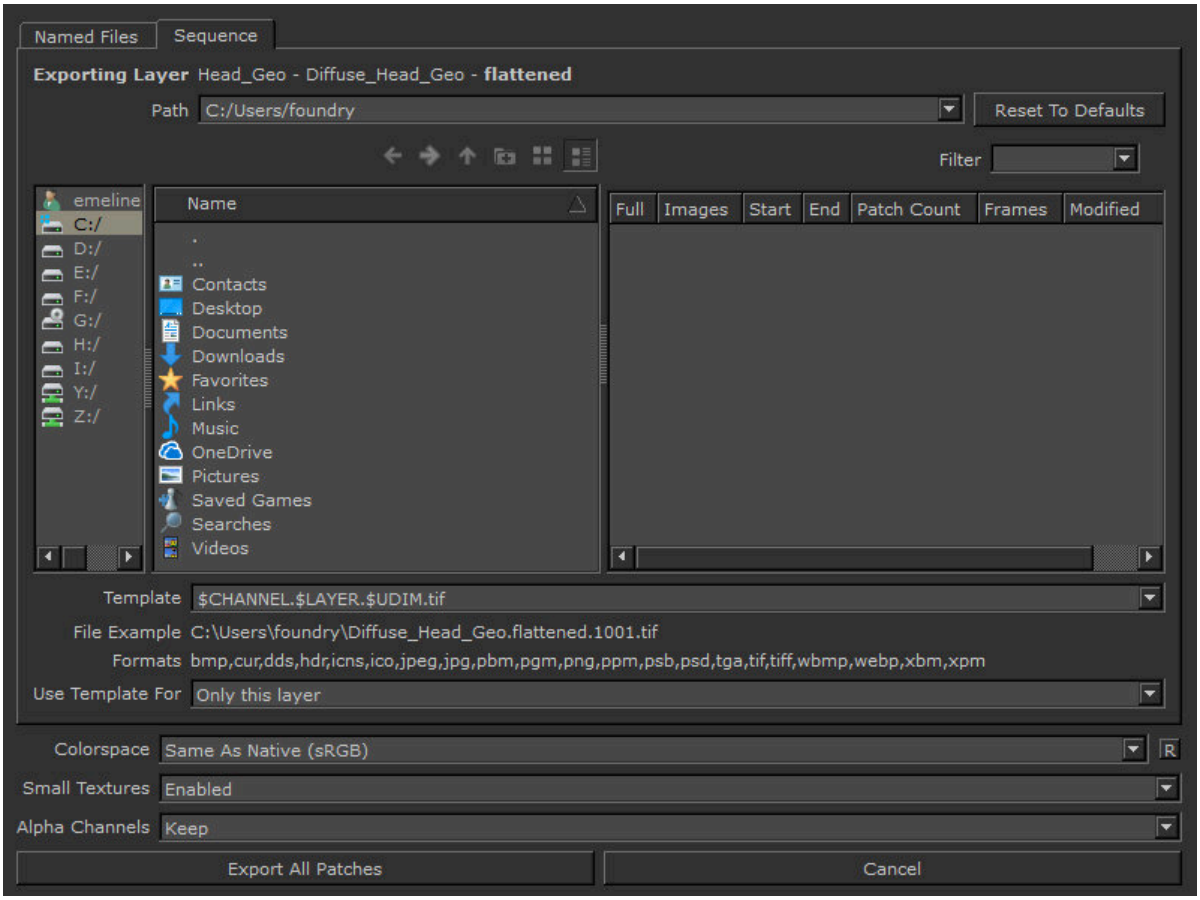
Instead of exporting work on whole channels, you may wish to export only a selection of layers by selecting them from the list in the **Layers** palette. As with channels, you can export multiple layers as they are in your layer stack or flattened into a single file.

From the **Layers** menu, select **Export** or **Export Flattened**.

1. To export layers without flattening them:
   - from the **Layers** menu, select **Export** > **Export Selected Layers** to export the currently selected layers, or
   - from the **Layers** menu, select **Export** > **Export All Layers** to export all layers in the layer stack.
2. To flatten the layers before exporting them:
   - from the **Layers** menu, select **Export Flattened** > **Export Selected Layers Flattened** to flatten and export the currently selected layers, or
   - from the **Layers** menu, select **Export Flattened** > **Export All Layers Flattened** to flatten and export all layers in the layer stack.

> **Tip:** The **Export Flattened** menu options are also available from the right-click dropdown menu on the main canvas and in the **Layers** palette.

The **Export** dialog displays.



Note: Since exporting all the layers in a layer stack would be the equivalent of exporting an entire channel, the **Export All Layers** and **Export All Layers Flattened** options in the **Layers** menu exports the same result as the **Export Current Channel** and **Export Current Channel Flattened** options in the **Channels** menu.

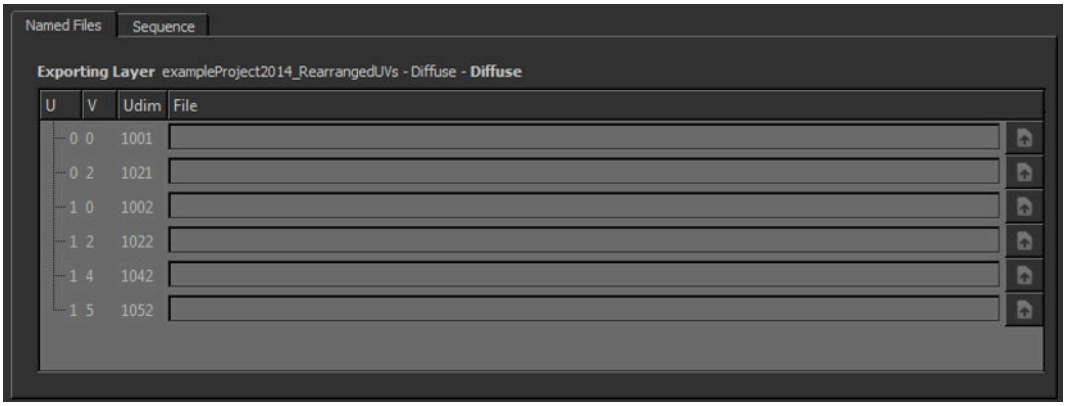
Exporting Individual Files

Exporting individual files lets you individually pick the patches to export, and where to export them.

Note: By default the **Named Files** tab doesn't display if you have more than 20 patches in your object.
To change this number, set the **Patch Cutoff** preference to a number greater than the number of patches in the file. Alternatively, you can use the **MARI_NAMEDFILEIMPORT_SEQUENCELIMIT** environment variable to override the preference setting. Please see the section on Environment Variables That Mari Recognizes for more information on environment variables.

1. Click the **Named Files** tab.

   The tab shows all the patches on the object.



2. For each patch, set the file to export it to.

   You can either:

   - type the file path and name directly, or
   - use the open button to open a navigation window to create a new file in a directory, or select an existing one to overwrite.

3. Select whether to enable **Small Textures**.

With the small textures dropdown set to enabled, Mari exports any patches filled with a constant color as 8x8 pixel textures, no matter what resolution the patches are in Mari.

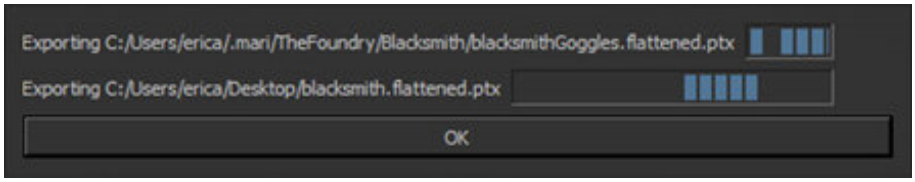
- Enabling **Small Textures** saves disk space if you're exporting a channel that hasn't been fully painted.
- Disabling **Small Textures** lets Mari always export patches at their full size (even if they haven't been painted).

4. Click **Export All Patches** to export the textures.

   Mari exports your textures to disk in the background.

5. To check the progress of the export click the icon at the bottom right of the main window.

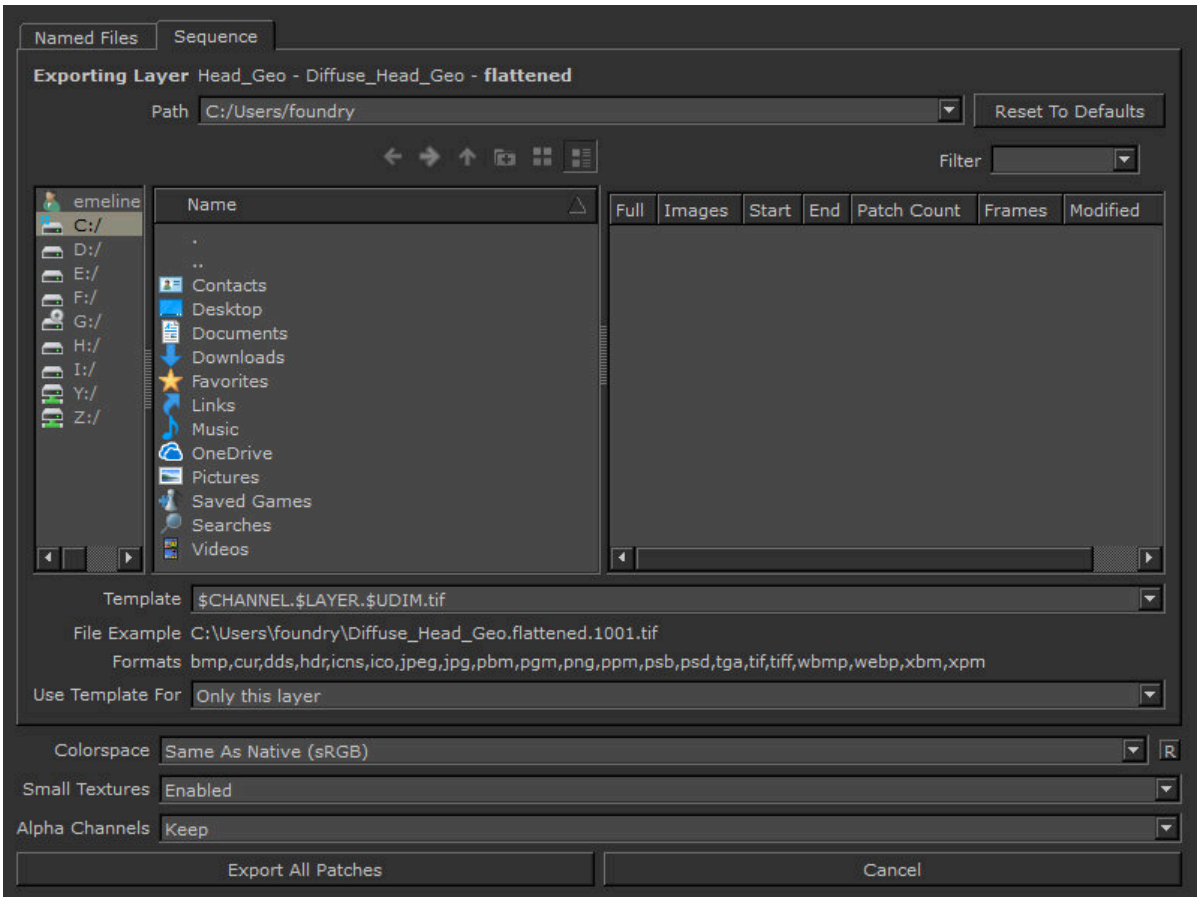
   Mari shows the export progress.



Exporting a Sequence of Files

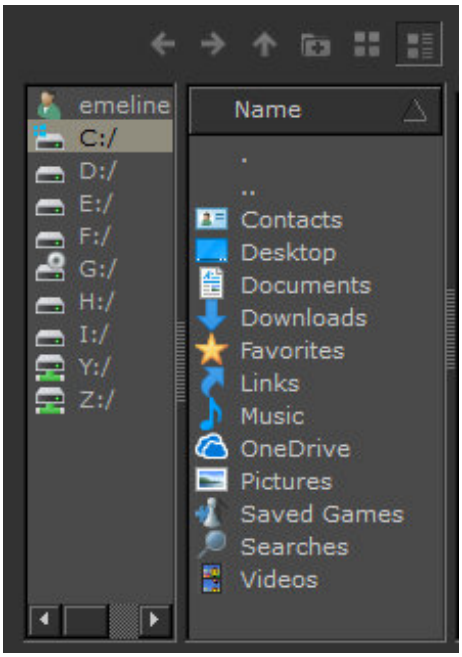
If your patches have a consistent naming/numbering system, you can import them as a sequence.

1. Click the **Sequence** tab.



2. Enter the location to export the files to in the **Path** field. You can either:
   - enter the full path to the directory (or pick from the dropdown), or
   - use the navigation controls to navigate to the directory.

Tip: Make sure you check the path and name carefully, especially if you have been swapping between assets - Mari defaults to the last place you've exported. Don't risk overwriting your textures!



3. When you've selected the path, the file area shows all the existing image sequences under that path.
4. If you are looking for a specific folder or file, you can use the **Filter** field to specify a name or file format to search for.
5. Either use the default filename template or set a **Template** for Mari to use when generating the names of images in the selected sequence.

   You can either:

   - Edit this line manually. As you edit, the **File Example** updates with the name of an example file in the sequence.
   - Click on an existing set of images in the main list. The **Template** automatically populates with the corresponding values.

   If you edit the line manually, you can use any of the following variables, which Mari replaces with the appropriate values when it exports the images:

   - **$ENTITY** - the name of the object exported.
   - **$CHANNEL** - the name of the channel exported.
   - **$LAYER** - the name of the layer exported.
   - **$NODE** - the name of the node.
   - **$UDIM** - the UDIM patch value.
   - **$FRAME** - the animation frame number.
   - **$NUMBER** - the number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28").
   - **$COUNT** - the total number of files in the channel.
   - **$COLORSPACE** - the chosen colorspace for the selected images.

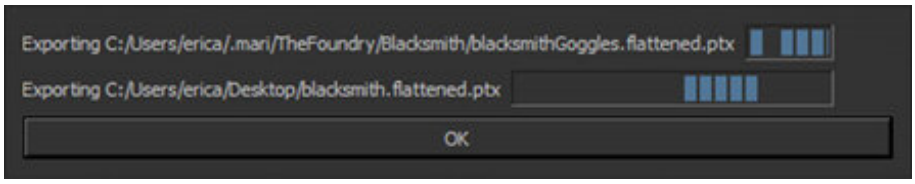
- **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").

> **Note:** If you want to create a layered **.psd** file from your layer stack for an individual channel in your scene, set the template to **$CHANNEL.$UDIM.psd**.

6. Set the **Use Template For** field to determine if the template above is used for **Only this layer** (the selected layer in the channel's layer stack) or **Everything** (to include all layers in the stack).
7. Set the **Colorspace** in which you want to export the data. By default, Mari selects **Automatic**, which is determined by the file name, size, and type of data in the channel.
8. Decide whether **Small Textures** is enabled or disabled.
   - Enabling **Small Textures** saves disk space and exports any patches that you haven't painted yet, or have painted as a single color, as 8x8-pixel textures.
   - Disabling **Small Textures** lets Mari always export patches at their full size (even if they haven't been painted).
9. In the **Alpha Channels** field, choose to either **Keep** or **Remove** the alpha when importing channels. This can help decrease the resulting file size when exporting.
10. Click **Export All Patches** to export the textures.

    Mari exports your textures to disk in the background.
11. To check the progress of the export click the icon at the bottom right of the main window.

    Mari shows the export progress.



> **Note:** The steps above assume you are exporting a sequence of patches with the udim number (1001, 1002, etc.) in the file names. For more information on how UDIM numbers are allocated to patches, please see Using UDIM Values.

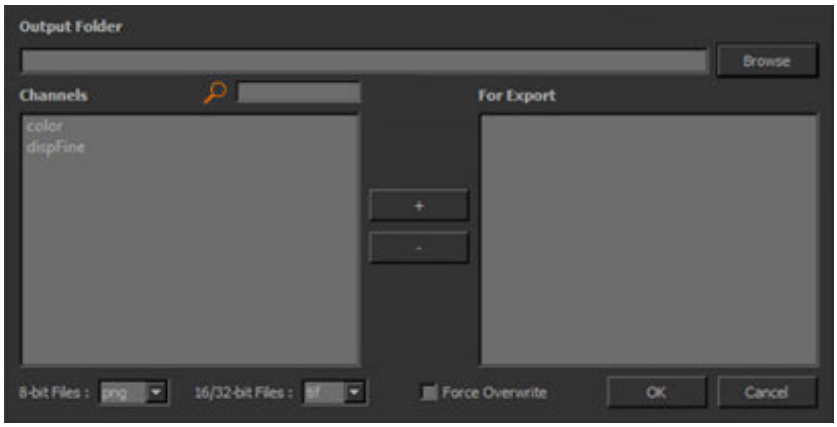
Exporting to Maya

Mari can also export data for use in Maya. It does this by exporting textures and creating a custom **.ma** file that references them. All exported files contain UDIM offsets so that Maya can import and read the UV patches correctly.

To create the **.ma** file for use in Maya follow the steps below:

1. Navigate to *Menubar* | **Python** > **Examples** > **Export for Maya**.

   The **Mari To Maya Export** dialog displays.



2. In the **Output Folder** pane, either enter the filepath or click **Browse** to find the location to which you would like the files to be exported.
3. From the list in the left pane, select which channels you want to export.
4. Once you have selected the channel and the filepath, click **+** (plus).

   The selected channel and filepath are added to the **For Export** pane on the right.
5. If you have added a channel by mistake, or want to edit the filepath, simply select the channel from the **For Export** pane and click **-** (minus)**.**
6. At the bottom of the dialog, the **8-bit Files** and **16/32-bit Files** dropdown menus list all of the file formats in which you can export your channels for use in Maya. Before you select **OK**, make sure you set your desired file format.

   If you have already exported these files to the same file location, you can opt to select the **Force Overwrite** checkbox to export over and replace them.
7. Select **OK**.

   Your channel is exported to the selected file location in your choice of file format. The textures, regardless of whether they are diffuse or displacement, are exported as UV patches.

Using Your Textures in Maya

Once the **.ma** file has been created, you can load your textures into Maya. Follow the simple steps below to set up your textures:

1. In your Maya scene, import the **.ma** file that was created after export from Mari.

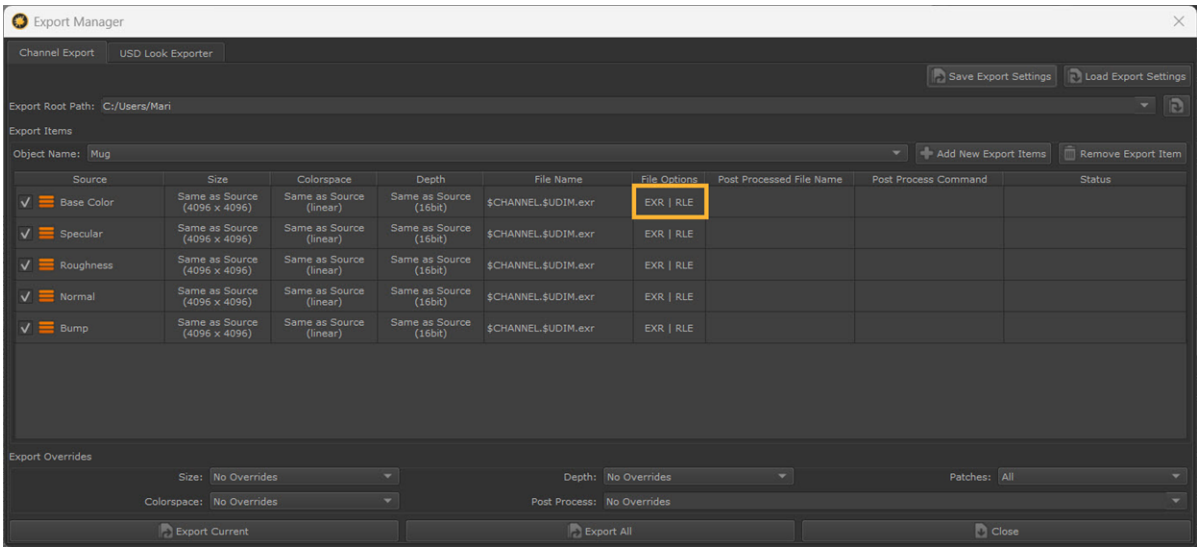
   This can be found under the directory you saved the exported file to.
2. Navigate to your Hypershade. You should now see that for each channel you exported out of Mari, a Lambert shader was created containing the channel.
3. You can hook up the layered texture element to different parts of your other, independent shader to achieve the desired look.

Exporting EXR Files for Render

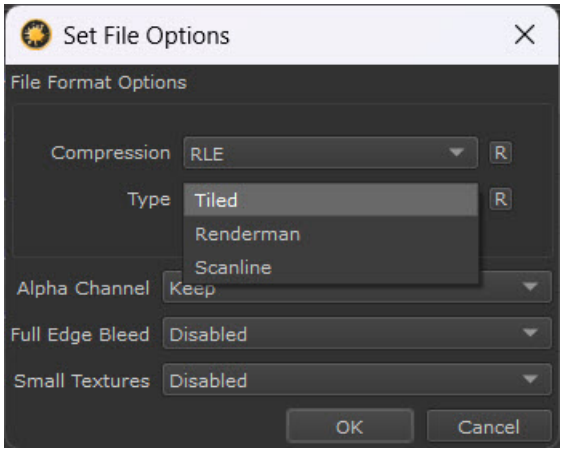
You can use the Export Manager Dialog to set file options when exporting EXR files, allowing your images to work better with renderers.

How to access these options:

1. From the top menu bar, select **Channels** > **Export Manager**
2. This opens the Export Manager Dialog. Select a location for your export in the field **Export Root Path**.
3. If you don't already have you export items loaded, select **Add New Export Items** and pick which channels or bake points you'd like to export from.
4. Double click on an EXR file.



5. The **Set File Options** Dialog will display. Choose a compression type from the **Type** dropdown menu.



- **Tiled**:

  Default. EXR files will be saved in the tiled format without mipmaps. The tile size is 256x256.

- **Renderman**:

  EXR files will be saved in the Renderman compliant format. The tile size is 32x32 and mipmaps will be generated.

- **Scanline**:

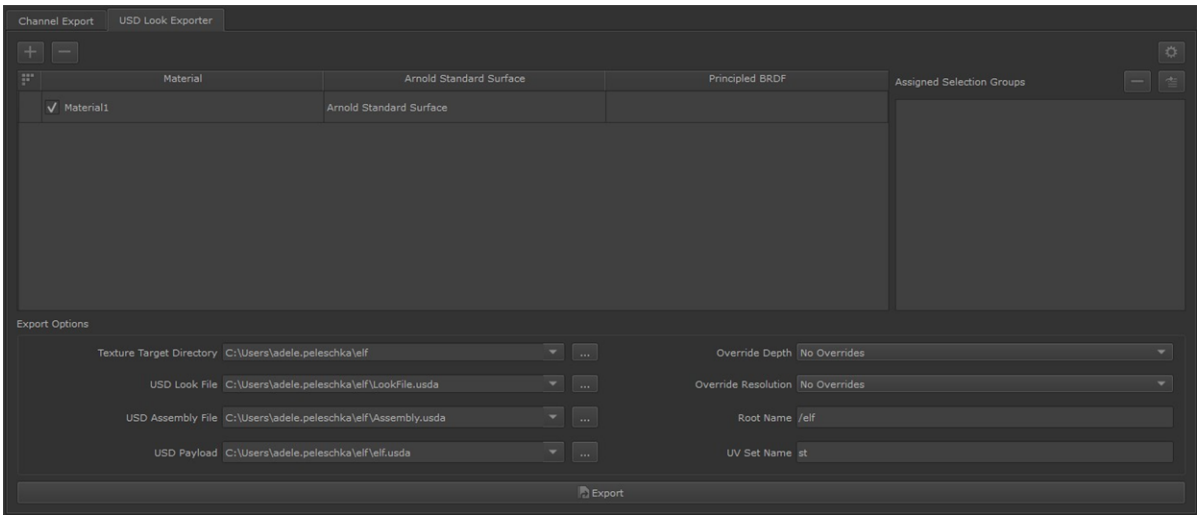
  EXR files will be saved in the scanline format. Currently only available via PythonAPI with an opt-in flag.

6. Once you have chosen an option, select **OK**.
7. Configure other export settings and overrides if needed, then select **Export Current** or **Export All** to export your EXR(s) to the chosen location.

Exporting a Flattened USD Preview of Your Asset's Look

The USD Export feature enables you to generate flattened preview looks from a USD Preview Surface, Arnold Standard Surface, PxrSurface or Principled BRDF Shader in Mari, so that they may be used in other programs that support the USD format, such as Katana.

Note: The USD Export tab has changed between Mari 5.0 and Mari 6.0, and the legacy USD Export has been hidden for the updated USD Look Exporter. The legacy tab can still be accessed by using the environment variable `MARI_USD_LEGACY_EXPORT_TAB`. For more information on environment variables in Mari, see Environment Variables That Mari Recognizes.



The **USD Look Export** tab as it appears in the **Export Manager**.

USD Export Types

The USD export options for Mari are as follows:

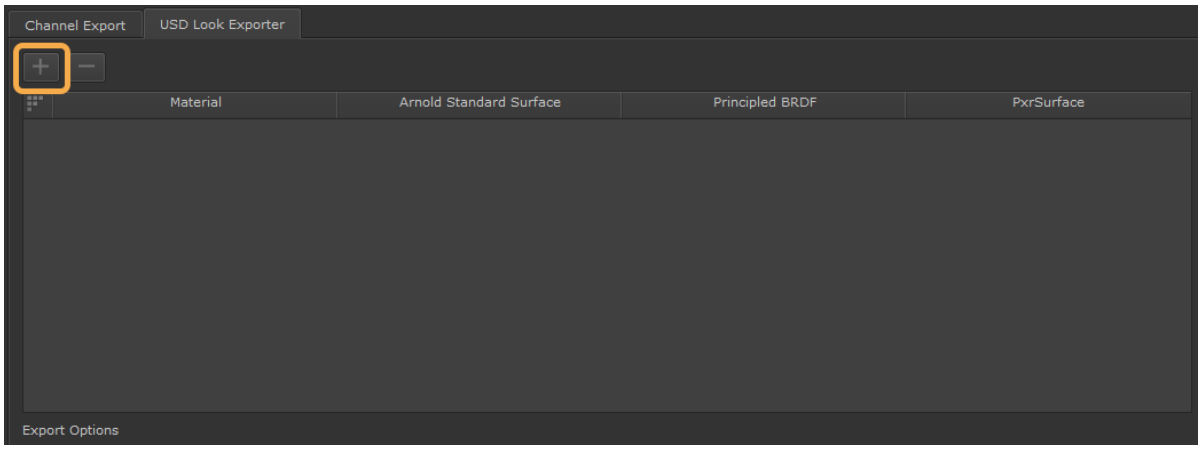
- **USD Look File** - This export contains the 'look' of an asset, for example any materials, assignments or shading networks associated with your object will be exported.
- **USD Assembly File** - This export bundles the 'look' of an asset with the geometry it is associated with.

Note: To export USD Assembly files, you must specify an associated USD payload file.

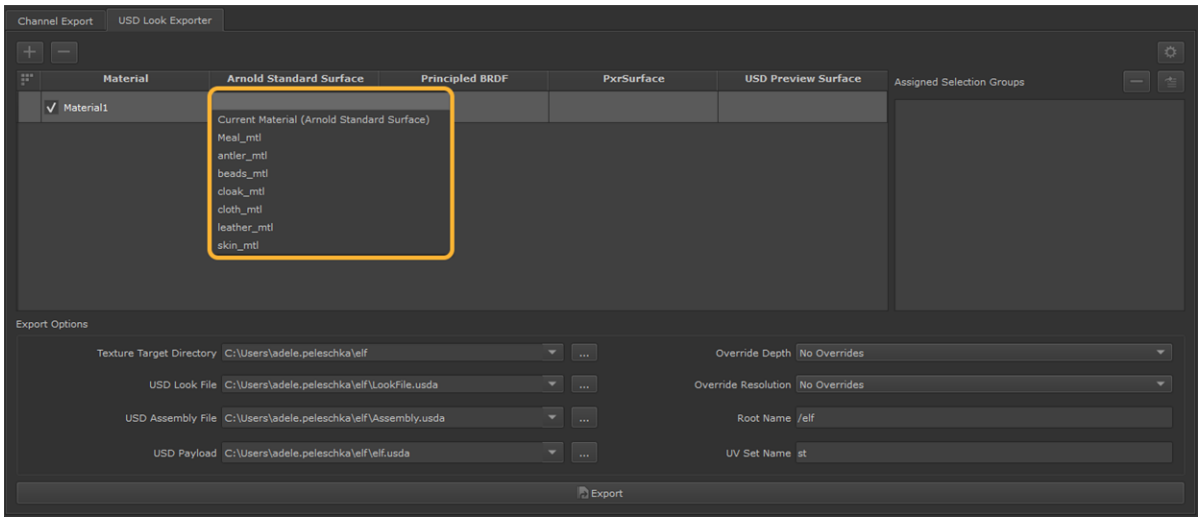
Exporting USD Preview Looks

To export a USD Preview Surface in Mari:

1. Open the **Export Manager** through **Channels** > **Export Manager** or the icon in the project toolbar.
2. Open the **USD Look Export** tab.
3. In the USD Look Exporter, click the plus button at the top of the window.

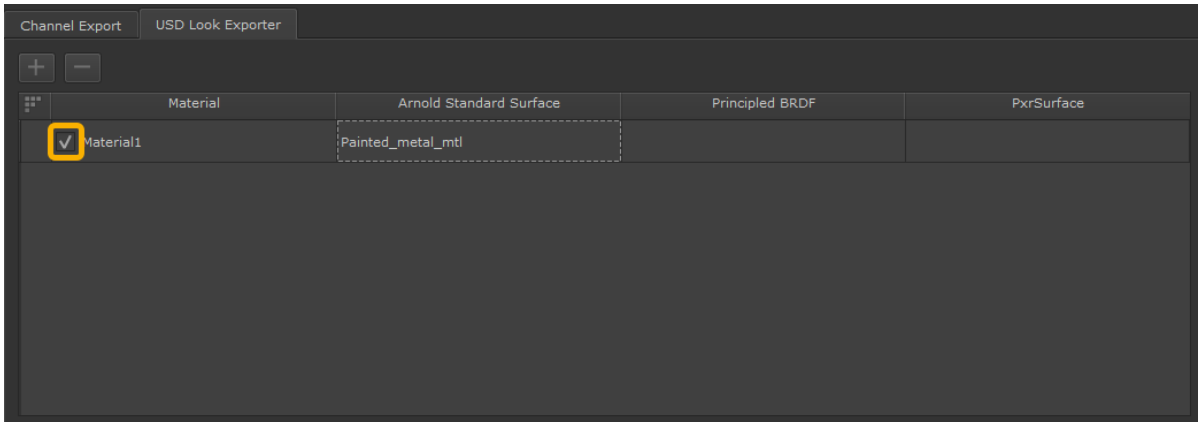


4. Double click the box under the shader that corresponds to the shader type that you are exporting, and select your shader from the drop down.



Note: The shader types available for export in the USD Look Exporter are dependent on the shaders present in your node graph. For example, if your node graph only contains an Arnold Standard Surface shader, only that shader is present in the USD Look Exporter. If you have Arnold Standard Shader, USD Preview Surface and Principled BRDF in your node graph, all three are available in the USD Look Exporter.

5. Esure that the checkbox next to your shader is checked.



6. Right click your material and select **Edit Shader Inputs**. In the window that appears, specify the export **Size**, **Color Space**, **Depth**, and **Format** per channel if needed.

7. Specify the target destination for your export and any variables for the file names of your textures. See Exporting Textures from Channels for more details.
   - If the export is a flattened preview, the target destination can be specified in the **Texture File Name** field.

- To generate a USD 'look' file of your channels, specify a target destination in the **USD Look File** field.
- To export an assembly file containing any geometry used with the shaders in your scene, specify a target destination in the **USD Assembly File** field.

8. Once all the relevant target destinations have been set, click **Export to USD**.

> **Note:** When exporting USD files, a root name must be specified in the **Root Name** text field. Any root name specified to this parameter must include the **/** (forward slash) prefix.

Session Scripts

Session scripts are a light-weight archive of a project. All the data you choose to record is saved in the **.msf** (Mari Session File) format, with the necessary files (models, images, projectors, and so on) to rebuild your project.

The purpose of session scripts is to enable you to share your session (project) with other users and to allow them to import the data into another session, such as objects, Node Graphs, channels, patches, lights, projectors, subdivision, Ambient Occlusion, and so on. An example of where this might be useful would be where two artists are working on a single asset broken into separate models, such as a table and chairs. If artist A is painting the chair models, and artist B is painting the table model, and artist A is asked to handover their work in its current form to artist B, it might be faster and easier for artist B to combine the sessions rather than work on two separate projects. By allowing artist A to export their current session for artist B to import into their own session, both projects can be worked on simultaneously.

Additional documentation is located under **Python** > **Documentation**.

Session Scripts Limitation

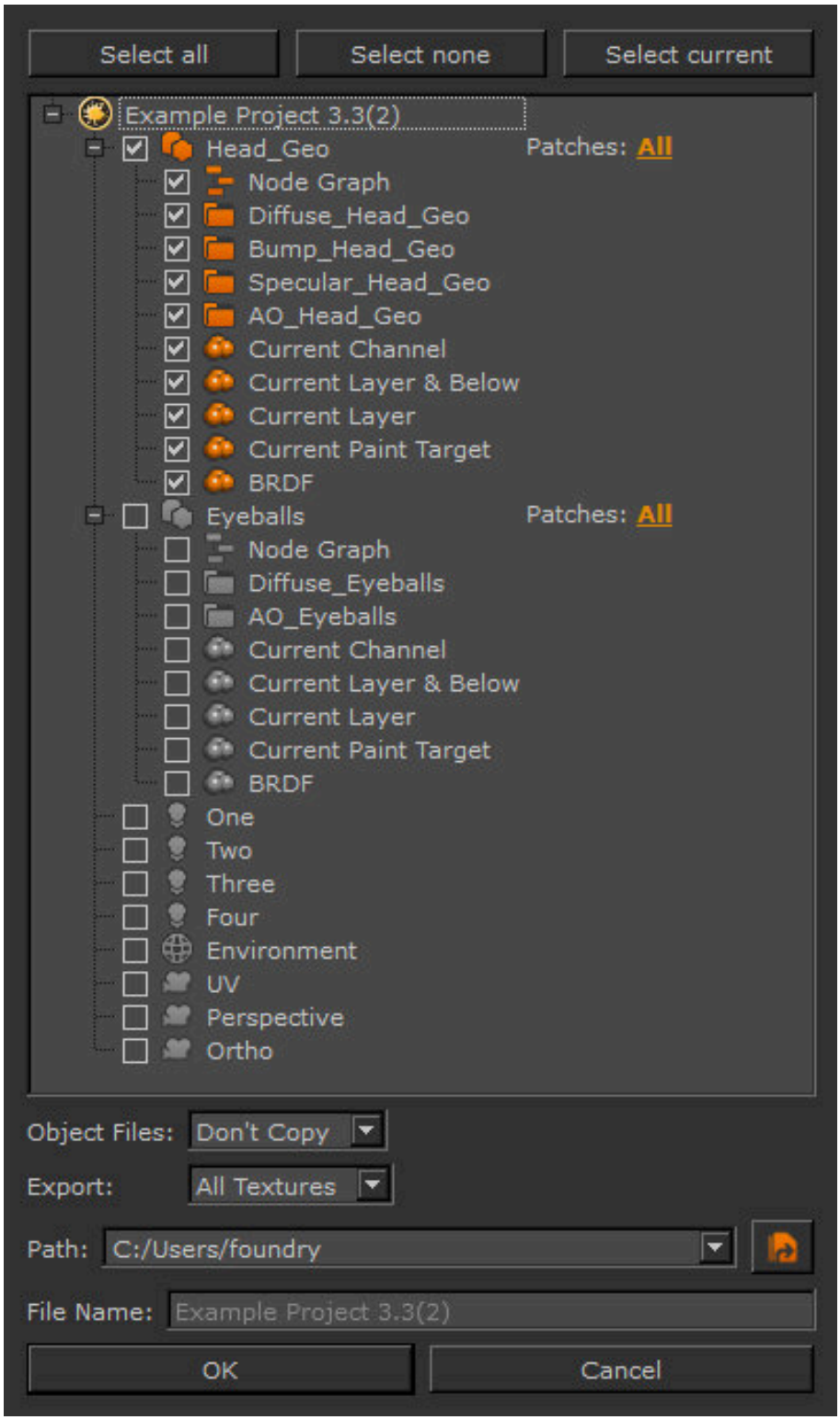
While session scripts can be incredibly useful when working between multiple projects, there is one limitation that you should be aware of before continuing. If you import a session with transformed geometry (rotated, scaled, and panned) located at a leaf of the scene hierarchy, then the transformation to the parent and ancestor scene nodes are not retained.

Exporting a Session

To export a session, you first need to have a project open. Then follow the steps below:

1. Navigate to **File** > **Session** > **Export Session**.

   The **Export Mari Data** dialog opens.



2. Select the objects, Node Graphs, channels, patches, lights, projectors, subdivision, Ambient Occlusion, and so on that you want to export by checking the box next to the item in the Mari entity tree.

Note: Mari's shaders, channels, and layers all work on top of the underlying Node Graph system. Therefore, Session Scripts from Mari 3.2 onwards saves the corresponding nodes to selected channels in the Node Graph format regardless of whether the Node Graph checkbox is enabled. If you enable the Node Graph checkbox, the entire Node Graph is saved. When it's not enabled, only the nodes corresponding to the selected channels are saved.

3. Choose whether or not the **Object Files** are copied to the export directory specified in the **Path** field. If they aren't copied, no additional files are created, other than the project file, at the export directory location.

Tip: Object files can be **.obj**, **.fbx**, or **.abc** files.

4. Choose whether or not to **Export** all textures (image data for layers) for the selected items, or to only create a template. The template can be used to set up projects for artists to work from, so channels, layers, shader, and so on have the desired settings from the start.
5. Specify the directory to which you want to export the session data in the **Path** field.
6. Specify the name to give the exported **.msf** file and project in the **File Name** field.
7. Once you have set your export options, click **OK**.

   A progress bar displays the status of the export. Once it reaches 100%, another dialog reminds you not to close the project until all the background jobs have finished. These background jobs can be seen by clicking on the running man icon in the lower-right hand corner, and comprise the export of images for layers and the Image Manager.

   Once the export is complete you can close the project and open a new one.

Warning: Attempting to export a session without geometry selected is only successful when merging the session into another session with the same geometry. Otherwise, you must select the geometry before proceeding.

Importing a Session

To import a session, you have two options for handling an exported session: you can either create a new project from the session data or you can import it into an open project. To import it into an open project, follow the steps below:

1. Navigate to **File** > **Session** > **Import Session**.

The **Import Session** dialog opens.

2. Specify the directory where Mari is looking for the **.msf** file in the **Path** field.

   This is the location where the file was exported to in the Exporting a Session instructions, and the file has the same name as the project that it was exported from.

3. Choose whether to import **All Data** from the **.msf** file, or only import the data **Onto Selected Object**.

   If you import all data, this imports data, including geometry, as separate objects into the current project. If you import data onto the selected object, this imports the shader and channel information in the **.msf** file onto the currently-selected object.

> **Note:** The **Import** option only is available when a project is open.

4. Once you have set your import options, click **OK**.

   A progress bar displays the status of the import.

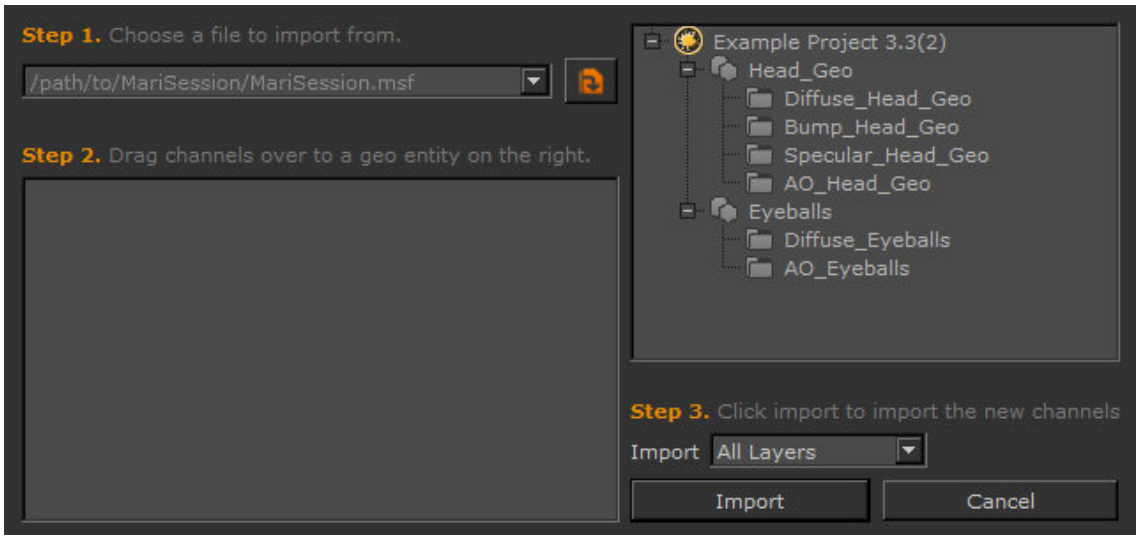
Once the import has finished, Mari displays the entities that were imported (objects, light, projectors, subdivision, and so on) and displays the data you imported.

Importing Channels

To import a channel, you first need to have a project open. Then follow the steps below:

1. Navigate to **File** > **Session** > **Import Channels**.

   The **Import Channels** dialog opens.



2. Specify or browse for the **.msf** file.

   Selecting an appropriate file populates the Geo Entity tree on the left-hand side of the dialog with the data in the **.msf** file.
3. The tree on the right-hand side of the dialog is your current, open project. Drag and drop channels from the tree on the left to any object in the tree on the right.
4. Once you have finished adding the channels you want to create, from the **Import** dropdown, select whether you want to export **All Layers** or the **Selected Layers**.
5. Click **Import**.

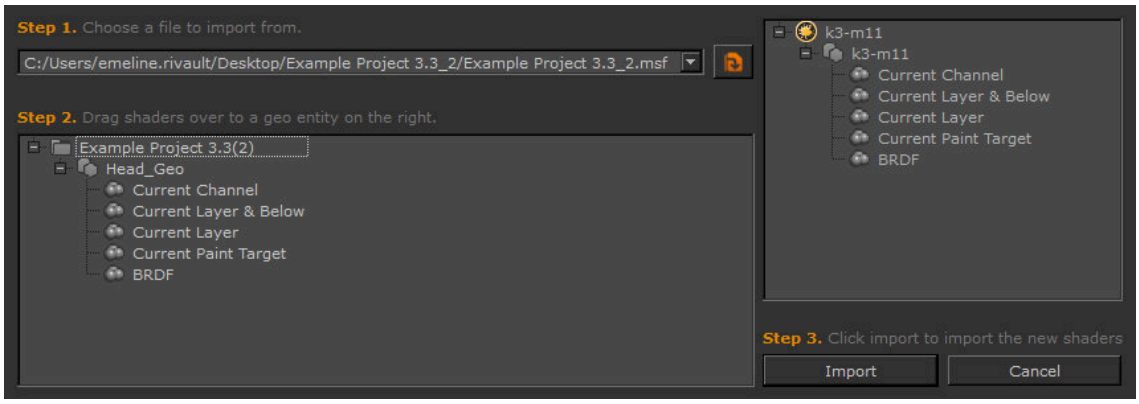
   Once the import process has finished, you can continue to add more channels at any time.

Importing Shaders

To import a shader, you first need to have a project open. Then follow the steps below:

1. Navigate to **File** > **Session** > **Import Shaders**.

   The **Import Shaders** dialog opens.



2. Specify or browse for the **.msf** file.

   Selecting an appropriate file populates the Geo Entity tree on the left-hand side of the dialog with the data in the **.msf** file.
3. The tree on the right-hand side of the dialog is your current, open project. Drag and drop shaders from the tree on the left to any object in the tree on the right.
4. Once you have finished adding the shaders you want to create, click **Import**.

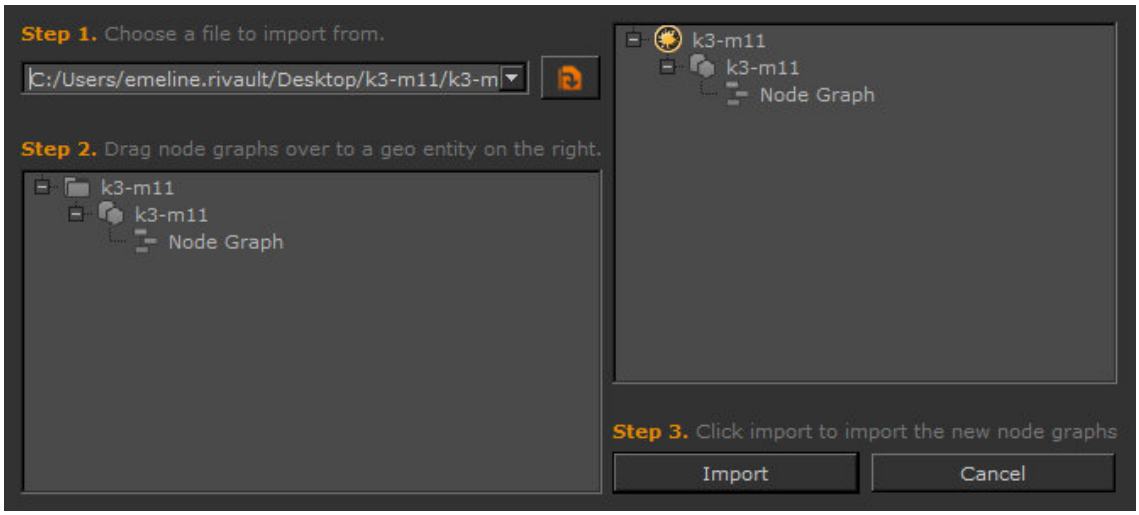
Once the import process has finished, you can continue to add more shaders at any time.

Importing Node Graphs

To import Node Graphs, you first need to have a project open. Then follow the steps below:

1. Navigate to **File** > **Session** > **Import Node Graphs**.

   The **Import Node Graphs** dialog opens.



2. Specify or browse for the **.msf** file.

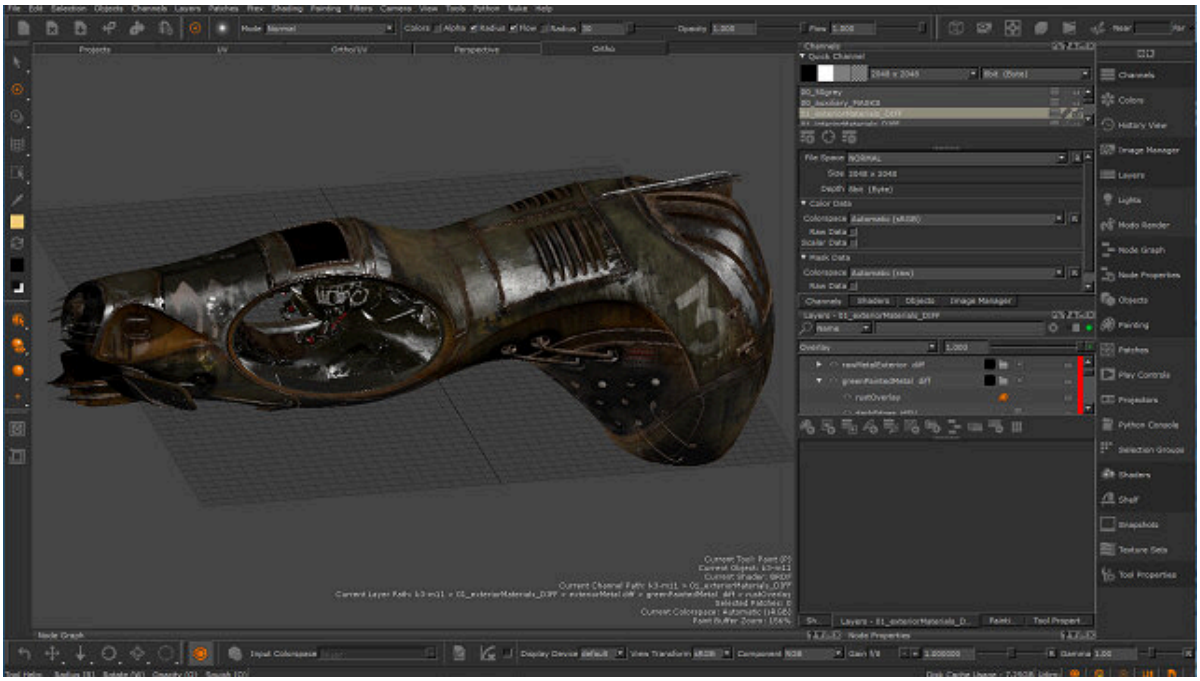
Selecting an appropriate file populates the Geo Entity tree on the left-hand side of the dialog with the data in the **.msf** file.

3. The tree on the right-hand side of the dialog is your current, open project. Drag and drop Node Graphs from the tree on the left to any object in the tree on the right.
4. Once you have finished adding the Node Graphs you want to create, click **Import**.

   Once the import process has finished, you can continue to add more Node Graphs at any time.

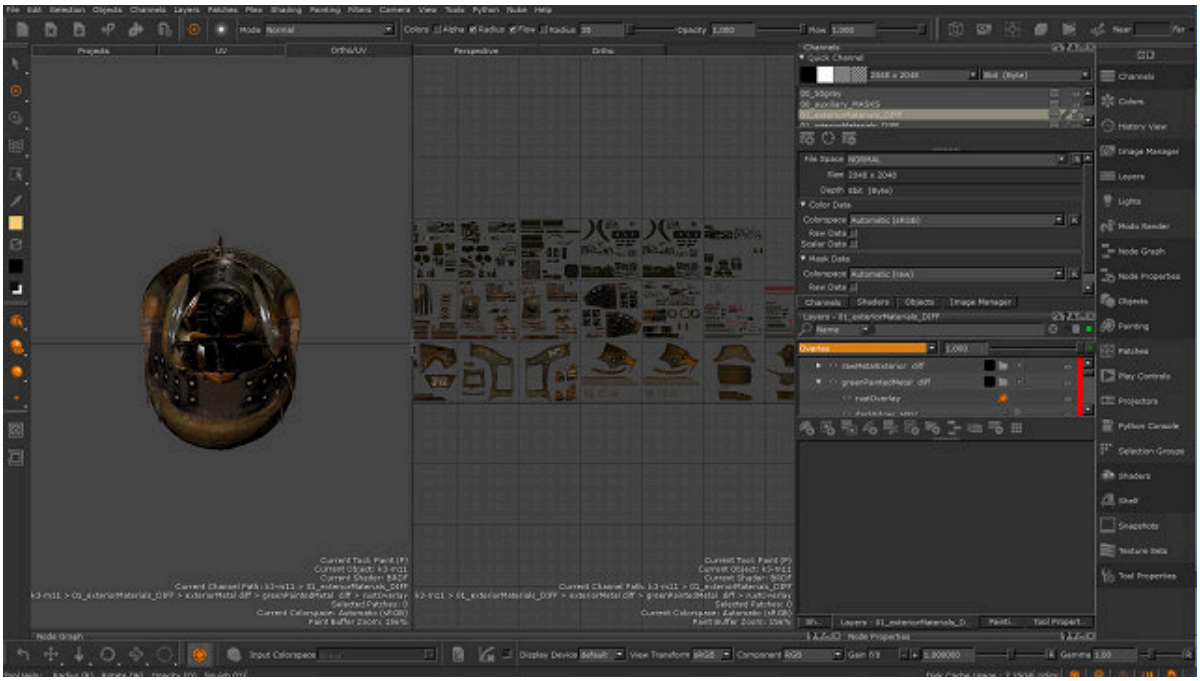
Mari Quick Start Guides

Follow these Mari written tutorials to get a good understanding of how Mari works. These tutorials are basic, but they cover the main areas you need to know to make the best use out of Mari.



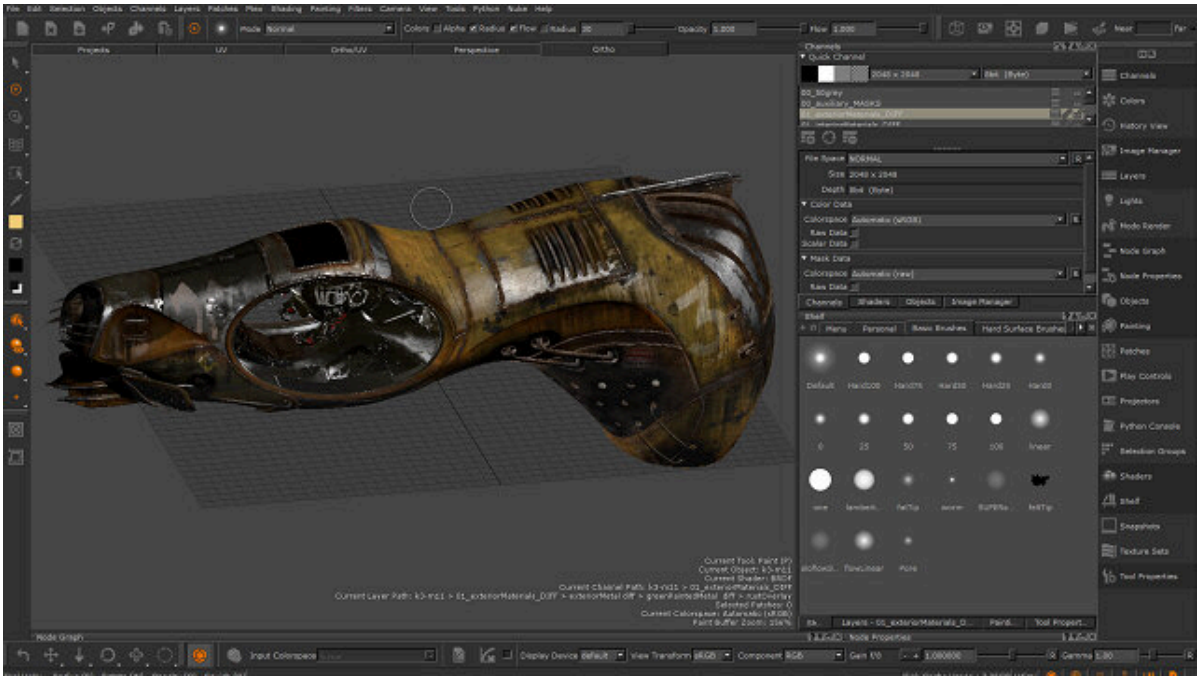
Set up a Project

Follow this step by step tutorial to create a project.



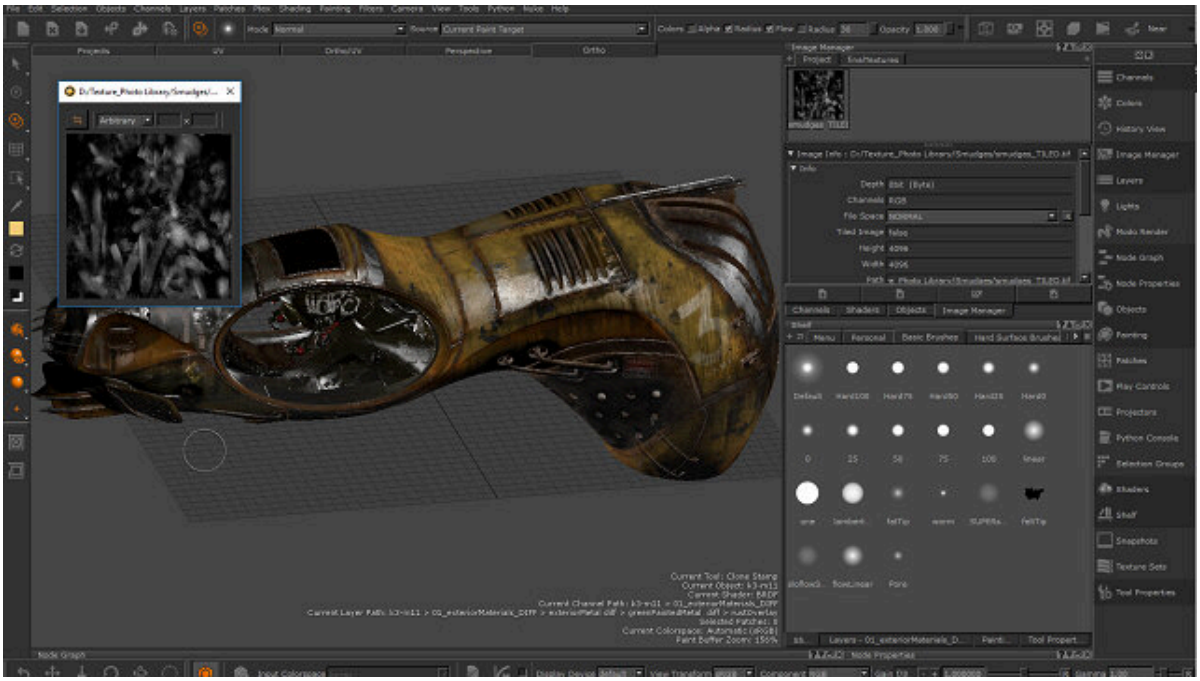
Set the View

Follow this step by step tutorial to set the view and lighting on your geometry.



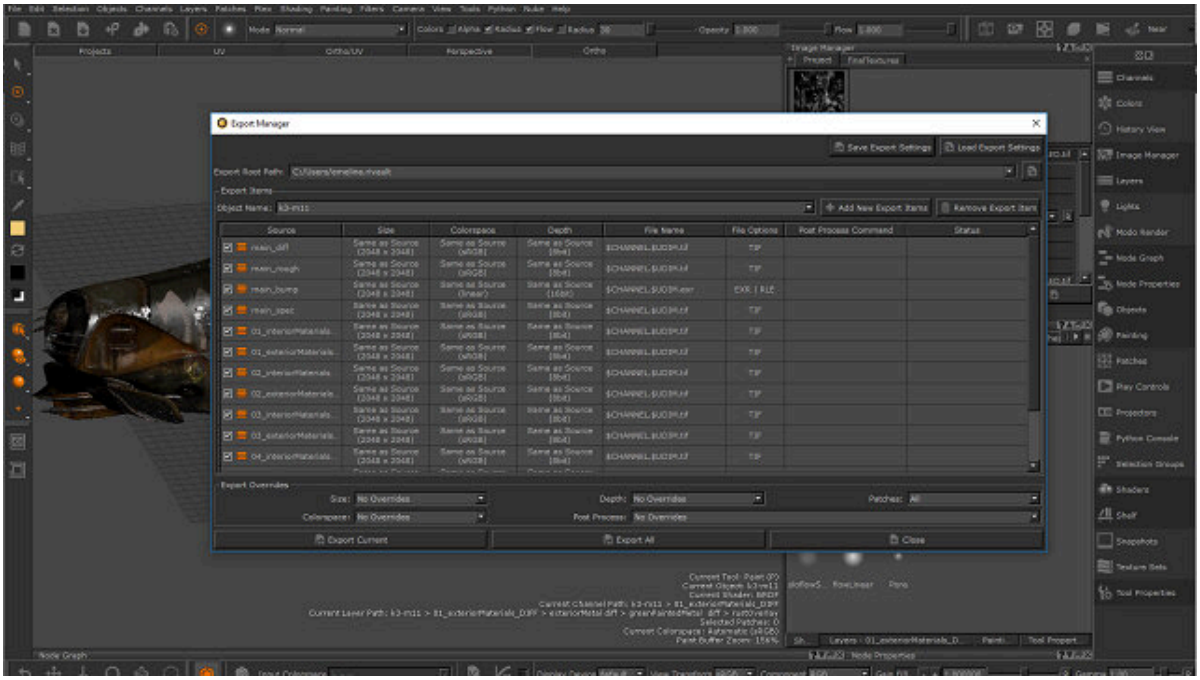
Paint

Follow these basic steps to paint on a model in Mari.



Paint Through and Clone Stamp

Follow this step by step tutorial to use images to create textures in Mari.



Export and Import

Learn how to export channels or layers and re-import them into your project.

Setting up a Project

This lesson teaches you what Mari projects are, and how to create them.

Resources Needed

To complete this lesson, you need access to a computer that can run Mari, and the following sample file:

• **blacksmith_body.obj**

How Long Should It Take?

Plan on spending about 20 minutes to complete this lesson.

About Mari Projects

A Mari project holds your work on geometries and associated textures. Once you've created a Mari project, you can work on it, save and close - and then re-open it to continue working on the same textures later.

Projects also hold other items, such as projectors, shaders, and so on. Mari saves some settings at the project level. For instance, each project has:

• a project **shelf**, to hold brushes, colors, and images for you to use specifically in that project, and

• contents of the various palettes - **Image Manager**, and so on.

When you first create a project, you can specify its parameters, including:

• the details (such as resolution and color depth) of the initial channel in the project.

• the range of animation frames to import.

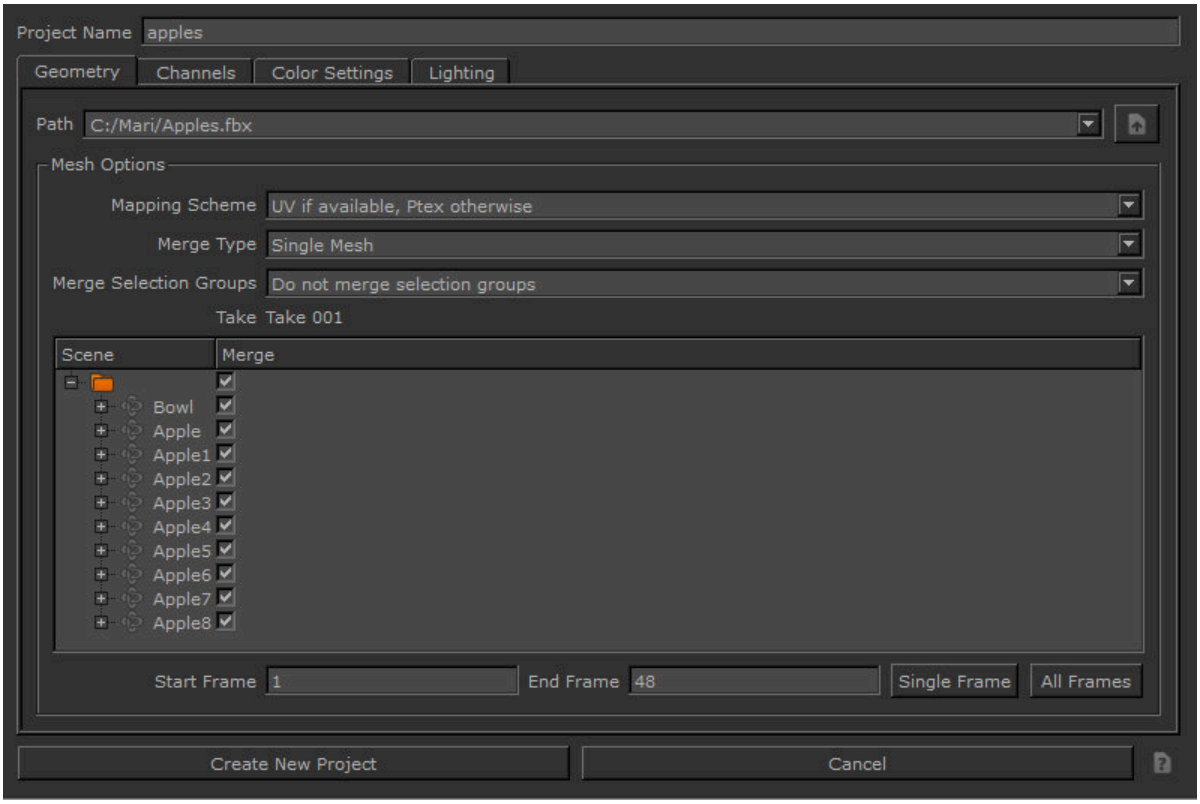
Opening a New Project

1. Start Mari.

   The Mari workspace displays, showing the **Projects** tab. This holds all the projects you have been working on.

2. As with most commands in Mari, you can select the **New Project** option a few different ways, pick any one of the options below:
   - from the **File** menu, select **New**,
   - click on the toolbar,
   - click the **New** button,
   - right-click in the **Projects** tab and select **New** from the dropdown menu, or
   - press the **Ctrl**/**Cmd**+**N** shortcut key.

   The **New Project** dialog box displays.



The **New Project** dialog displays the **Name** for your project and has four tabs:

- **Geometry** - the model you'll be painting on.
- **Channels** - parameters for the channels and textures you'll be creating.
- **Color Settings** - parameters for the colorspace of your project and for the color pickers, swatches, and images.
- **Lighting** - the lighting options for your model and the environment light.

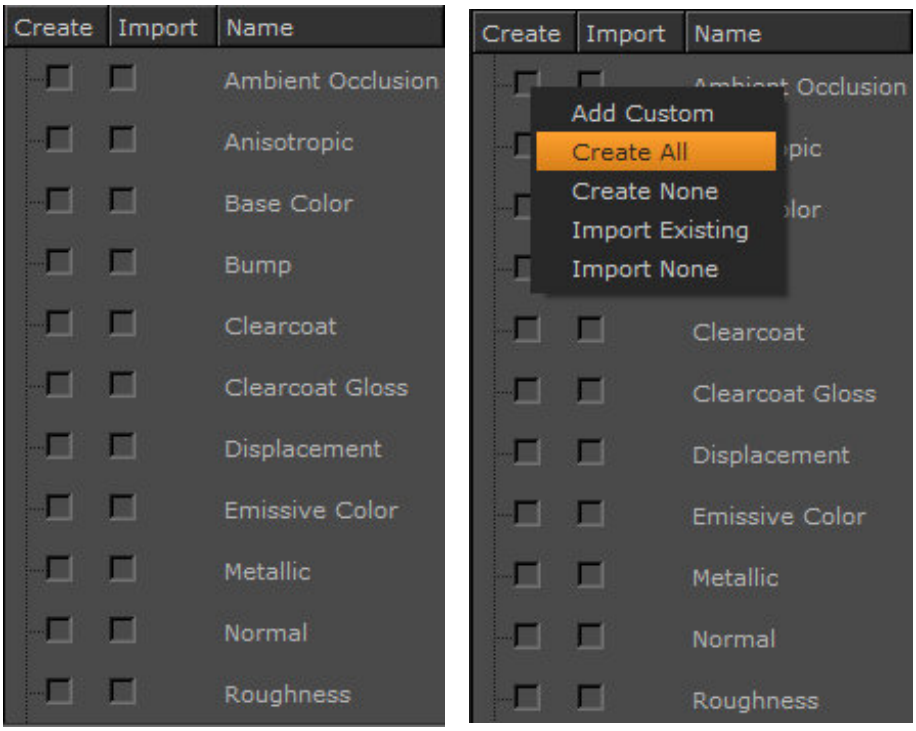
Next we'll specify the name and texture options, and then select the geometry.

Note: For more information on what kind of models Mari expects, see Modeling Requirements.This model displays with no errors. Mari loads the geometry and displays it in your workspace.

Tip: For more on the different options in the **New Project** dialog - and other dialog illustrated in this tutorial, see the Managing Projects, and the New Project Dialog.

Specifying Project Options

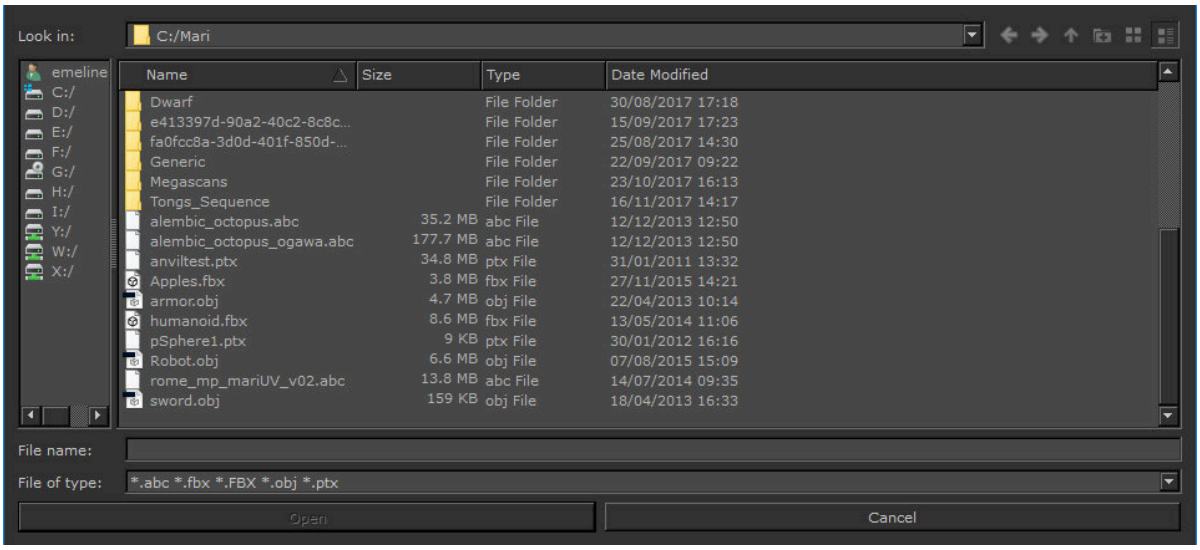
1. For the project **Name**, type **1-Blacksmith**.
2. For the **Texture** options, right-click on the list of channels and select **Create All**.



Loading Geometry

1. Next, to bring in the geometry, click .

   The **Pick Mesh** dialog box displays.

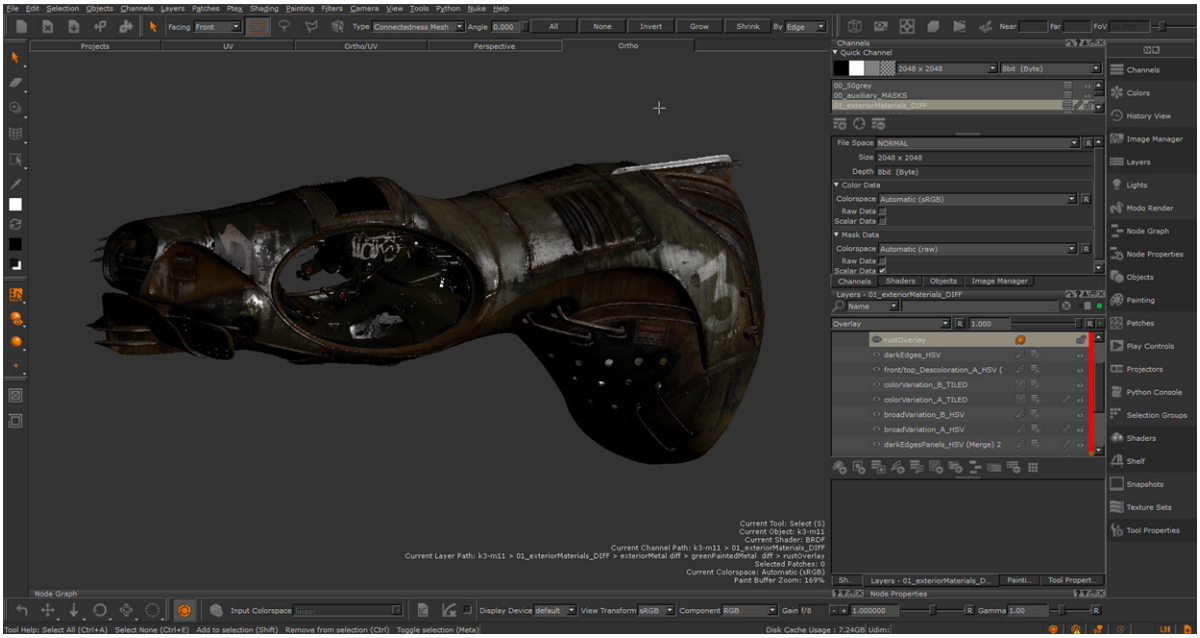


2. Navigate to your tutorial folder, click to highlight **blacksmith_body.obj**, and click **Open**.

> **Note:** When you load geometry, **Mesh Options** appear for the model you've entered in the **Path** field. For this tutorial, leave the mesh options as they are, but for more information on the **Mesh Options** refer to the *New Project Dialog* chapter in the *Mari Reference Guide* and the *Ptex* chapter in the *Mari User Guide*.

3. Click **Ok**.

   Mari conducts a quick "sanity check" on the model to make sure there are no obvious problems. If there are any problems, it displays **warnings** or **errors**. If there are warnings, you can continue. If there are errors, you must stop and fix them.



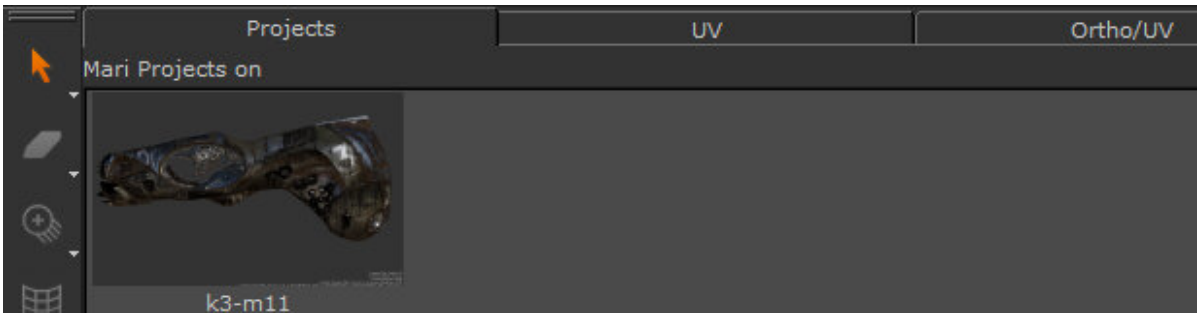
Tip: Depending on the size and complexity of a model, it can take several minutes the first time you load it into a Mari project. Once saved, however, the project should open quickly in future.

Saving Your Project

Save your project - again you can do this a few ways:

- from the **File** menu, select **Save**,
- click on the toolbar, or
- press the shortcut key **Ctrl**/**Cmd**+**S**.

Mari saves the project into its project directory. Now, if you click on the **Projects** tab at the top of the screen, you see your new project in the list.



Article: Take a look at the Q100042: Temporarily setting Mari's Project Location to a different folder article.

Where Do You Go from Here?

At this point, you should have a basic idea of what a Mari project is and how to create one. You learned how to:

- Name the project.
- Set texture options.
- Load geometry.
- Save the project.

Next, you'll use this project to practice some basics about setting the camera and lighting on the geometry in your project, before you begin to paint on it. When you're ready, proceed to Setting the View.

Setting the View

This lesson teaches you how to open a Mari project, and then set the view and lighting on your geometry (before painting on it).

Setting the **view** includes:

- orbiting
- zooming in or out
- panning
- viewing from six basic directions
- displaying a split view showing a 3D view plus UV patches

Setting the **lighting** includes:

- picking “flat”, “basic”, or “full” lighting,
- toggling shadows on or off,
- moving lights around and adjusting their properties.

Resources Needed

To complete this lesson, you need access to a computer that can run Mari, and the following sample file:

- **1-Blacksmith** - the Mari project file you created in Tutorial 1

How Long Should It Take?

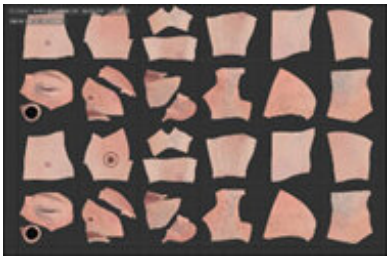
Plan on spending about 25 minutes to complete this lesson.

About Mari View and Lighting

The **view** in Mari is a view on the object you are painting. By default, Mari has four types of view:

- **UV** - this gives you a “flat” view of the patches in the model.

- **Perspective** - this show the model as a whole in 3D space, through a perspective camera.
- **Ortho** - this shows the model as a whole in 3D space, through an orthographic camera.
- **Ortho/UV** - this is a split screen, showing both the UV patches and the orthographic view.



UV View

Ortho View

Ortho/UV View

You can move each view separately.

> **Tip:** You can also use the , , and toolbar buttons or the **F10**, **F11**, and **F12** keyboard shortcuts to activate a particular view on any tab.

Within the views, you can:

- **Roll** or **orbit** the model (3D views only). Rolling turns the model on one axis, while orbiting lets you turn the model in all three axes. Think of rolling as like turning the model on a turntable, while orbit is turning it around in midair.
- **Pan** to move the view across the model.
- **Zoom** in or out.
- **Focus** on a selected patch.

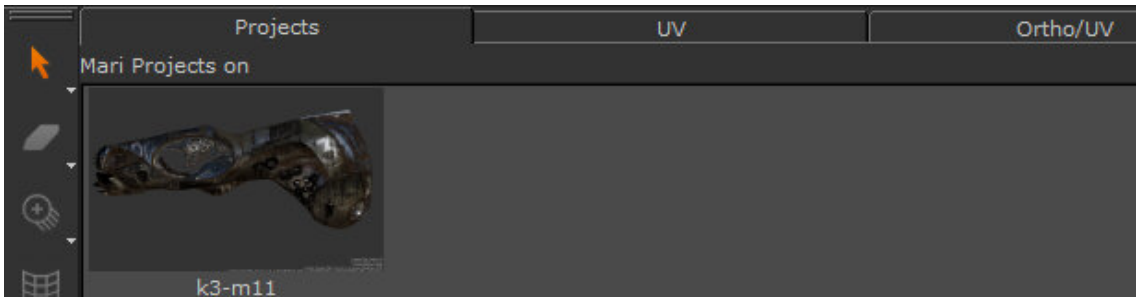
You separately configure the **lighting** on the object by selecting preset settings for “flat”, “basic”, or “full”, or specifying fine details such as shading parameters and separate lighting for each of the four configurable lights.

Opening Your Project

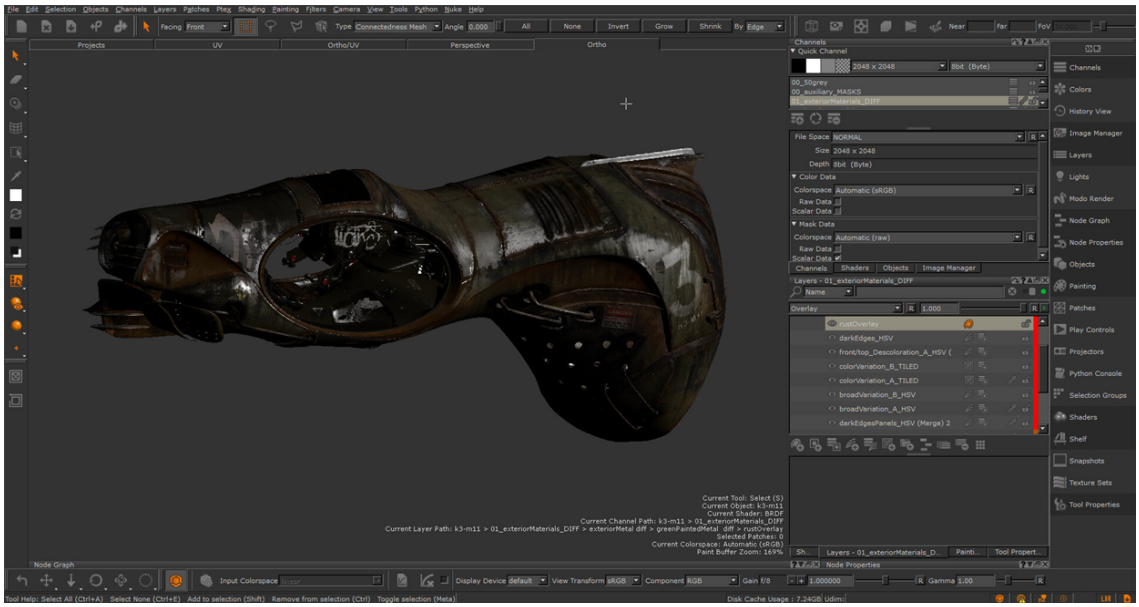
1. Start Mari.

   The Mari workspace displays.

2. On the **Projects** tab, double-click on your **1-Blacksmith** project to open it.



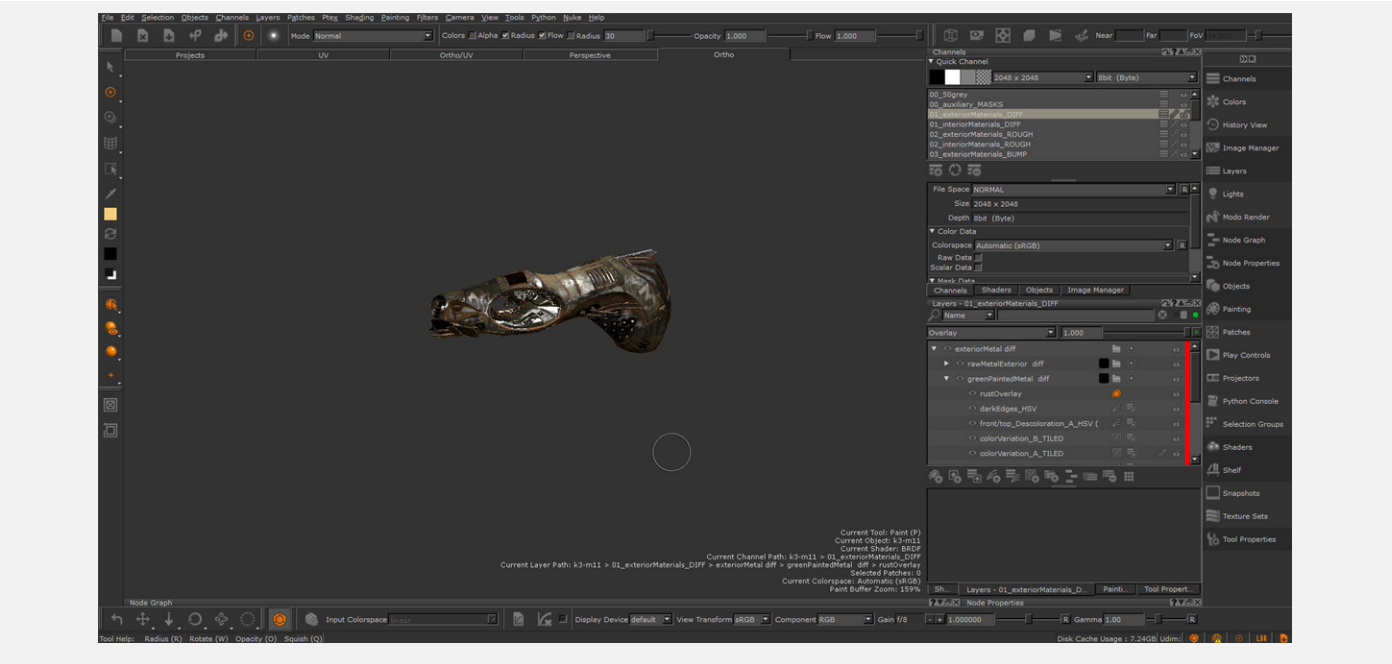
Your project opens and switches to the **Ortho** view.

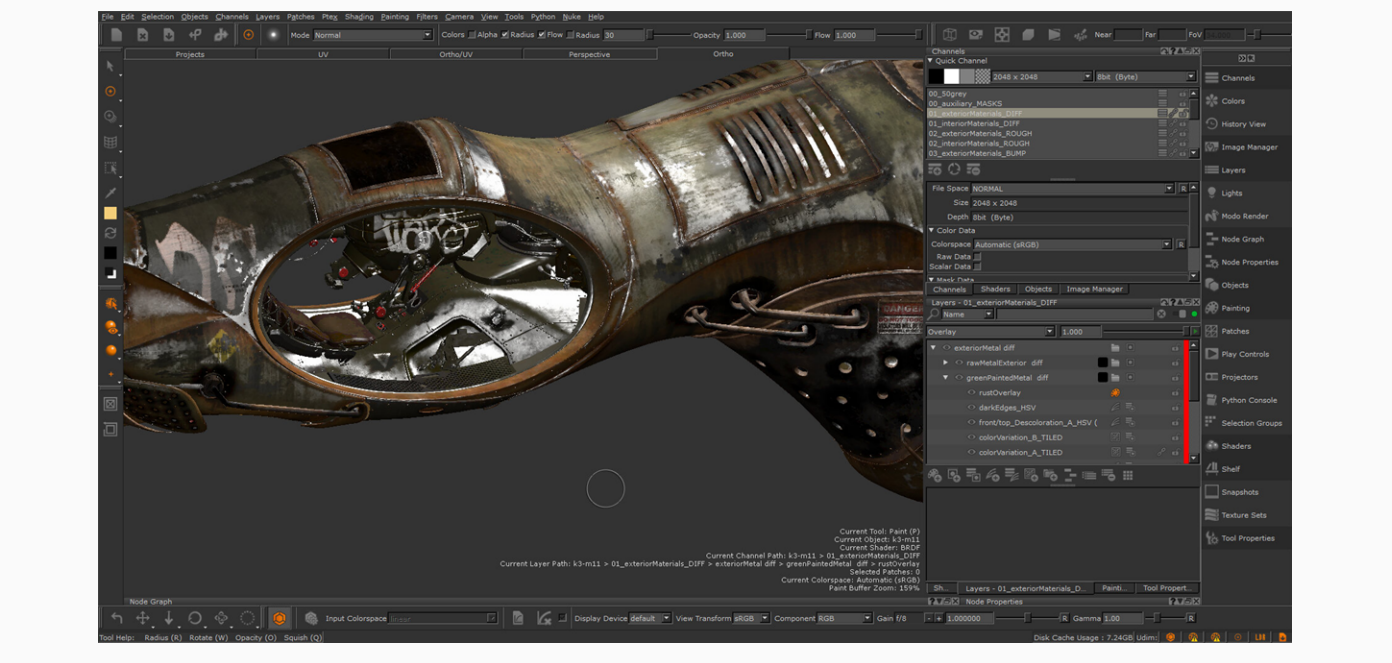


Adjusting the View

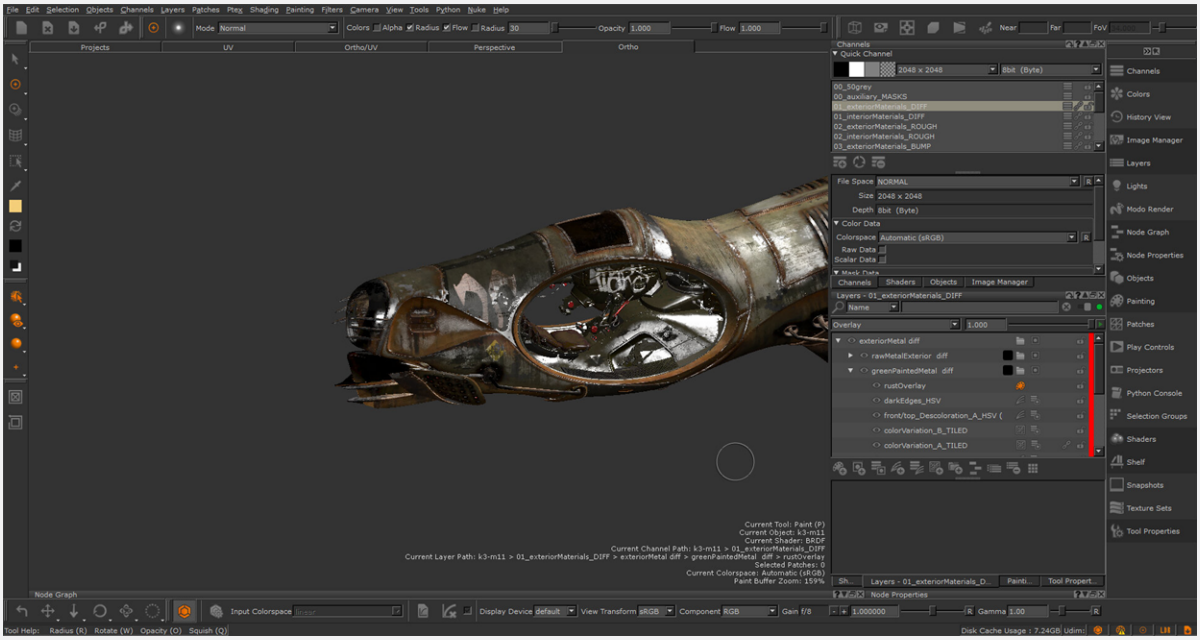
To adjust the view on your project, you can zoom, pan, orbit, select one of six preset views, display your model and patches in a split screen, or use perspective and orthographic views.

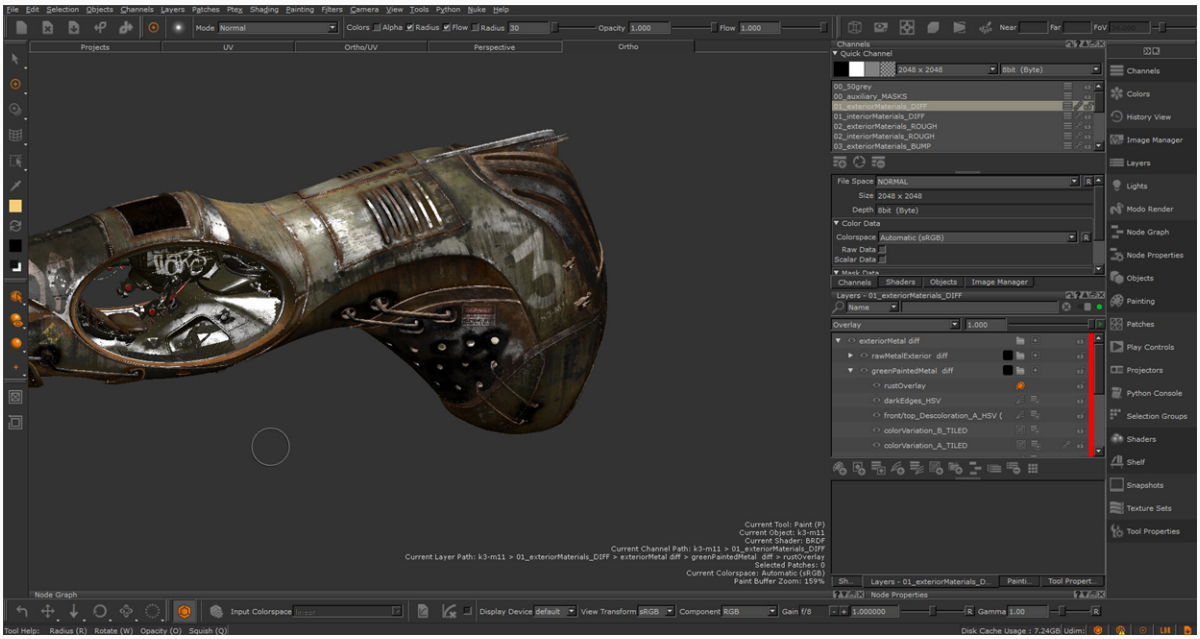
1. To **zoom out** and **in**, hold down the **Alt** key, right-click and drag the cursor to the left and right.



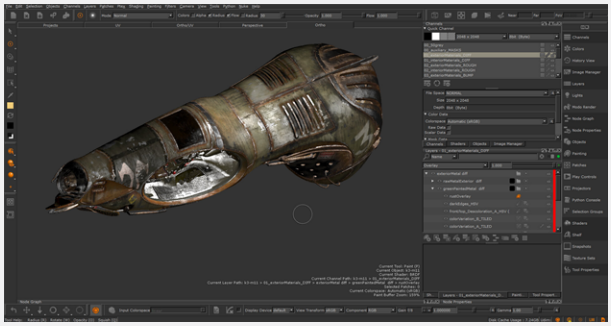


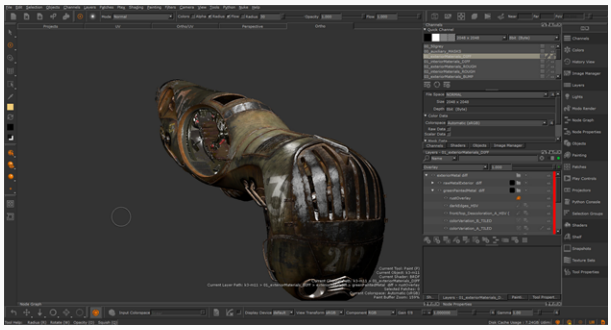
2. To **pan**, hold down the **Alt** key and middle-click (or hold **Alt** and **Shift** and left-click and drag).





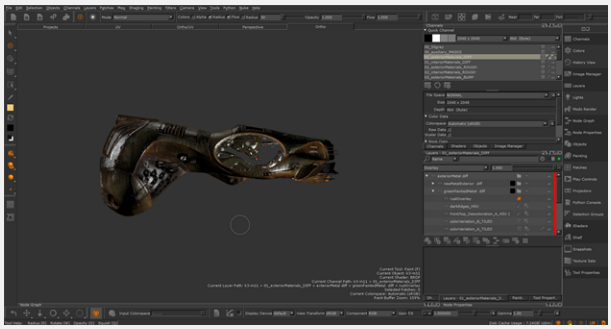
3. To **orbit**, hold down the **Alt** key, left-click and drag.

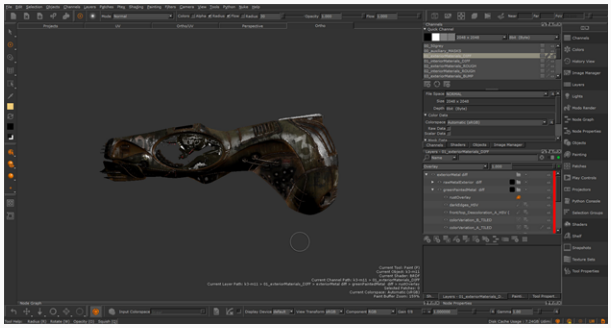


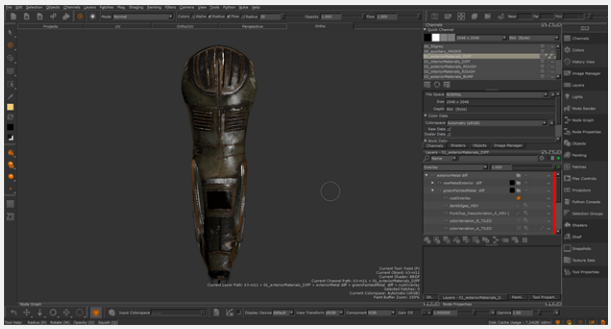


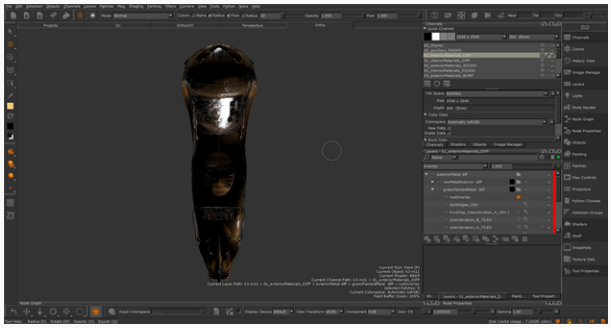
Tip: You can also roll by pressing **Ctrl**/**Cmd**+**R**, left-clicking, and dragging. When you orbit or roll, the pivot point is located in the center of the current view.

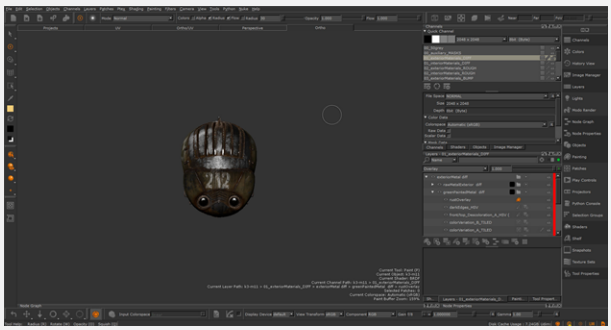
4. To view from six basic preset angles (camera “left”, “right”, “top”, “bottom”, “front”, and “rear”), press **1**, **2**, **3**, **4**, **5**, or **6**, or select the angle from the **Camera** menu.

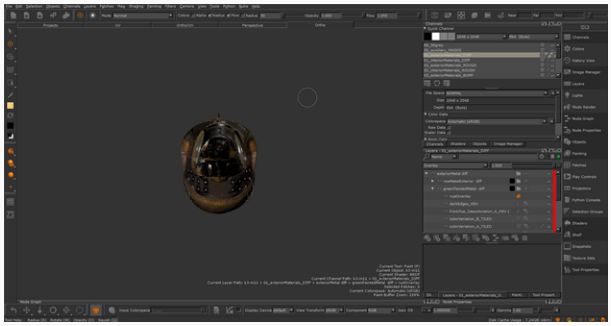






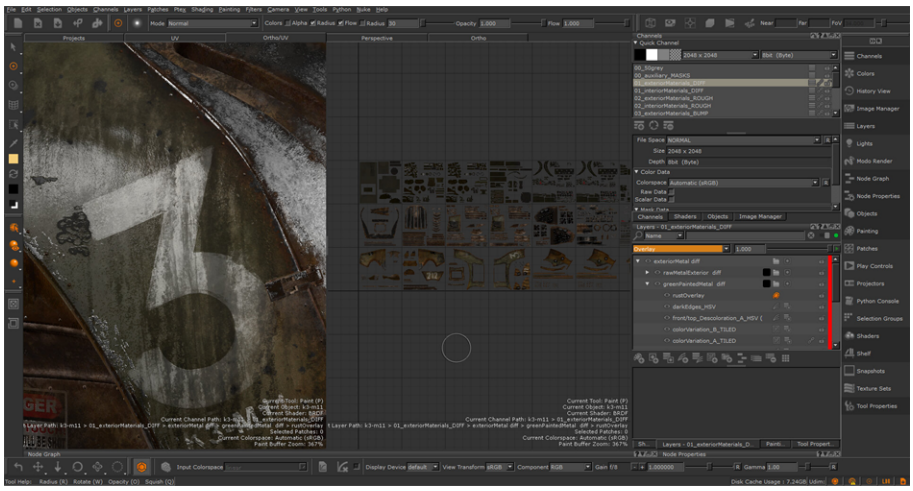






5. To see a flat view of the UV patches on the model, click the **UV** tab.

   To see a split screen showing both the Ortho and UV views, click the **Ortho/UV** tab.



6. To see a perspective view of your model, click the **Perspective** tab (or press **F11**).

   In the **Perspective** view, you can alter your model's appearance in the viewer by entering **Near**, **Far**, and **FoV** values in the **Canvas** toolbar. See Setting the Perspective Camera Details.

7. To see an orthographic view of your model, click the **Ortho** tab (or press **F10**). This is the default view mode, that is, the view you see when you start Mari.

> **Note:** The model is in the same position in steps 6 and 7; only the view mode has changed. Notice that the model's arms appear closer to the body due to the application of perspective.

8. To switch the view to display all selected patches, press **A**.

   If you don't have any patches selected, the view focuses on the model as a whole (changing to show the whole model).

Tip: If you find that the view starts cutting through the front clipping plane, try pressing **A**.

9. To center the view at a particular place on the model, move the cursor there and press **F**.

   At the current size and angle, the view changes to focus on the cursor point. If you hover the cursor over the canvas where no part of the object is present, pressing **F** brings the focus on the entire object.

Tip: You can paint in any of the views.

Moving the view bakes the current paint if **Bake Behavior** is set to **AutoBakeAndClear**. You can set this on the **Painting** palette, under **Projection** (if you can't see the **Painting** palette, select **View** > **Palettes** > **Painting** to display it).

Experiment: Try zooming, panning, and orbiting with the camera. Change between **Ortho** and **Perspective** modes. Have you noticed that objects closer to the camera distort far more than objects further away?

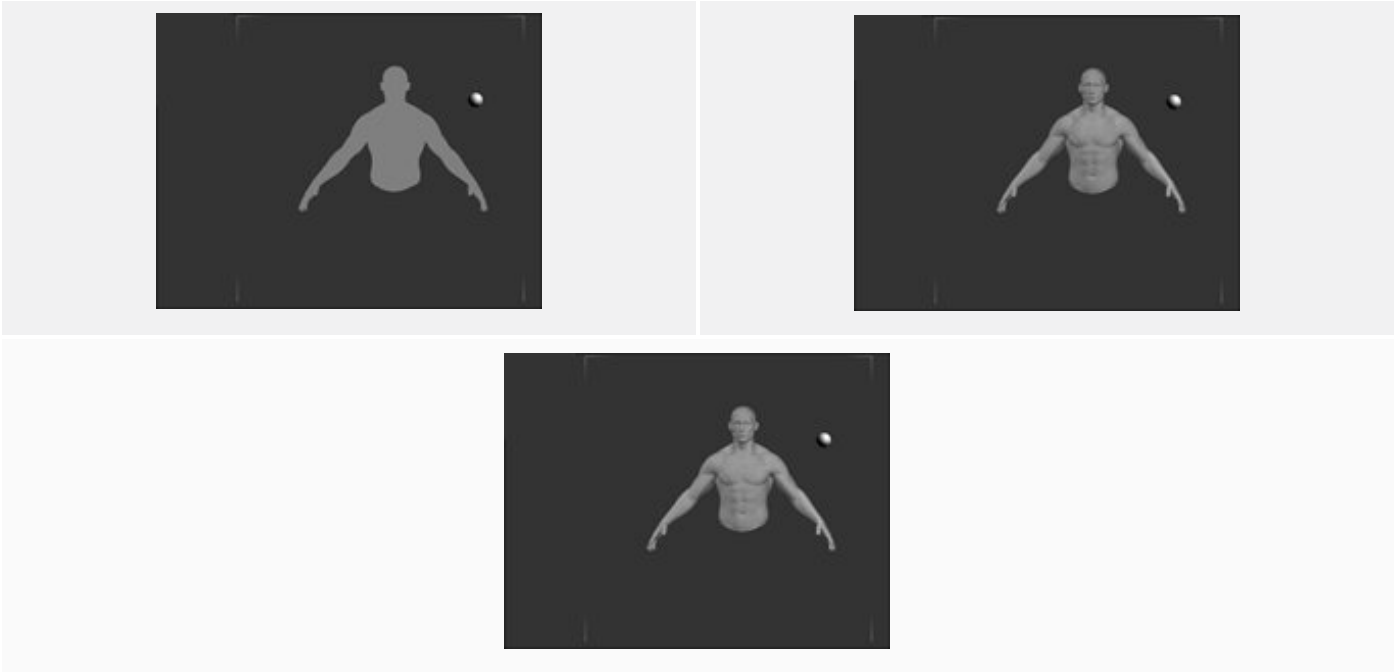
Adjusting the Lighting

Mari comes with five lighting modes: flat, basic, basic with shadows, full, or full with shadows. In basic and full modes, you can move the lights around the model.

1. To change the lighting:
   - from the **Shading** menu, select **Flat**, **Basic**, **Basic with Shadows**, **Full**, or **Full with Shadows**,

     OR
   - on the **Project Controls** toolbar, click , , , , or .

     If you can't see the **Project Controls** toolbar, right-click on the toolbar area and choose **Project Controls** from the dropdown menu.



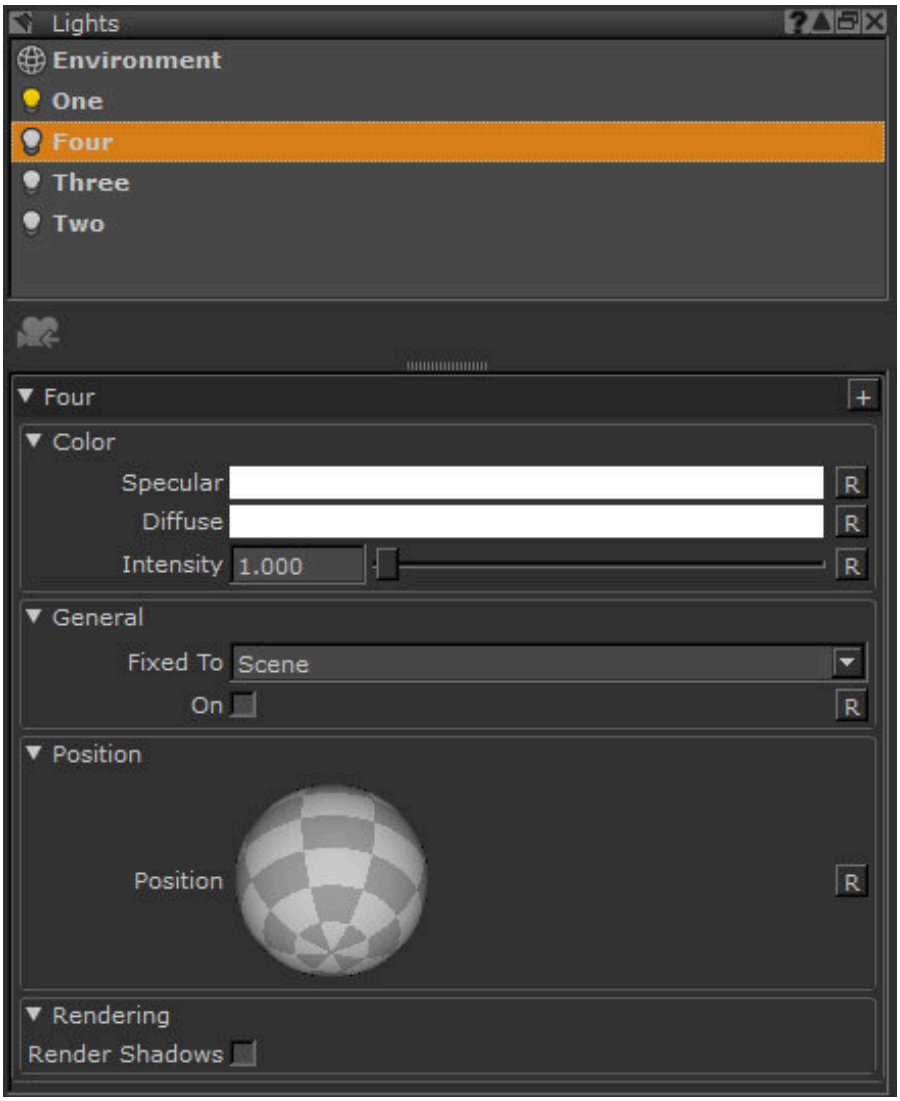
2. Notice the four lights on the canvas (you may need to select the **Transform Selected Objects** tool to see them).

   By default, the lights appear as gray spheres .

> **Tip:** Ensure that the lights are on by clicking any of the light bulbs in the **Lights** palette, and the Render **Lights** option in the **Display Properties** dialog. You may need to zoom out to view the lights on the canvas.

3. To move a light around, select the **Transform Selected Objects** tool (if you didn't already). Click on a light and drag it around the canvas, or use the transform handles to adjust it.

   As you move the light, you can see the effect on the model.
4. You can also make fine adjustments to lighting using the **Lights** palette. If it's already open, click the **Lights** tab to give it focus; or if it's closed, right-click in the toolbar area on top of the Mari workspace and select **Lights** to open it.



The **Lights** palette displays details of four basic lights, which you can enable or disable, and customize by adjusting their **Color** and **Fixed To** information, and an environment light, which you can adjust to add environment lighting from an image. For details, see the Customizing the View.

5. Lighting can be relative to the **Scene** or a **Camera**:
   - **Scene** (default) - the light is fixed to the model, for example if a light is set at the model's back, it always points at the model's back, no matter how you move the view.
   - **Camera** - the light is fixed to a camera view, for example you can have a light always shining from above the camera, allowing you to move the model around and cast light on different areas.

Video: Watch the demo.

6. A single light can be moved to the position of the current camera by right-clicking the light and selecting **Move to Camera Position** or clicking on the **Move to Camera Position** icon when the desired light is selected.
7. Select the **Render Shadows** checkbox to produce shadows for an individual light.

Shadows render and display immediately for the selected light.

Note: Mari saves custom lighting with the project. If you need the same lights in another project, you need to set them up separately.

Where Do You Go from Here?

At this point, you should have a basic idea of the different options for adjusting the view and lighting on your model. You learned how to:

- Zoom, pan, and orbit.
- Select one of six preset angles.
- Toggle split view.
- Change to flat, basic, or full lighting.
- Move and configure lights.

Now that you know how to create a project, load geometry, and adjust the position of the view and lighting, you can (finally) begin to paint! When you're ready, proceed to Painting.

Painting

This lesson teaches you the basic steps to painting on a model in Mari. This includes, selecting a brush, selecting a color, and painting with that brush and color.

Resources Needed

To complete this lesson, you need access to a computer that can run Mari, and the following sample file:

- **1-Blacksmith** - the Mari project file you created in Tutorial 1 and configured in Tutorial 2

How Long Should It Take?

Plan on spending about 25 minutes to complete this lesson.

About Painting in Mari

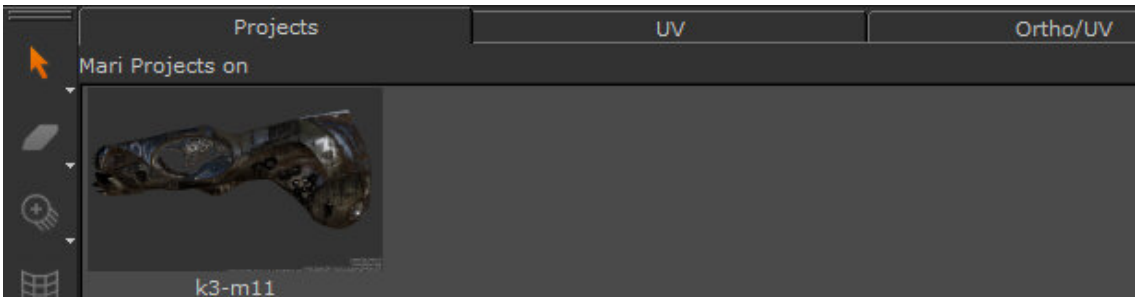
Painting in Mari is similar to painting in other standard paint programs - you can select and configure both brushes and colors, and then use them to paint an object. In Mari, you configure brushes in the **Tool Properties** palette and colors through the **Colors Palette**. You also have the option of saving colors and brushes to your **Shelf** for easy access. The Mari **Shelf** comes with lots of presets, as well as any items you have created or customized for yourself (and saved there).

After you paint (also as with other programs), in Mari you need to **bake** the paint buffer into the object. Mari includes various ways of baking onto selected patches, including an **Autobake** option that bakes by default every time you move the camera.

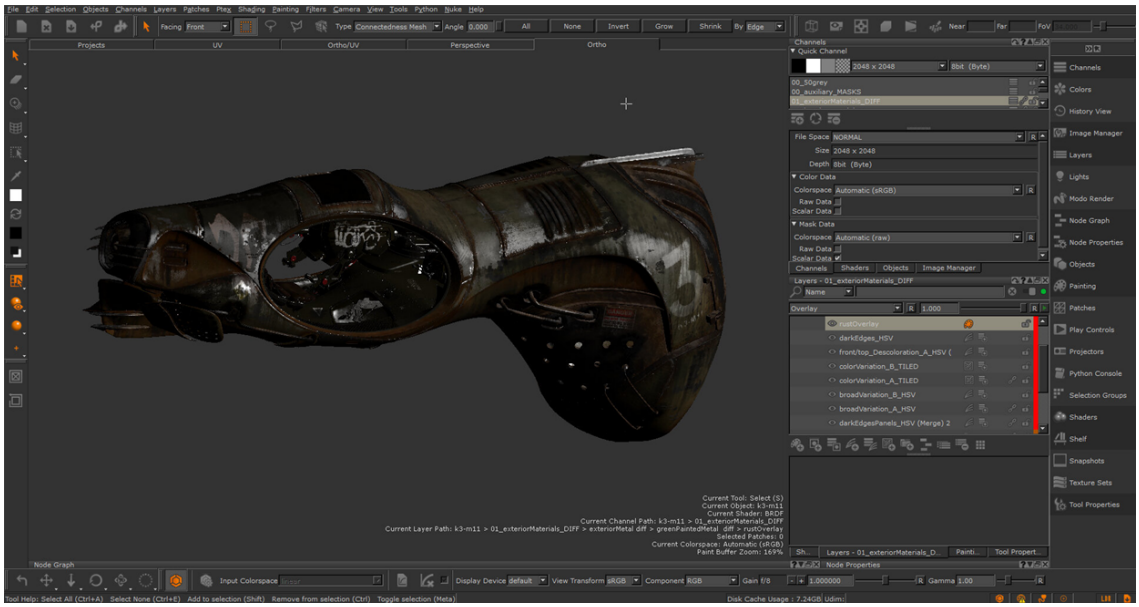
Opening Your Project

1. Start Mari.

   The Mari workspace displays.
2. On the **Projects** tab, double-click on your **1-Blacksmith** project to open it.



   Your project opens and switches to the **Ortho** view.



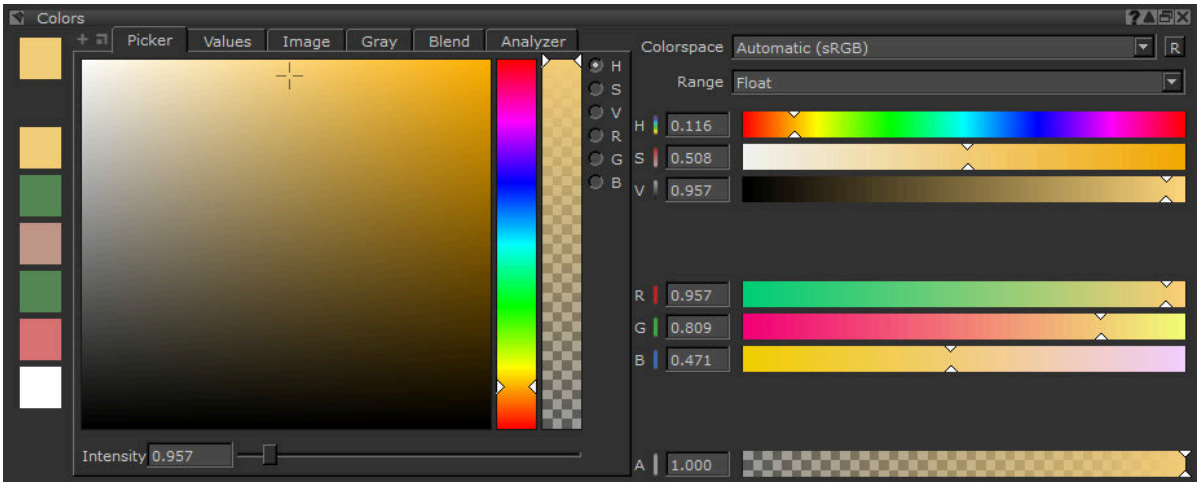
Opening the Colors and Shelf Palettes

Basic painting on your model involves selecting a brush, selecting a color, and using them to paint. In the steps below, you'll select a color from the **Colors** palette and a brush from your **Shelf**. To begin, let's open and arrange those palettes.

> **Tip:** By default, the **Colors** and **Shelf** palettes are already open when you start Mari (but follow the instructions below to see how to open and close palettes).

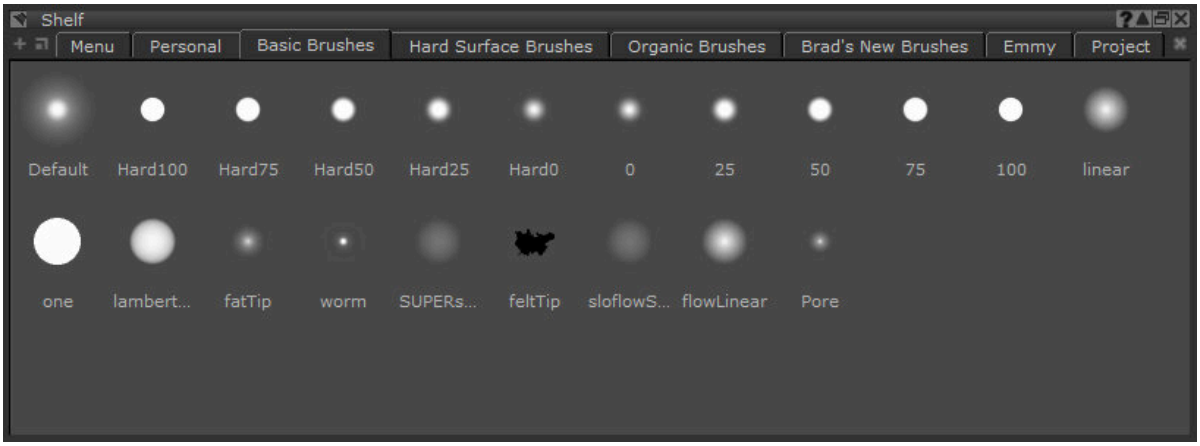
1. To open the **Colors** palette:
   - from the **View** menu, select **Palettes** > **Colors**, or
   - right-click in the toolbar area and select **Colors** from the dropdown menu.

   The **Colors** palette displays.



2. To open the **Shelf** palette:
   - from the **View** menu, select **Palettes** > **Shelf**, or
   - right-click in the toolbar area and select **Shelf** from the dropdown menu.

   The **Shelf** palette displays.



The shelf can hold your customized sets of colors and brushes - but when you first start Mari, the shelf is empty.

Experiment: The *MariReference Guide* outlines the different options for arranging palettes. Take a few minutes to experiment with docking, undocking, sizing, moving, collapsing, and stacking palettes (including the **Colors** palette and **Shelf**). Over time, you'll find a configuration that works best for you. And when you exit Mari, it remembers that configuration for the next time you re-start the program.

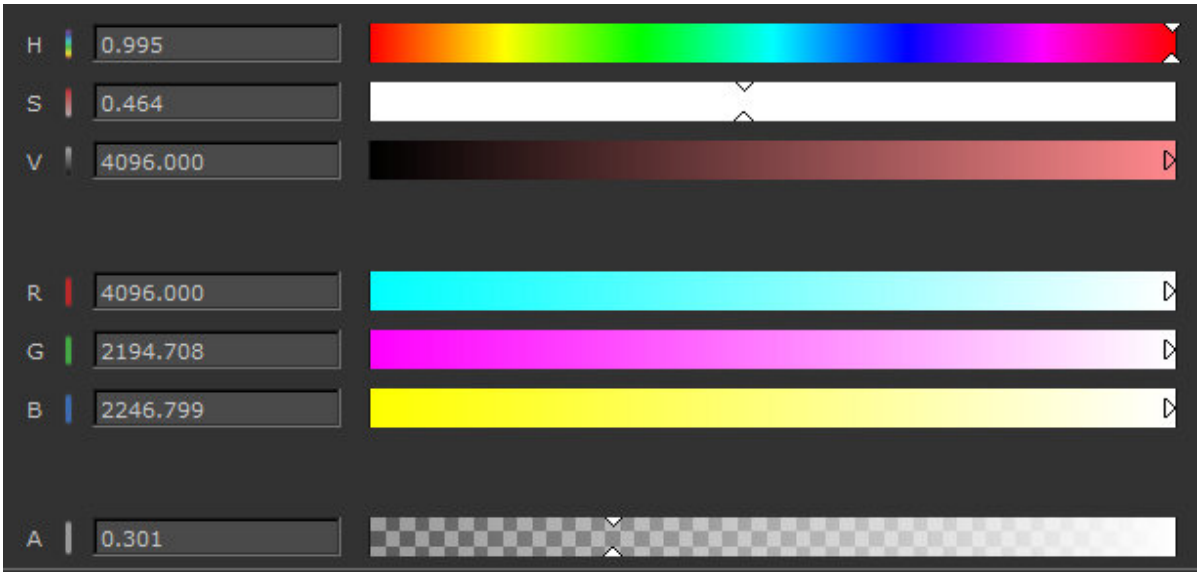
Selecting a Color

1. You can select a color from the **Colors** palette a few different ways:
   - click in the color field

- click to select the **H**, **S**, **V**, **R**, **G**, or **B** next to the vertical sliders (to the right of the color field), then click and drag the sliders up or down to select color and alpha

- enter an **Intensity** value or drag the slider to adjust the intensity of the color

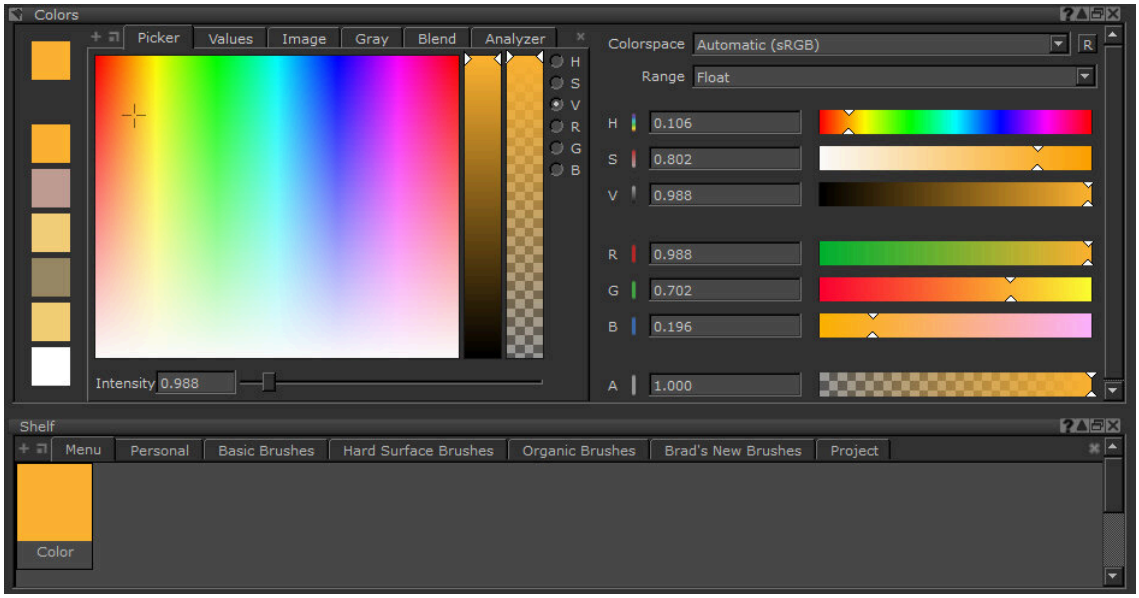
- on the right side of the **Colors** palette, drag the horizontal sliders left or right, or type specific values in the boxes, to specify HSV, RGB, and Alpha.



The selected color displays in the foreground swatch.

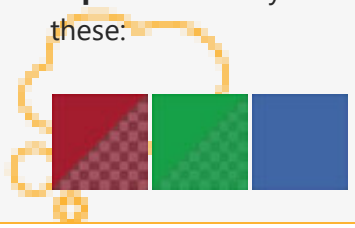
Tip: In addition to adjusting the intensity with the **Intensity** slider, you can also adjust it by selecting the **V** component, to the right of the vertical color sliders, and dragging the left slider or dragging the RGB slider.

2. When the color you want is selected in the **Colors** palette, drag the swatch to the **Shelf**.



You don't have to drag the color to the shelf to use it, but it's useful for storing colors you use regularly.

Experiment: Play around with all the different options for selecting colors. See if you can get these:



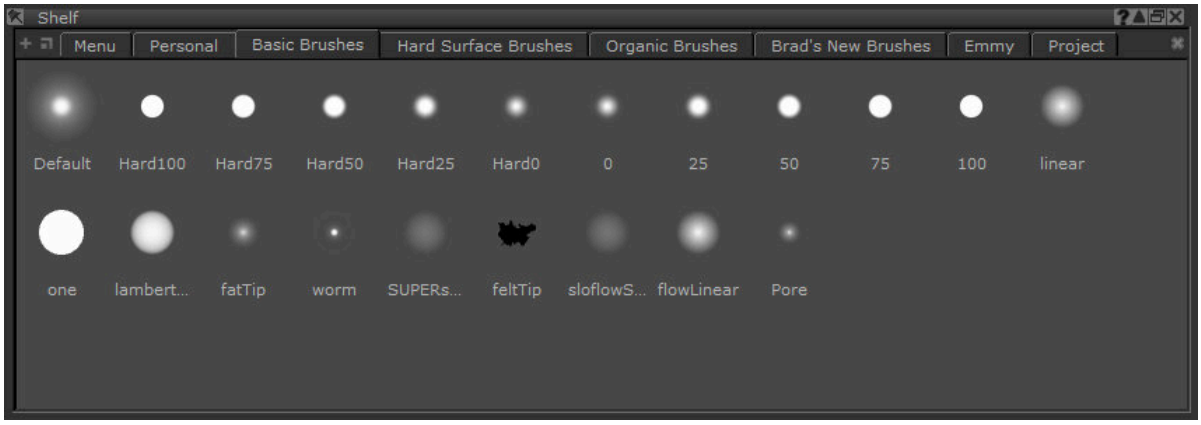
Tip: If you hover the mouse over a swatch, it displays RGB, HSV, and Alpha values:

Selecting a Brush

To select a brush:

1. In the **Tools** toolbar, select a painting tool such as **Paint**, **Blur**, **Vector Paint**, **Paint Through**, **Gradient**, and **Clone Stamp**.
2. Open the **Shelf** palette:
   - from the **View** menu, select **Palettes** > **Shelf**, or
   - right-click in the toolbar area and select **Shelf** from the dropdown menu.

The **Shelf** palette displays.



The **Shelf** palette has seven shelves:

- **Menu** - items that you can select from the **F9** pie selection control menu.
- **Personal** - selected items you use regularly.
- **Basic Brushes** - a set of predefined basic brushes.
- **Hard Surface Brushes** - a set of predefined hard surface brushes.
- **Organic Brushes** - a set of predefined organic brushes.
- **Brad's New Brushes** - a set of predefined brushes.
- **Project** - items just for the current project.

3. Click one of the seven shelves.
4. Click a brush to select it.

Video: At the bottom of the **Tool Properties** palette is a scratch area for you to test out the selected brush. You can press **N** to swap back to the last brush tip you used. This lets you quickly swap back and forth between two brush tips. Watch the demo.
This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Tip: You can also customize brushes in the **Tool Properties** palette, and save them to the **Shelf** palette. This basically involves selecting a preset brush from the **Shelf** palette, custom-configuring that brush by selecting options on the **Tool Properties** palette, and saving it by dragging and dropping the brush icon in the **Tool Properties** toolbar of your selected painting tool to the **Shelf** palette's **Menu**, **Personal**, **Project** or custom shelf. See Customizing a Brush for more information.

Paint

1. When you have selected a color and a brush (from the **Colors** and the **Shelf** palettes), you can use them to paint on the model. First select the paint tool:
   - click on the toolbar, or
   - press **P**.
2. Then paint.

Tip: You can undo and redo any action by pressing **Ctrl/Cmd+Z** and **Ctrl/Cmd+Y**, or clicking and on the **Project** toolbar, or selecting **Undo** and **Redo** from the **Edit** menu.
And, you can undo or redo multiple actions at once, by clicking on a specific action in the **History View** palette (from the **View** menu, select **Palettes** > **History View**).

The **History View** palette lists all actions you've performed since opening the project. There is no limit on the size of the action history, but it clears when you close the project.

Redoing an action in Mari only reverses a previously undone action. You cannot use the Redo command in Mari to repeat a command - it's reversing an Undo (not a “do that again”).

Baking Your Painting

1. As with other paint programs, what you paint stays in a paint buffer until you “bake” it onto the model. To bake in Mari, make sure all patches with unbaked painting are selected, and then:
   - from the **Painting** menu, select **Bake**, or
   - click on the status bar, or
   - press **B**.

Note: the **Painting** palette (see the Painting Palette) also includes a **Bake Behavior** option that, by default (**AutoBakeAndClear**), automatically bakes every time you adjust the view.

2. When you exit Mari, **do not save** the project. (We'll start the next lesson with the unpainted blacksmith you imported in Tutorial 1.)

Where Do You Go from Here?

This lesson covered the most basic options for painting in Mari:

• selecting a color in various ways, and dragging it to the **Shelf** palette

• selecting a brush from the **Shelf** palette

• painting on the model and baking.

The next tutorial, Painting Through and Clone Stamping, covers two main options for using images to create textures in Mari.

Painting Through and Clone Stamping

This lesson introduces using images to create textures in Mari. This includes “painting through” an image onto your model and “clone stamping” part of an image onto your model.

Resources Needed

To complete this lesson, you need access to a computer that can run Mari, and the following sample files:

• **1-Blacksmith** - the Mari project file you created in Tutorial 1 and used in Tutorial 2 and Tutorial 3

• **Tattoo.png** - an image of a tattoo, in your Mari tutorial directory

How Long Should It Take?

Plan on spending about 25 minutes to complete this lesson.

About Painting from Images in Mari

Like with other paint programs, Mari lets you superimpose image files onto the surface of an object. To manage image files, Mari includes the **Image Manager** palette. To paint with images you've loaded into the **Image Manager** palette, Mari has two main tools: **Paint through** and **Clonestamp**:

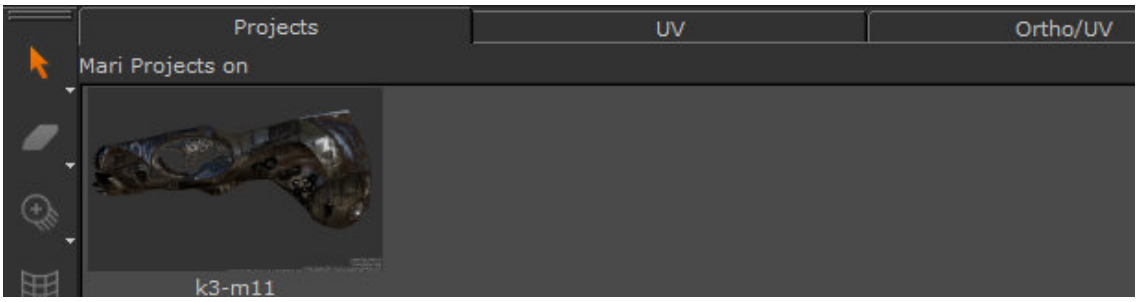
- The **Paint through** tool ( ) lets you position an image *over* an object, and then as you paint, copy from the image directly onto what's under the brush.
- The **Clonestamp** tool ( ) lets you position an image *alongside* an object, with a source point on the image relative to where the brush is on the object. Then as you paint, the source point on the image moves in sync with the brush on the object, and Mari copies from the image onto the corresponding spot under the brush on the object.

  You can also clone stamp from painting already on an object, in the paint buffer, or in another channel.

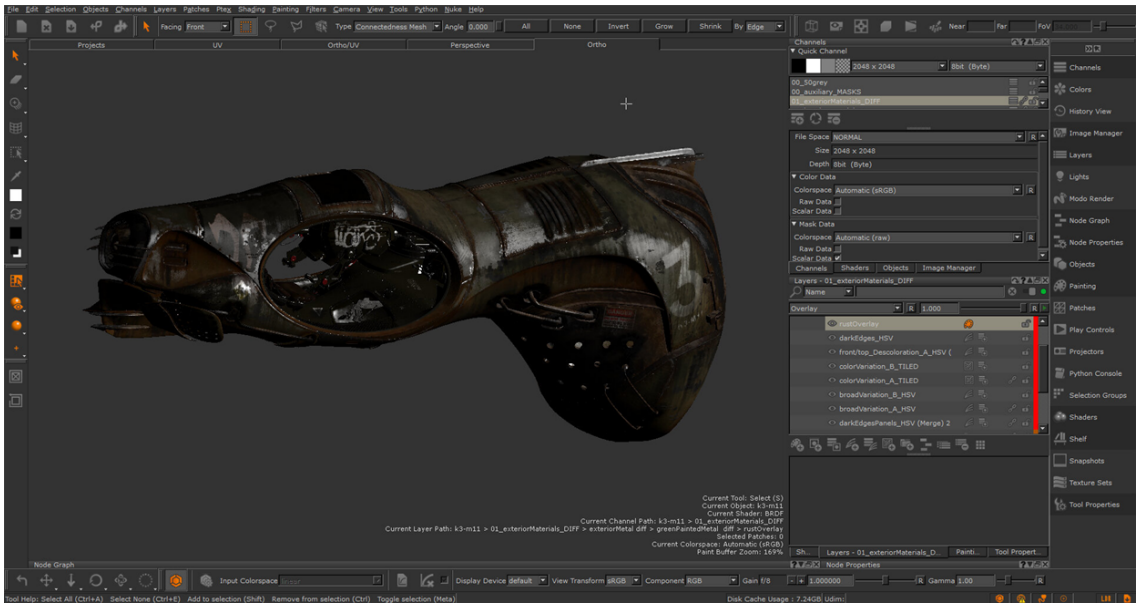
Opening Your Project

1. Start Mari.

   The Mari workspace displays.
2. On the **Projects** tab, double-click on your **1-Blacksmith** project to open it.



Your project opens and switches to the **Ortho** view.

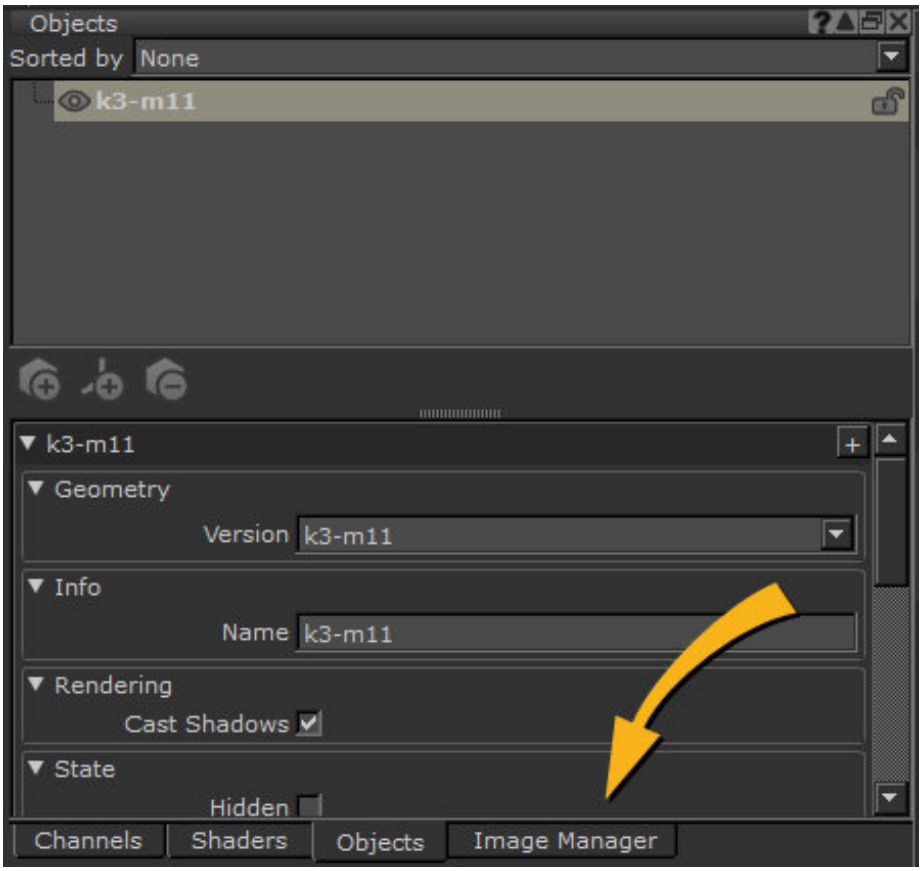


Loading an Image

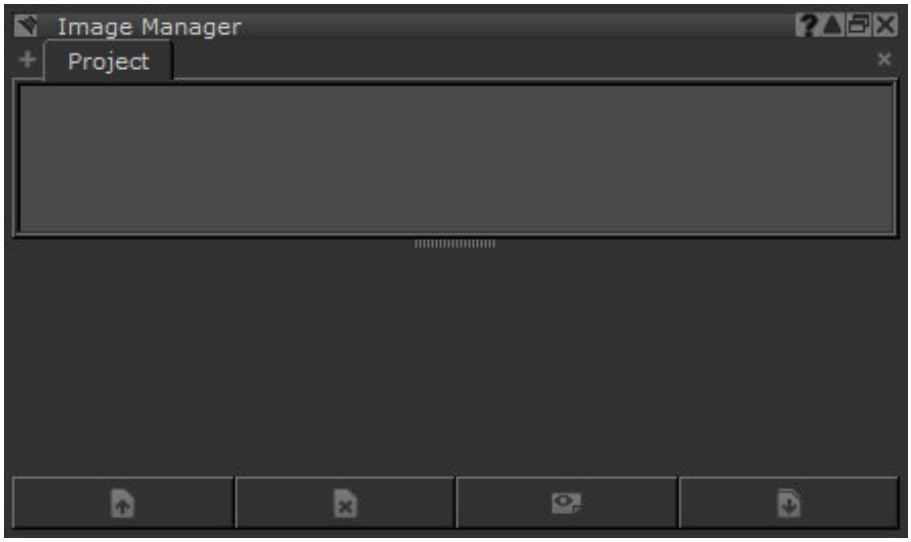
For both painting through and clone stamping, we'll use an image of a tattoo that looks like this:

We'll start by loading that image in Mari's **ImageManager** palette.

1. To open the **ImageManager**:
   - click the **ImageManager** tab if it is already open but hidden (as in the default Mari layout),

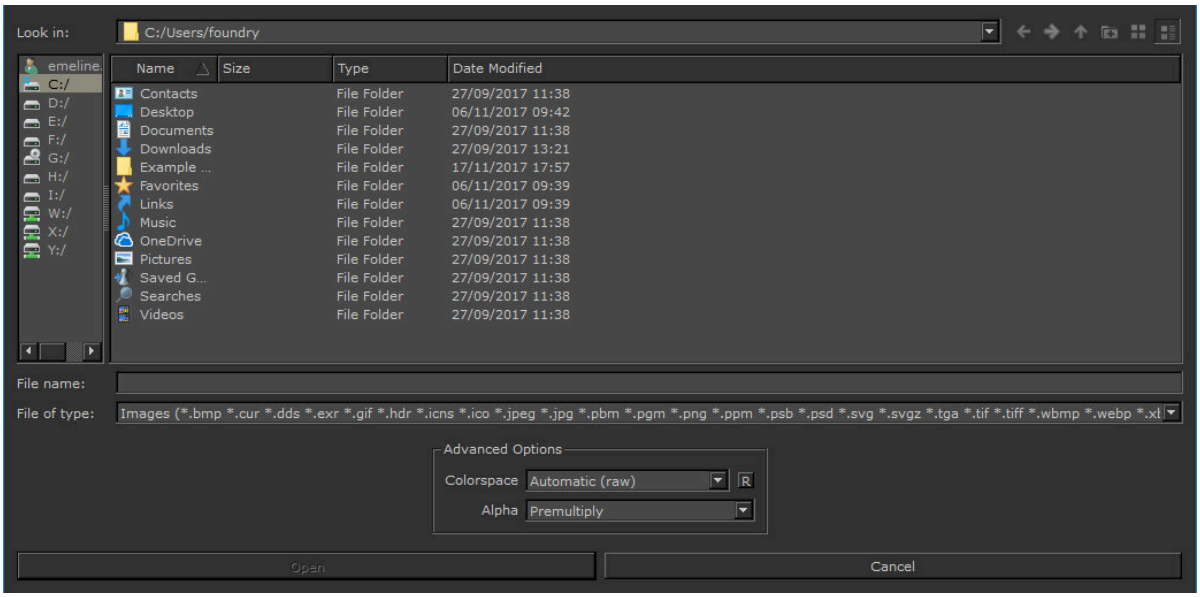


- from the **View** menu, select **Palettes** > **ImageManager**, or
- right-click in the toolbar area and select **ImageManager** from the dropdown menu. The **ImageManager** palette displays.



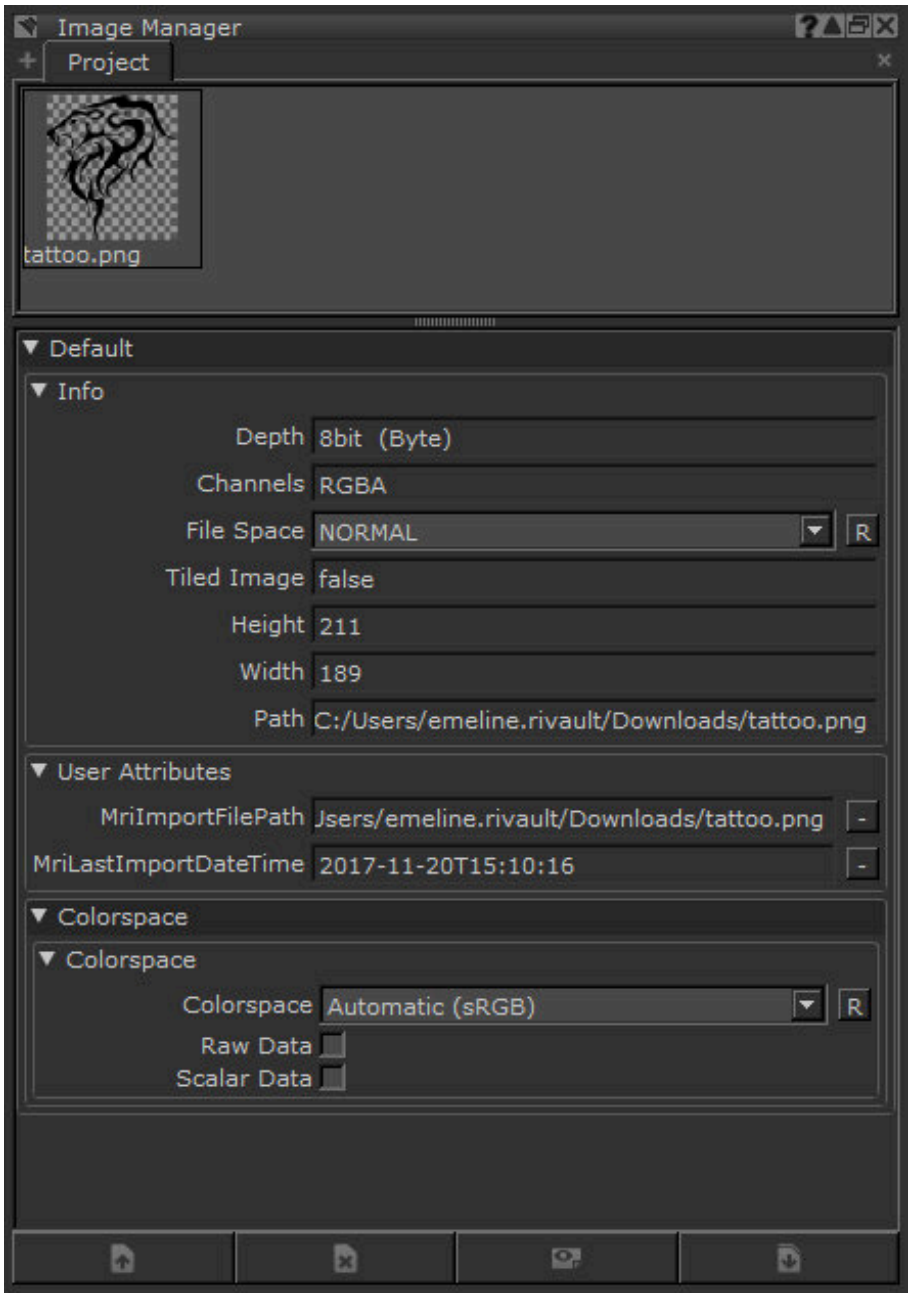
2. To load an image, click .

   The **Openan Image** dialog box displays.



3. Navigate to your tutorial folder, and select Tattoo.png.

   The Tattoo.png image displays in the **ImageManager**.



Paint Through

Painting through an image basically involves:

- selecting the **PaintThrough** tool,
- dragging an image onto the canvas, and
- painting.

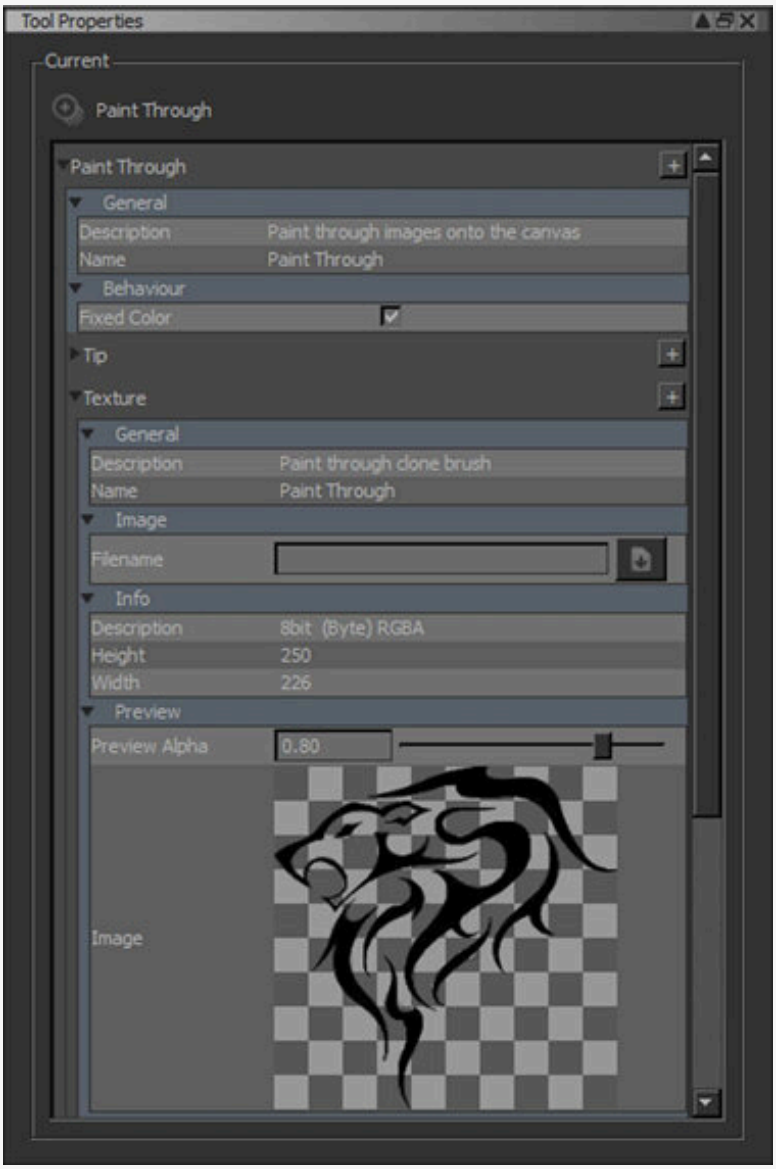
1. Select the **PaintThrough** tool by:

- clicking on the toolbar, or
- pressing **U**.

2. Drag the image from the **ImageManager** and drop it onto the model.

3. Move and resize the image:
   - To resize the image, hold down **Ctrl**/**Cmd**+**Shift**, click and drag.
   - To move the image around the canvas, hold down **Shift**, click and drag.
   - To rotate the image, hold down **Ctrl**/**Cmd**, click and drag.

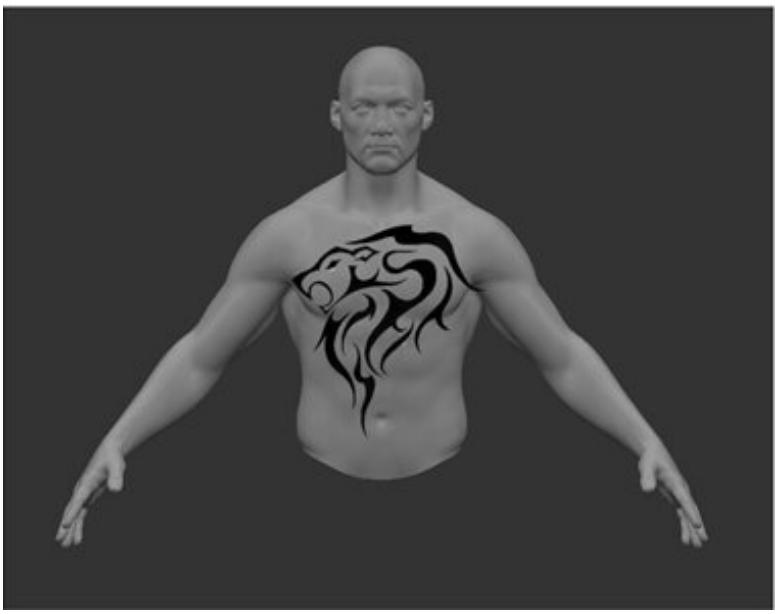
Tip: The **Tool Properties** palette includes many options for manipulating paint-through images.



You can also double-click the image in the **Image Manager**, and crop a smaller portion to use for painting through and clone stamping.

Experiment: Try sizing, moving and rotating the image. See if you can get it to look like this:

4. Paint through the image onto the model (hold down the left mouse button and move, as you do in any standard paint program).
5. When you're done painting through, press **P** to go to the **Painting** tool to see just the painted-on object.



Tip: To hide the overlying image, press the **?** (question mark) key. To paint the whole image onto the model in one step, press the **'** (apostrophe) key. To repeat the source image (so you can paint past the edges and have the source image repeat), press the **;** (semicolon) key.

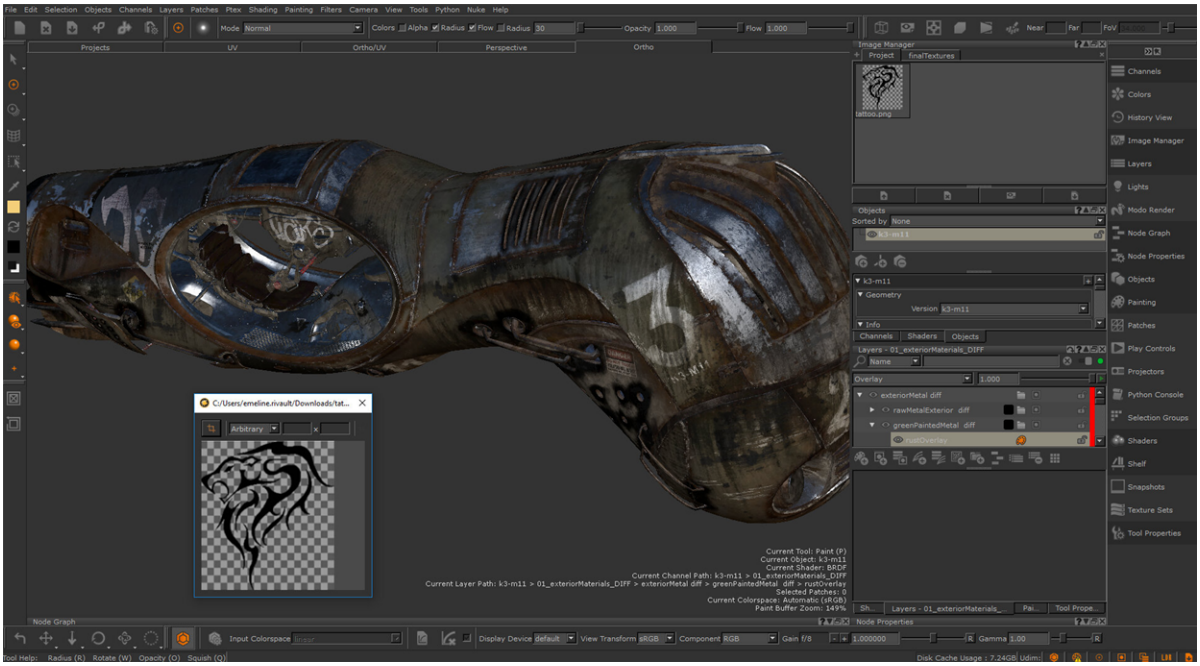
Clone Stamp

Clone stamping is like painting through, but the image does not have to sit directly on top of the model.

1. Clear the painted-through textures:
   - from the **Painting** menu, select **ClearPainting**, or
   - click on the toolbar.

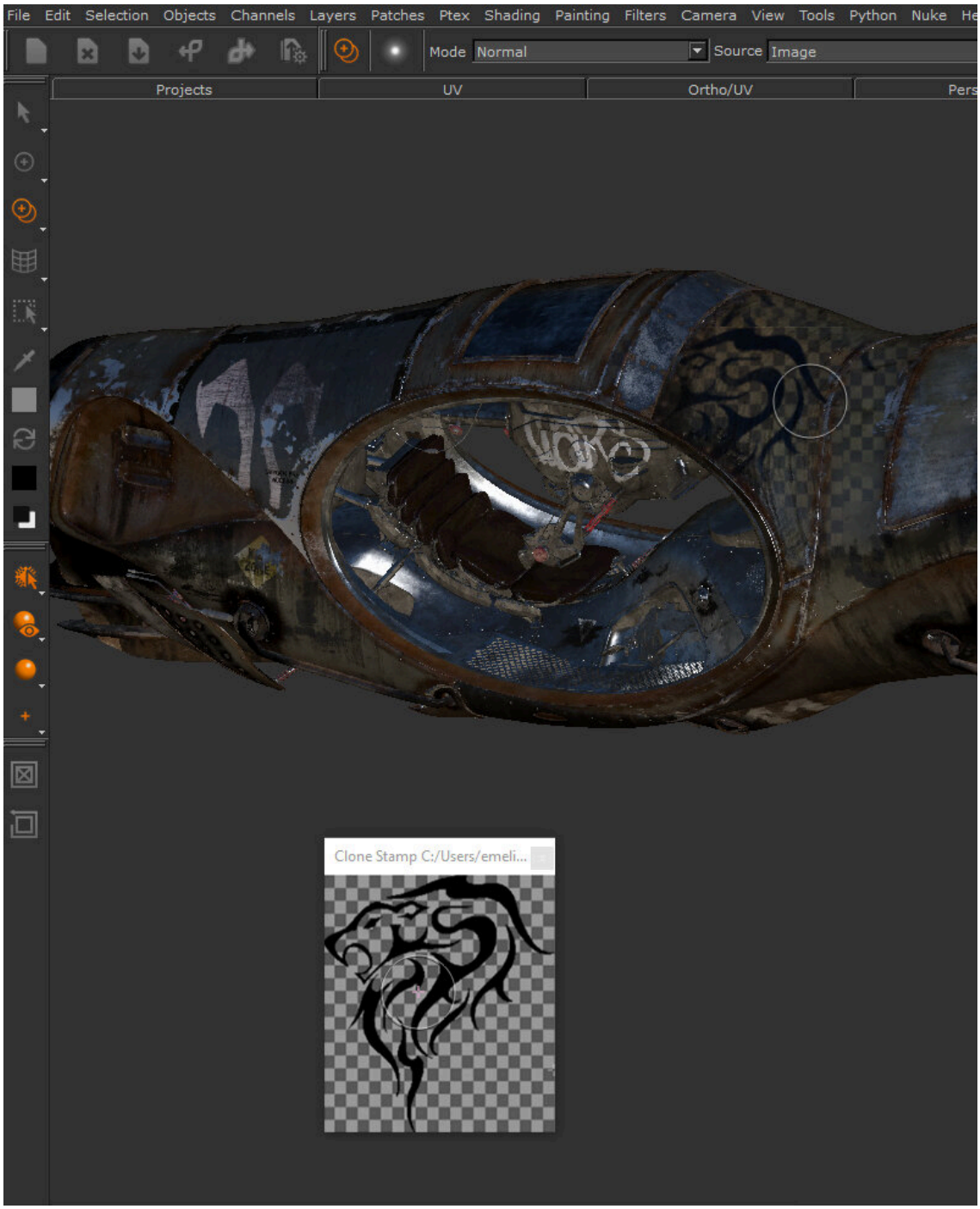
   This removes any unbaked painting on the model. In this case, the model re-displays as imported.
2. In the Tools toolbar, select the **CloneStamp** tool .
3. Drag the image from the **ImageManager** and drop it onto the canvas.

   The image displays in a separate window.
4. Re-size and position the image window alongside your model. To scale the image within the window, hold down **Alt**, right-click, and drag. To pan the image, hold down **Alt**, left-click, and drag.

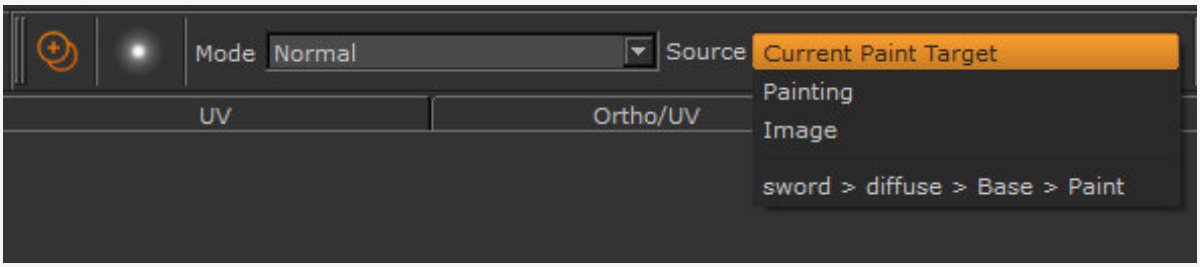


5. To select the source point for cloning on the image, press **Ctrl**/**Cmd** and click.

   Once you start painting, a magenta + displays the source point on the image.
6. Once you've selected the source, you can paint. The cursor moves on the image in sync with the cursor on the model as it clone stamps from the image onto the model.


File Edit Selection Objects Channels Layers Patches Ptex Shading Painting Filters Camera View Tools Python Nuke He
Mode Normal
Source Image
Projects
UV
Ortho/UV
Pers
Clone Stamp C:/Users/emeli...

Tip: You can also clone stamp from existing textures - painted on the surface of the object, in the paint buffer, or painted in another channel. To select a clone source, use the **Source** menu in the **Project** toolbar and select from the origins available.



Experiment: Change the view (as covered in Setting the View), and try both painting through and clone stamping the image onto another side of the object. Do you like one method of using images to paint more than the other?

Where Do You Go from Here?

This lesson covered the two main options for painting images onto a model:

- dragging an image and positioning it on top of an object, then **Paintingthrough** the image onto the surface of the object, and
- dragging an image and positioning it on the canvas next to an object, then **Clonestamping** from a part of the image to the surface of the object.

The next tutorial, Exporting and Importing, covers procedures for saving and re-importing your work.

Exporting and Importing

This lesson explains how you can **export** channels or layers you've painted in Mari as a set of image files, then re-**import** them into your project. This familiarizes you with just a few of the methods for exporting and importing.

Resources Needed

To complete this lesson, you need access to a computer that can run Mari, and the following sample files:

• **1-Blacksmith** - the Mari project file you created in tutorial 1 and used in tutorial 2, tutorial 3, and tutorial 4.

How Long Should It Take?

Plan on spending about 25 minutes to complete this lesson.

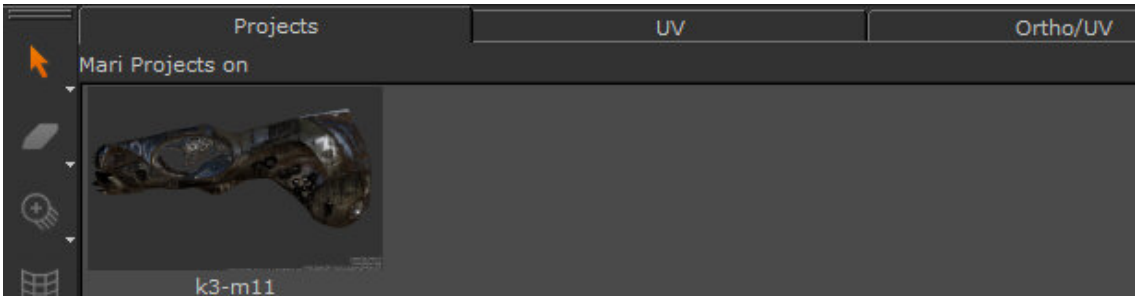
About Mari Exports

Mari includes the facility to **export** the painting in a project to a series of image files. You might use this, for example, to modify part of your work in an external program, and then re-**import** those modifications into the project. Mari lets you select formats and naming conventions for the export. Each export contains the painting baked in a selected channel, one image file per patch. When you import a series of previously exported files, Mari lets you know whether it's a complete set.

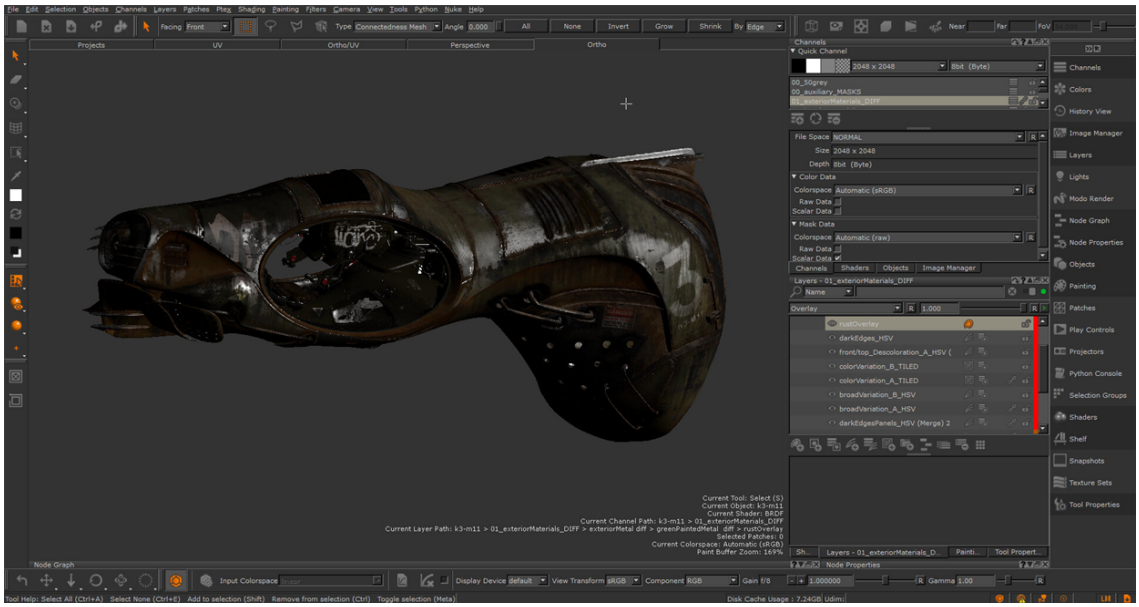
Opening Your Project

1. Start Mari.

   The Mari workspace displays.
2. On the **Projects** tab, double-click on your **1-Blacksmith** project to open it.



Your project opens and switches to the **Ortho** view.

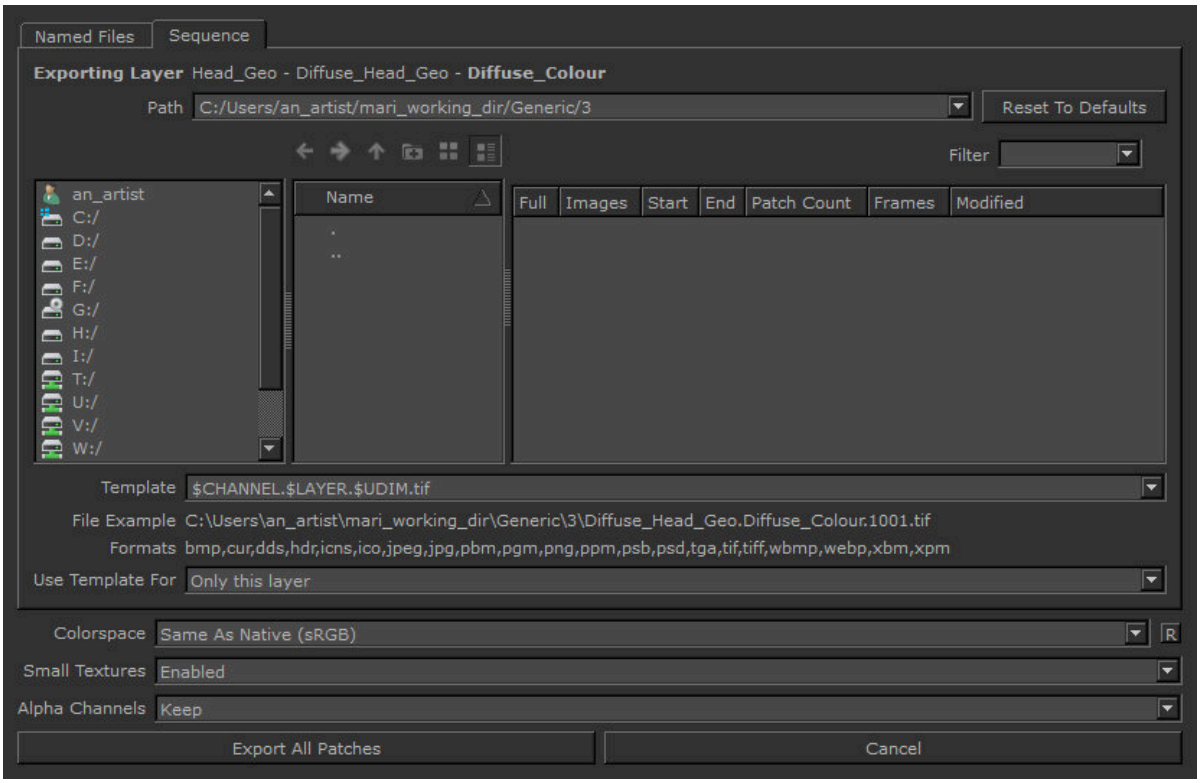


Exporting a Channel

1. To export the current channel, **Diffuse**, from the **Channels** menu, select **Export** > **Export Current Channel**.

   This exports the currently-selected channel, **Diffuse**. To export any channel, right-click the channel you want to export in the **Channels** palette, and select **Export** > **Export Current Channel** from the dropdown menu.

   The **Export** dialog box displays.



Note: The **Named files** tab does not display if your model contains more than 20 patches.

2. If necessary, use the two left-hand navigation boxes to navigate to the folder you want to export to.
3. In the **Template** field, Mari names each image file in the export, one per layer and per patch, based on this template. The default is **$CHANNEL.$LAYER.$UDIM.tif**

Note: Notice the **File Example** below it, which displays what a filename looks like with that template. In the illustration, it's: **diffuse.Base.1001.tif**

So each filename includes the following (with examples from the illustration):
-**$CHANNEL** - the name of the channel you're exporting layers from: **Diffuse**.
-**$LAYER** - the name of the layers you're exporting: Base.
-**$UDIM** - an incremental number identifying each patch in those layers: **1001** (followed by **1002**, **1003**, and so on).
-**tif** - the extension for the selected file format. You can set the format for the exported files by changing this extension. Mari supports standard image formats (such as **.png**, **.jpg**, **.tga**, and **.dds**).

Tip: The formats available to export depend on the color depth of the channel. Channels with color set to Half-Float or Float can only be exported to **.psd**, and **.psb** format. On Windows, channels with color set to Half-Float or Float also can be exported to **.dds** format.

4. Set the **Use Template For** field to **Only this layer** in order to only use the template for the layer displayed at the top of the dialog.
5. Set the **Colorspace** in which you want to export the data. By default, Mari selects **Automatic**, which is determined by the file name, size, and type of data in the channel. For this tutorial, keep the default setting.
6. Enable **Small Textures** to export any patches that you haven't painted yet, or have painted as a single color, as 8x8-pixel textures at the correct size.
7. Set **Alpha Channels** to **Keep** so that alpha channels aren't removed when exporting channels.
8. Click **Export All Patches**.

   This exports all patches. You may be notified that the selected directory does not exist, and prompted about whether you want to create the directory.

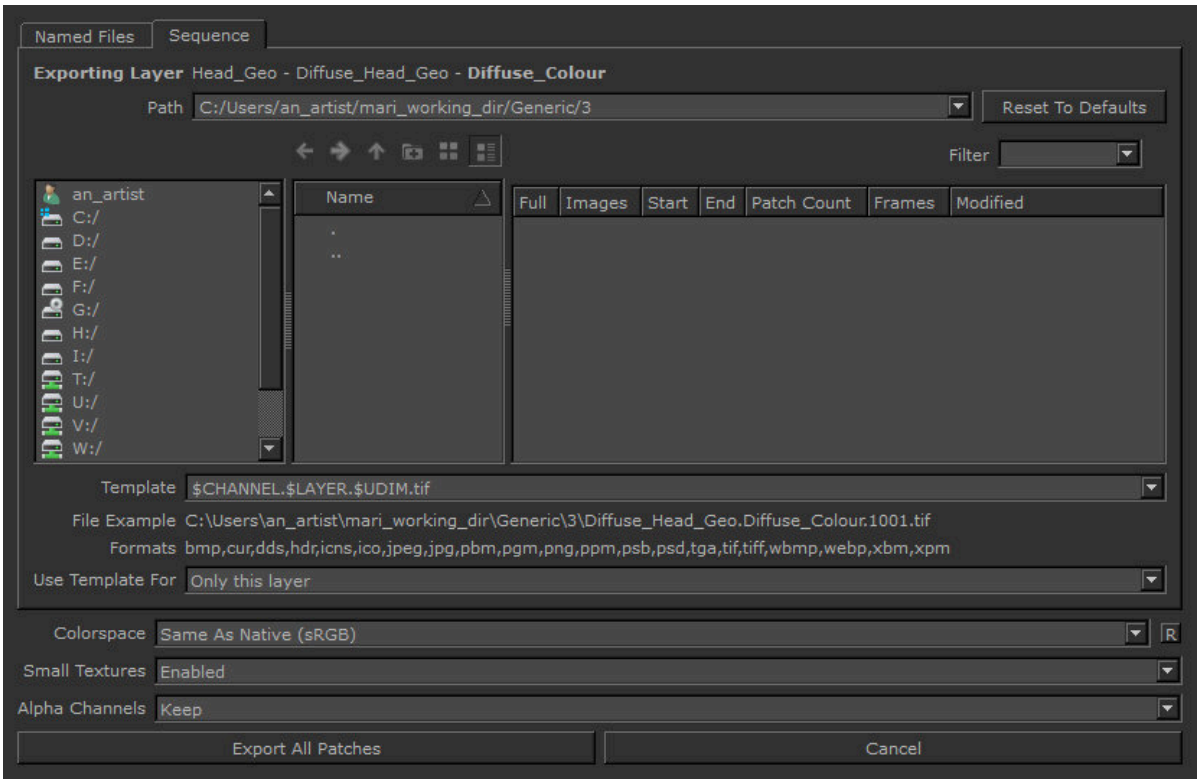
Tip: For a complete explanation of export fields, see the *Mari Reference Guide* and the *Exporting and Importing Textures* chapter in the *Mari User Guide*.

Note: As the options above only export channels one dialog at a time, to export multiple channels at once, see the Export Manager Dialog.

Tip: The Export Manager Dialog also has additional file options for EXR files. See Exporting EXR Files for Render.

1. To export selected layers, from the **Layers** menu, select **Export** > **Export Selected Layers**. This exports any selected layers in the layer stack (in this case, **Base**).

   The **Export** dialog box displays.



Note: The **Named files** tab does not display if your model contains more than 20 patches.

2. If necessary, use the two left-hand navigation boxes to navigate to your tutorial folder.
3. In the **Template** field, Mari names each image file in the export, one per channel, layer and patch, based on this template. The default is **$CHANNEL.$LAYER.$UDIM.tif**.

Note: Notice the **File Example** below it, which displays what a filename looks like with that template. In the illustration, it's: **diffuse.Base.1001.tif**

So each filename includes the following (with examples from the illustration):
-**$CHANNEL** - the name of the channel you're exporting layers from: **Diffuse**.
-**$LAYER** - the name of the layers you're exporting: Base.
-**$UDIM** - an incremental number identifying each patch in those layers: **1001** (followed by **1002**, **1003**, and so on).
-**tif** - the extension for the selected file format. You can set the format for the exported files by changing this extension. Mari supports standard image formats (such as **.png**, **.jpg**, **.tga**, and **.dds**).

> **Tip:** The formats available to export depend on the color depth of the channel. Channels with color set to Half-Float or Float can only be exported to **.psd**, and **.psb** format. On Windows, channels with color set to Half-Float or Float also can be exported to **.dds** format.

4. Set the **Use Template For** field to **Only this layer** in order to use the template for only the layers selected. If **Only this layer** is selected when attempting to export, Mari asks you for each individual layer whether you want to use the template.
5. Set the **Colorspace** in which you want to export the data. By default, Mari selects **Automatic**, which is determined by the file name, size, and type of data in the channel. For this tutorial, keep the default setting.
6. Enable **Small Textures** to export any patches that you haven't painted yet, or have painted as a single color, as 8x8-pixel textures at the correct size.
7. Set **Alpha Channels** to **Keep** so that alpha channels aren't removed when exporting channels.
8. Click **Export All Patches**.

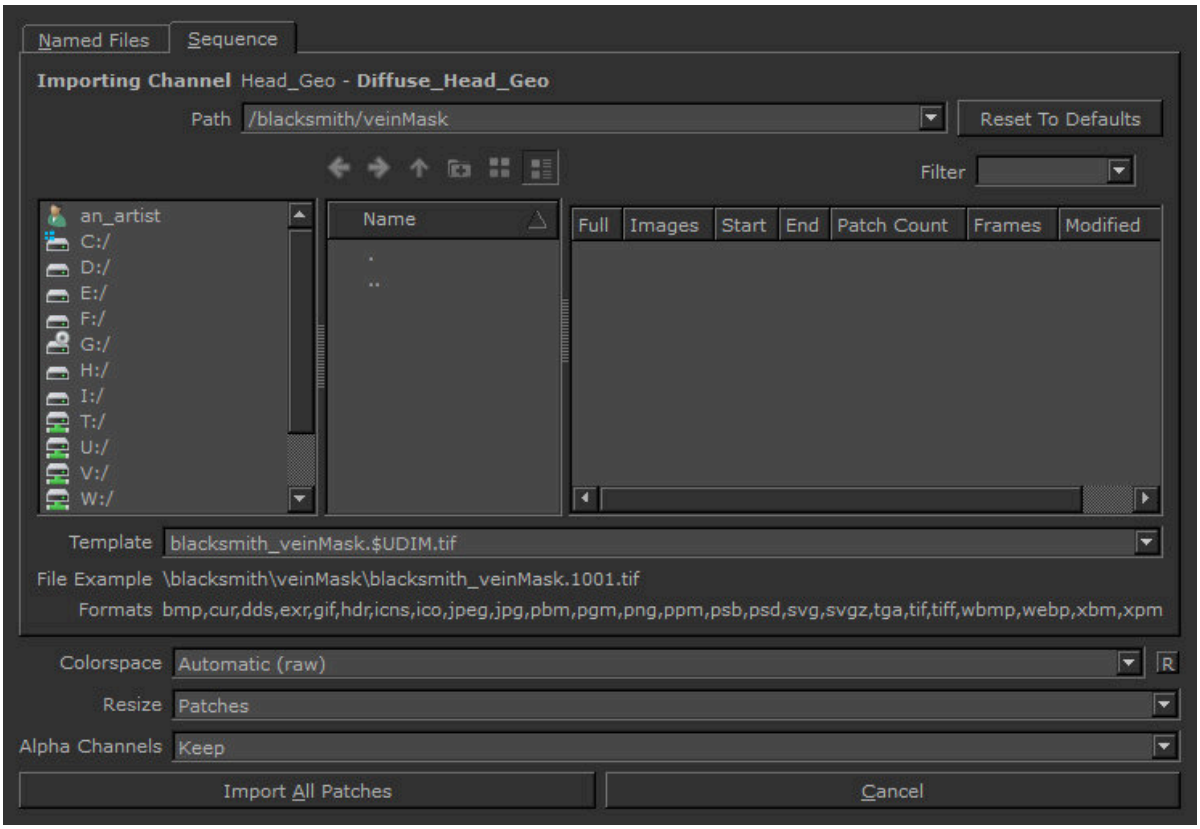
   This exports all patches.

> **Tip:** For a complete explanation of export fields, see the *Mari Reference Guide* and the *Exporting and Importing Textures* chapter in the *Mari User Guide*.

Importing a Channel

1. To import a channel (for example if you've tweaked some exported image files in another application and now want them back in your texture):
   - from the **Channels** menu, select **Import** > **Import into Current Channel** for the currently selected channel, or
   - right-click on the channel where you want to import the textures (in this case, **Color**) in the **Channels** palette, and select **Import** > **Import into Current Channel** from the dropdown menu.

   The **Import** dialog box displays.



Note: The **Named files** tab does not display if your model contains more than 20 patches.

2. Select the path for the imported images, by either:
   - typing it in the **Path** field at the top of the dialog,
   - clicking on it in your directory bookmarks (to the left), or
   - browsing to it in the middle panes.

   If you are using the middle panes to browse to the directory, you can use the buttons above the view to navigate back, forward, up one directory level, create a new directory, or switch between icon-only and full-details directory views.
3. Select the texture set to import.

   Mari shows all the image sets available under the selected directory, arranged by sub-directory.

   For each texture set, the **Import** dialog box includes:
   - **Full** - shows a green bar if the selected image sequence has a full set of images for the current model, a yellow bar if the selected image sequence has a partial set of images for the current model, or a red bar if the selected image sequence has no images for the current model.
   - **Images** - the name of each image to import, with the UDIM number represented as @ (in this case, **diffuse.@.tif**) and frame number (if present) represented as #.
   - **Start** and **End** - the first and last UDIM numbers in the image set (in this case, **1001** to **1055**).

- **Patch Count** - the number of images in the set (in this case, **50**).

4. In the **Template** field, Mari names each image file in the import, one per layer and patch, based on this template. The default is **$CHANNEL.$LAYER.$UDIM.tif**.

> **Note:** Notice the **File Example** below the **Template** field, which displays what a filename looks like with that template. In the illustration, it's: **diffuse.Base.@.tif**
>
> So each filename includes the following (with examples from the illustration):
> -**$CHANNEL** - the name of the channel you're importing: **Diffuse**.
> -**$LAYER** - the name of the layers you're importing: Base.
> -**$UDIM** - an incremental number identifying each patch in those layers: **1001** (followed by **1002**, **1003**, and so on).
> -**tif** - the extension for the selected file format. You can set the format for the exported files by changing this extension. Mari supports standard image formats (such as **.tif**, **.exr**, **.png**, **.jpg**, **.tga**, and **.dds**).

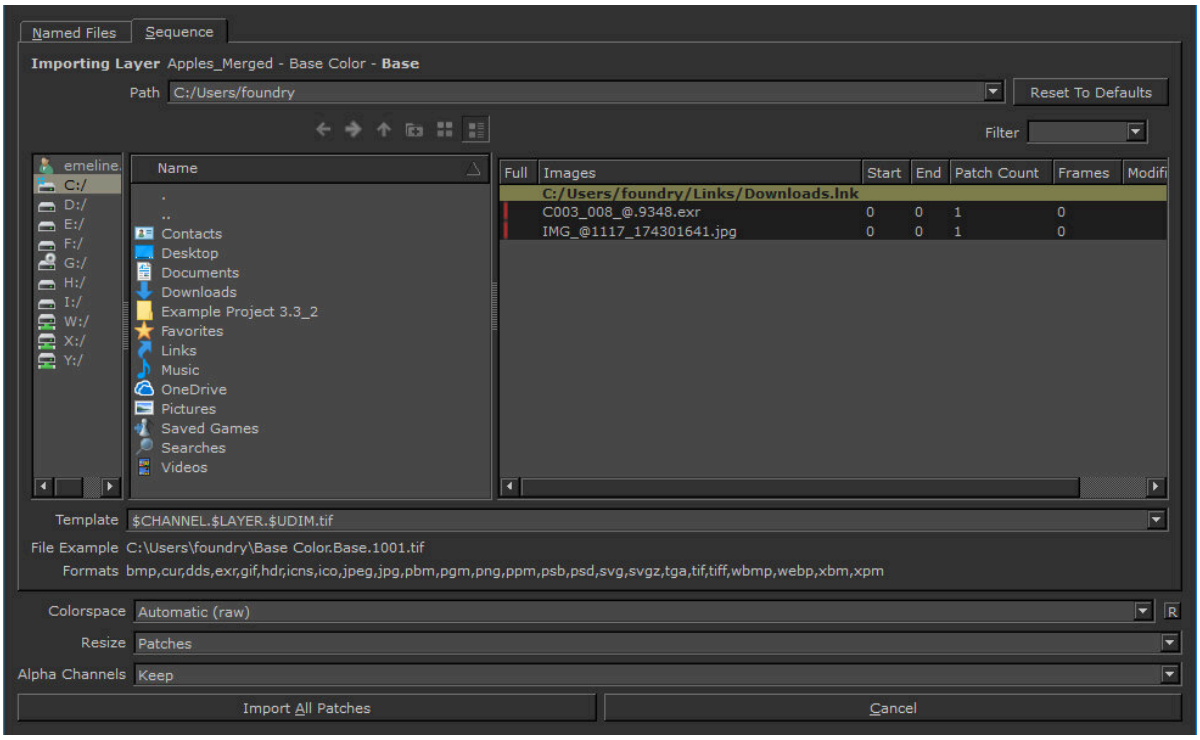
5. Set the **Colorspace** in which you want to import the data. By default, Mari selects **Automatic**, which is determined by the file name, size, and type of data in the channel. For this tutorial, keep the default setting.
6. If there are size mismatches between the patch and image size, set the **Resize** field to resize the **Images** to match the patches, or to resize the **Patches** to match the images.
7. Set **Alpha Channels** to **Keep** so that alpha channels aren't removed when exporting channels.
8. Click the **Import All Patches** to import the files into your project.

   Mari imports the textures from the selected files.

Importing a Layer

1. To import textures into a layer, from the **Layers** menu, select **Import** > **Import into Current Layer.** This imports all selected textures into the current, selected layer (in this case, **Base**).

   The **Import** dialog box displays.



Note: The **Named files** tab does not display if your model contains more than 20 patches.

2. Select the path for the imported images, by either:
   - typing it in the **Path** field at the top of the dialog,
   - clicking on it in your directory bookmarks (to the left), or
   - browsing to it in the middle panes.

   If you are using the middle panes to browse to the directory, you can use the buttons above the view to navigate back, forward, up one directory level, create a new directory, or switch between icon-only and full-details directory views.
3. Select the texture set to import.

   Mari shows all the image sets available under the selected directory, arranged by sub-directory.

   For each texture set, the **Import** dialog box includes:
   - **Full** - shows a green bar if the selected image sequence has a full set of images for the current model, a yellow bar if the selected image sequence has a partial set of images for the current model, or a red bar if the selected image sequence has no images for the current model.
   - **Images** - the name of each image to import, with the UDIM number represented as @ (in this case, **diffuse.@.tif**) and frame number (if present) represented as #.
   - **Start** and **End** - the first and last UDIM numbers in the image set (in this case, **1001** to **1055**).
   - **Patch Count** - the number of images in the set (in this case, **50**).

4. In the **Template** field, Mari names each image file in the import, one per channel, layer and patch, based on this template. The default is **$CHANNEL.$LAYER.$UDIM.tif**.

> **Note:** Notice the **FileExample** below the **Template** field, which displays what a filename looks like with that template. In the illustration, it's: **diffuse.Base.@.tif**
>
> So each filename includes the following (with examples from the illustration):
> -**$CHANNEL** - the name of the channel you're importing: **Diffuse**.
> -**$LAYER** - the name of the layers you're importing: Base.
> -**$UDIM** - an incremental number identifying each patch in those layers: **1001** (followed by **1002**, **1003**, and so on).
> -**tif** - the extension for the selected file format. You can set the format for the exported files by changing this extension. Mari supports standard image formats (such as **.tif**, **.exr**, **.png**, **.jpg**, **.tga**, and **.dds**).

5. Set the **Colorspace** in which you want to import the data. By default, Mari selects the colorspace determined by the value set in **Default Import Colorspace** in the **Mari Preferences** dialog under the **Color** tab. See Mari Preferences Dialog for more information.
6. If there are size mismatches between the patch and image size, set the **Resize** field to resize the **Images** to match the patches, or to resize the **Patches** to match the images.
7. Set **Alpha Channels** to **Keep** so that alpha channels aren't removed when exporting channels.
8. Click **Import All Patches**.

   This imports all patches.

> **Tip:** For a complete explanation of export fields, see the *Mari Reference Guide* and *Exporting and Importing Textures* chapter in *Mari User Guide*.

Where Do You Go from Here?

This lesson showed you how to:

- **export** a current channel or selected layer in a Mari project to a series of image files.
- **import** textures into a channel's layer stack or into a selected layer, for example after having manipulated the exported textures in another program.

Congratulations! You have now completed all modules in the Mari Tutorials. To better familiarize yourself with the features you are particularly interested in or to get answers to specific problems that arise during painting, please refer to the accompanying *Mari User Guide* and *Mari Reference Guide*.

Mari Sample Projects

Mari has three sample projects for you to explore different features and practice texturing tools.

Patches Project

Explore Multi-Paint with the Patches ShaderCat project.

Material Design Project

Explore procedural texturing with the Material Design project.

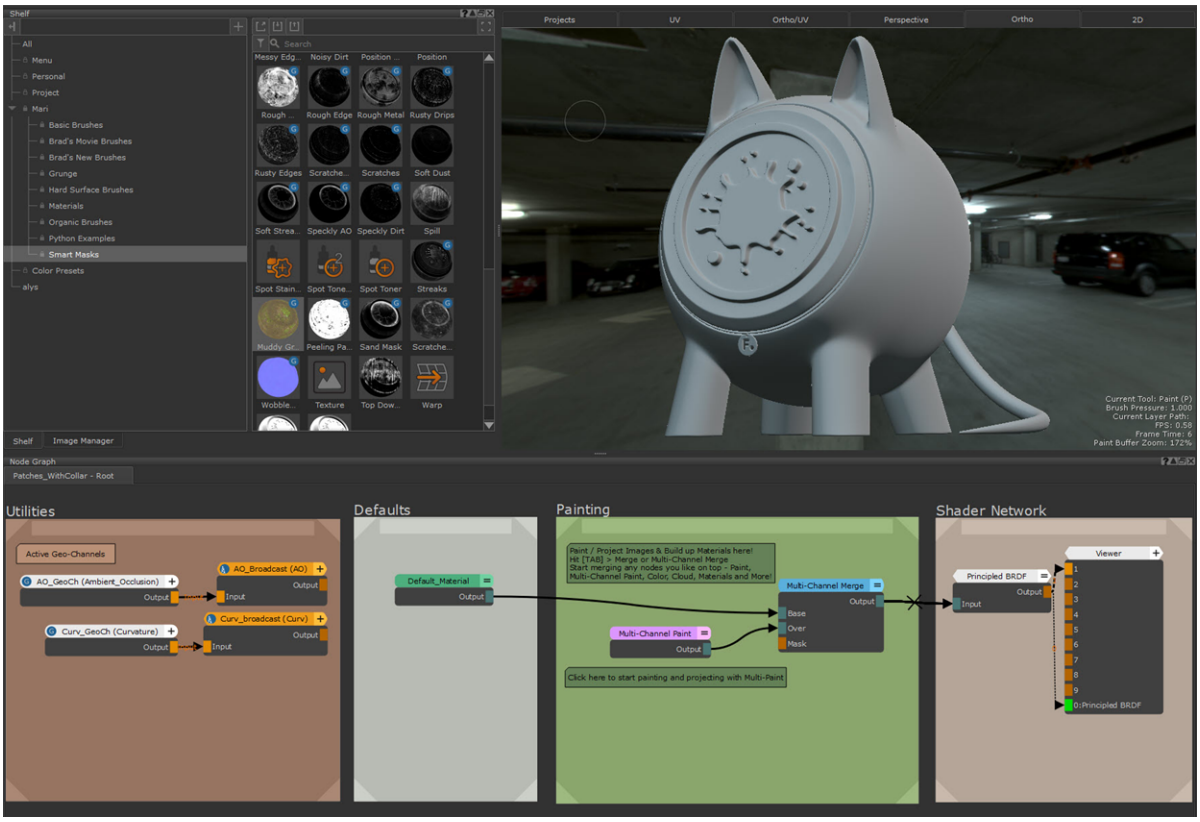
Example Project

Explore texturing workflows with the Mari Example project.

Example Project: Patches

Where to access: **Help > Example Projects > Create Patches ShaderCat Project**

The Patches project is designed to get you started on Multi-Paint in Mari.



Walk-through

This sample project is split into three sections:

- Utilities
- Defaults
- Painting
- Shader network

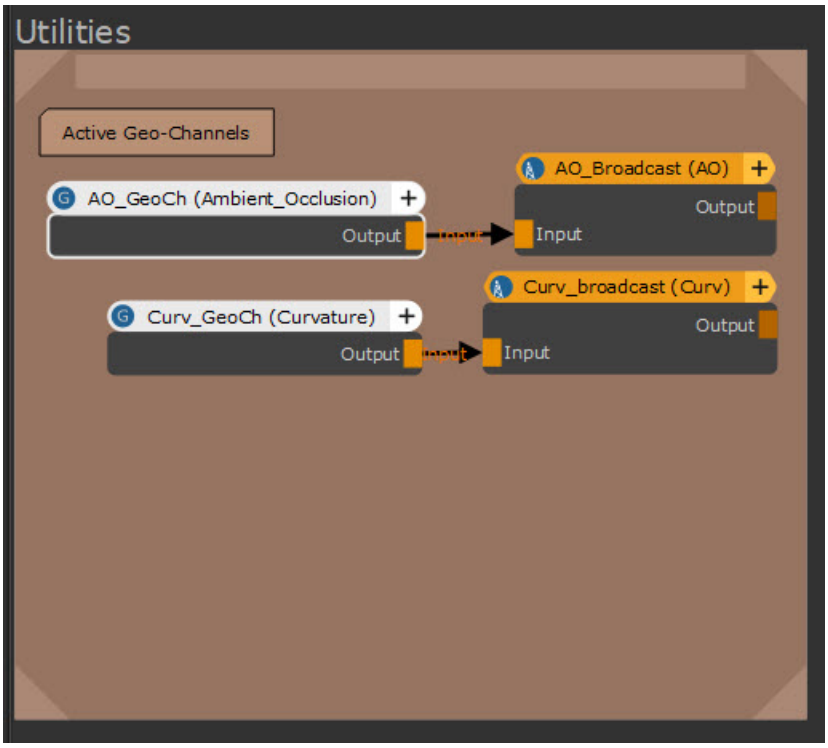
Utilities

The Patches project has Ambient Occlusion and Curvature Geo-Channels active. This means that the model's Ambient Occlusion and Curvature has been calculated for the model. If you wanted to add wear-and-tear or other photorealistic effects to the model, these effects correspond to the model's geometry.

> **Note:** Learn more about Geo-Channels go to Adding Geo-Channels to Objects and Geo-Channel Node.

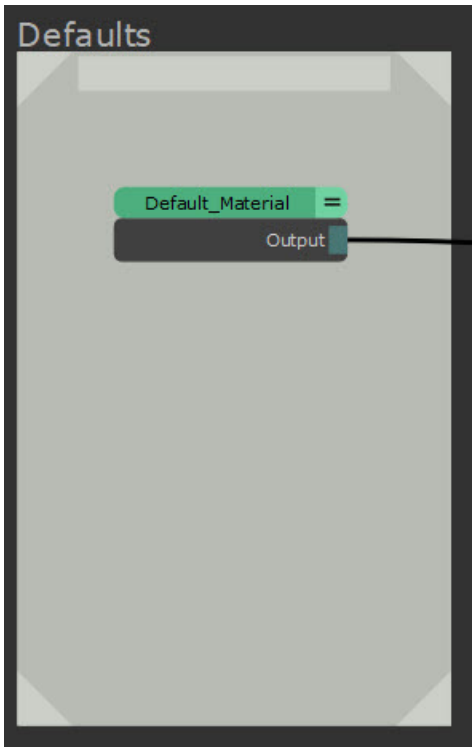
The Geo-Channel nodes are attached to Teleport Broadcaster nodes - you can set up Teleport Receiver nodes to apply the geo-channel to any materials you choose to add to this project.

Note: Learn more about Broadcast and Receiver nodes at Teleport Broadcaster and Teleport Receiver Nodes.



Defaults

The Default_Material node sets a gray tone as the model's base color. This is a helpful method to prevent a transparent border to appear around your paintwork.

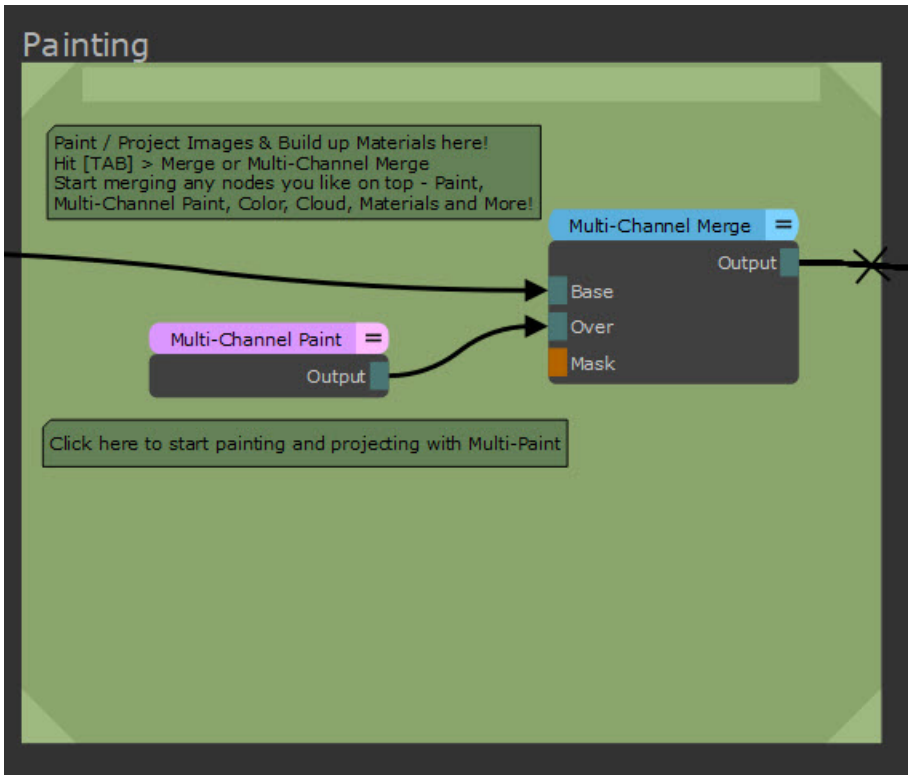


Painting

In the Painting section of the node graph is the **Multi-Channel Paint** node. This node enables you to paint into up to 8 streams simultaneously. For full information on how Multi-Paint works in Mari, visit Multi-Paint.

> **Experiment:** Mari 7.5+ ships with TexturingXYZ texture sets that can be used to test out Multi-Paint. Access them at the Mari downloads page.

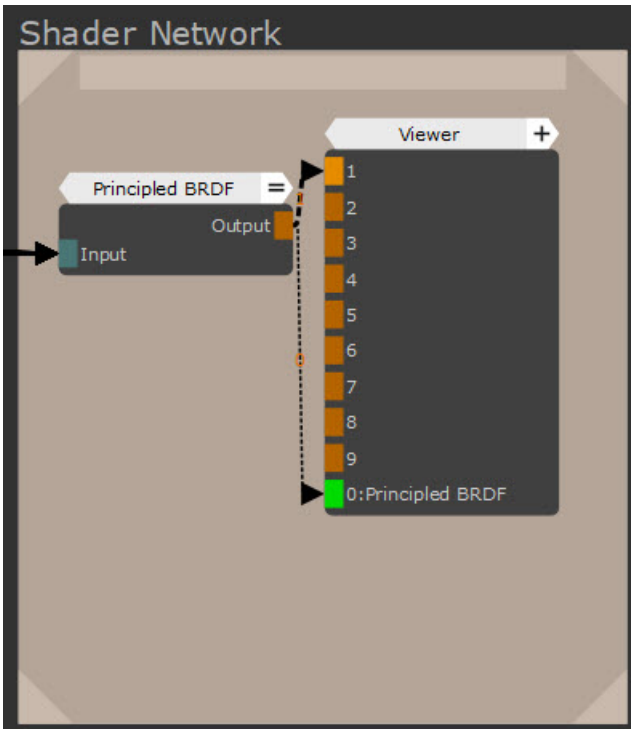
The Multi Channel Merge node enables you to merge two materials together. Use Multi-Channel Merge nodes with Multi-Channel Paint nodes to build up your model with multiple textures.



Shader Network

The output of the Multi-Channel Merge node is then attached to a Principled BRDF Shader node. Shaders control how Mari displays the model on the canvas under certain, user-specified lighting conditions.

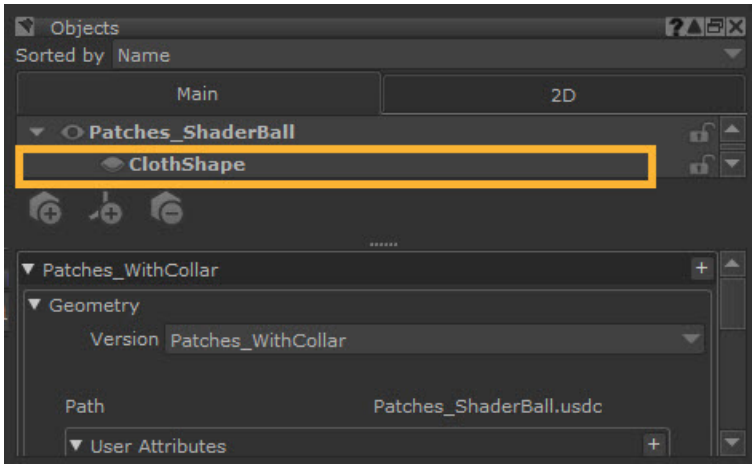
This Shader node is attached to a Viewer (which is created when a project is created). The viewer defines what is visible on the canvas.



Cloth Model

The Patches model also has another object you can use to experiment with Mari. In the Objects panel, find the **ClothShape** object nested in the **Patches_ShaderBall**. Click the eye icon to make this object visible, and the Cloth object will be visible over the Patches model.

This object is useful when creating fabric textures in order to see how they look in a draping pattern.



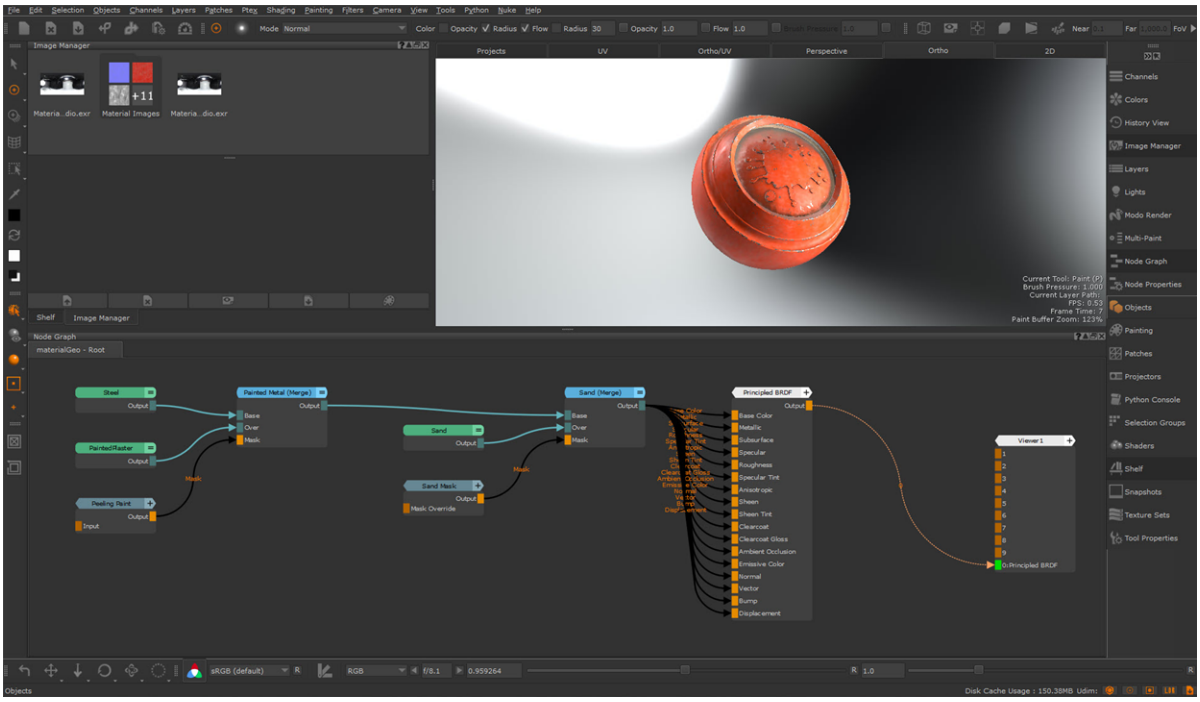
Where to Explore Next

- Multi-Channel Paint Node Workflow Example
- Multi-Channel Merge Node

Sample Project: Material Design

Where to access: **Help > Example Projects > Create Example Material Design Project**.

The **MaterialDesign** project is based around procedural workflows. It uses materials and smart masks to achieve the textures.

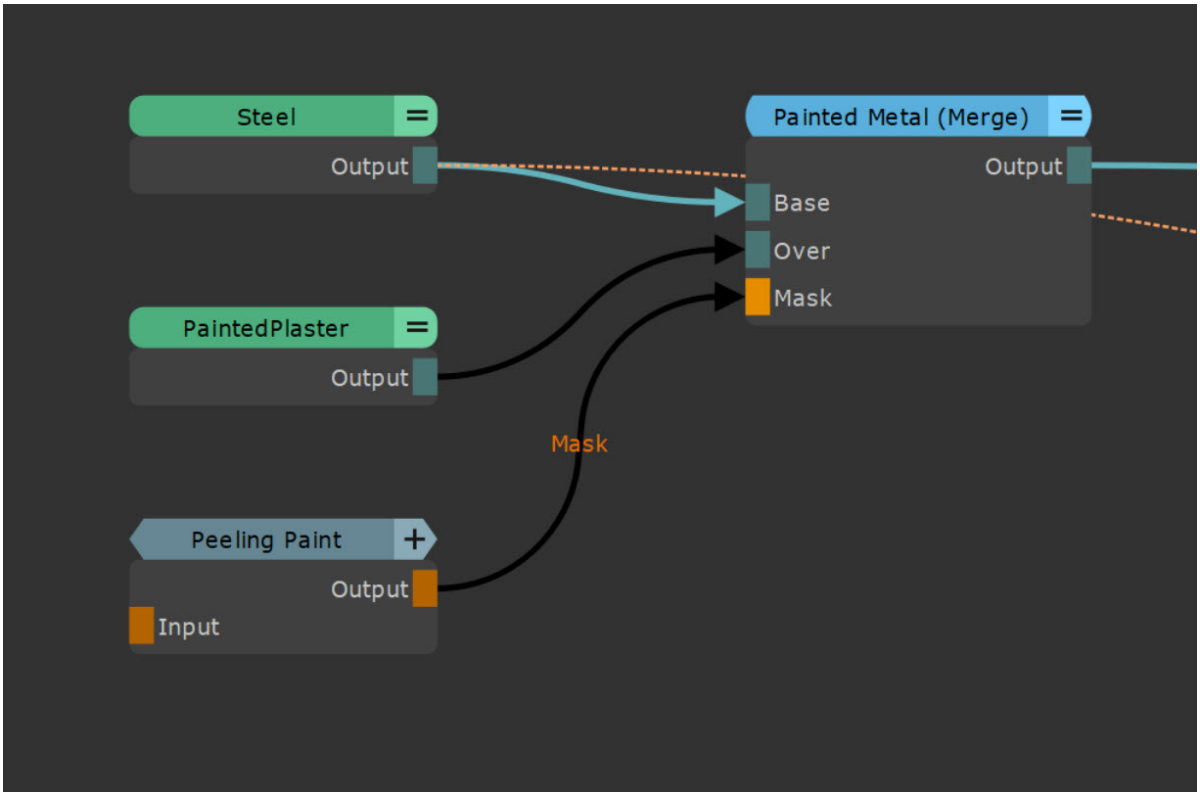


Walk-through

This sample project is split into three sections:

- Creating Painted Metal
- Adding a Sand mask
- Shader network

Creating Painted Metal



The first part of this project sets up the base materials that the object is made out of. In this example, the object is made of steel that is painted.

This is completed using the following steps:

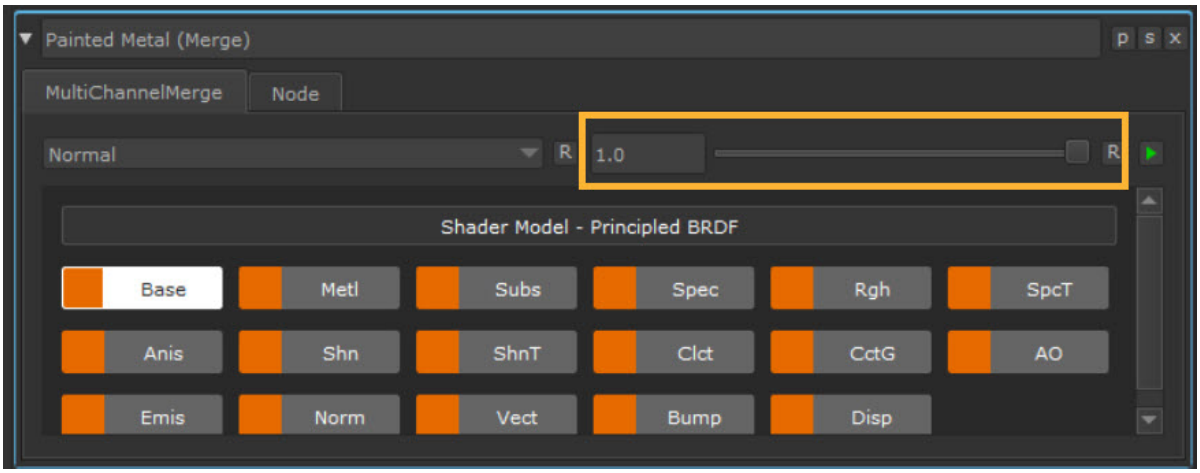
- The **Painted Metal (Merge)** node is aMulti-Channel Merge Node, which enables you to merge two materials together.
- The **Steel** material (from theWorking with Materials in the Node Graphshelf) is plugged into the **Base** input of the **Painted Metal (Merge)** node.

> **Tip:** Select the **Steel** material node and press **1** to view what the base steel material looks like without the paint texture on top of it.

- **PaintedPlaster** is another material from the Materials shelf. This goes on the **Over** input of the **Painted Metal (Merge)** node, which places it on top of the **Base** input.

Without applying a mask to the Merge node, the material in the Base input will not be visible. This is because the Paint is an opaque layer that covers the steel, so the opacity for each of the channels in the multi-channel merge node is set to 1.

A view of the Painted Metal (merge) node before applying a mask. The Base material is not visible.



The Multi-Channel Merge node properties, displaying the Base Channel's opacity value as 1. You can click through each channel to see each of their settings.

- To create the peeling paint effect that is realistic, the next step applied in this sample project is the **Peeling Paint** mask.

- A mask is a black and white image that defines where the **Base** material (**Steel**) will be visible and where the **Over** material (**PaintedPlaster**) material is visible. This effect is applied with a Smart Mask Node.
- This project uses the preset **Peeling Paint** mask that can be found at **Shelf > Mari > Smart Masks**.

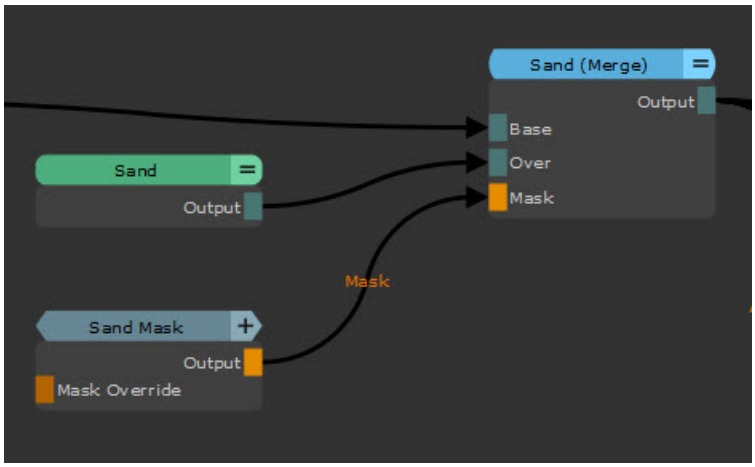
The **Peeling Paint** mask applied to the **Painted Metal (Merge)** node

Experiment: Clicking on the **Peeling Paint** node will open up the Node Properties panel with the node's parameters. From here you can adjust the pattern and Curvature properties to alter the look of the paint.

Tip: Watch this video - Behind the scenes: Smart Masks in Mari 7.1 - to understand more about Smart Mask structure.

Experiment: Open up the Peeling Paint mask using Ctrl+LMB or Ctrl+Enter to view the structure of the Smart Mask.

Now that the effect of a painted steel object has been completed, another texture is applied on top of it - sand that is gathering in the creases of the object and settling on top of it.



Now that the effect of a painted steel object has been completed, another texture is applied on top of it - sand that is gathering in the creases of the object and settling on top of it.

This effect is created with the following steps:

- A new **Multi-Channel Merge** node - labelled **Sand (Merge)** - is created and the **Painted Metal (Merge)** node is plugged into the Base input. This means that another texture can be applied on top of the peeled paint effect.
- The **Sand** material is placed in the **Over** channel. This material can be accessed from **Shelf > Mari > Materials**.
- The **Sand Mask** node is plugged into the **Mask** input, defining how the sand material will appear on the object. This smart mask is another preset found at **Shelf > Mari > Smart Masks**. In the Node Properties panel of the mask, you can edit the look of the sand by adjusting the **Ambient Occlusion**, **Topdown** and **Edge Pattern** parameter values.

Shader Network

The output of the Sand Merge node is then attached to a Principled BRDF Shader node. Shaders control how Mari displays the model on the canvas under certain, user-specified lighting conditions. This shader node is attached to a Viewer (which is created when a project is created). The viewer defines what is visible on the canvas.

Where to go from here

Check out the other Sample projects available in Mari:

- Patches Sample Project
- Example Project

Sample Project: Example

Where to access: **Help** > **Example Projects** > **Create Example Project**.

The ExampleProject showcases texturing by painting in Mari. It is made out of two objects, **Head_Geo** and **Eyeballs**.

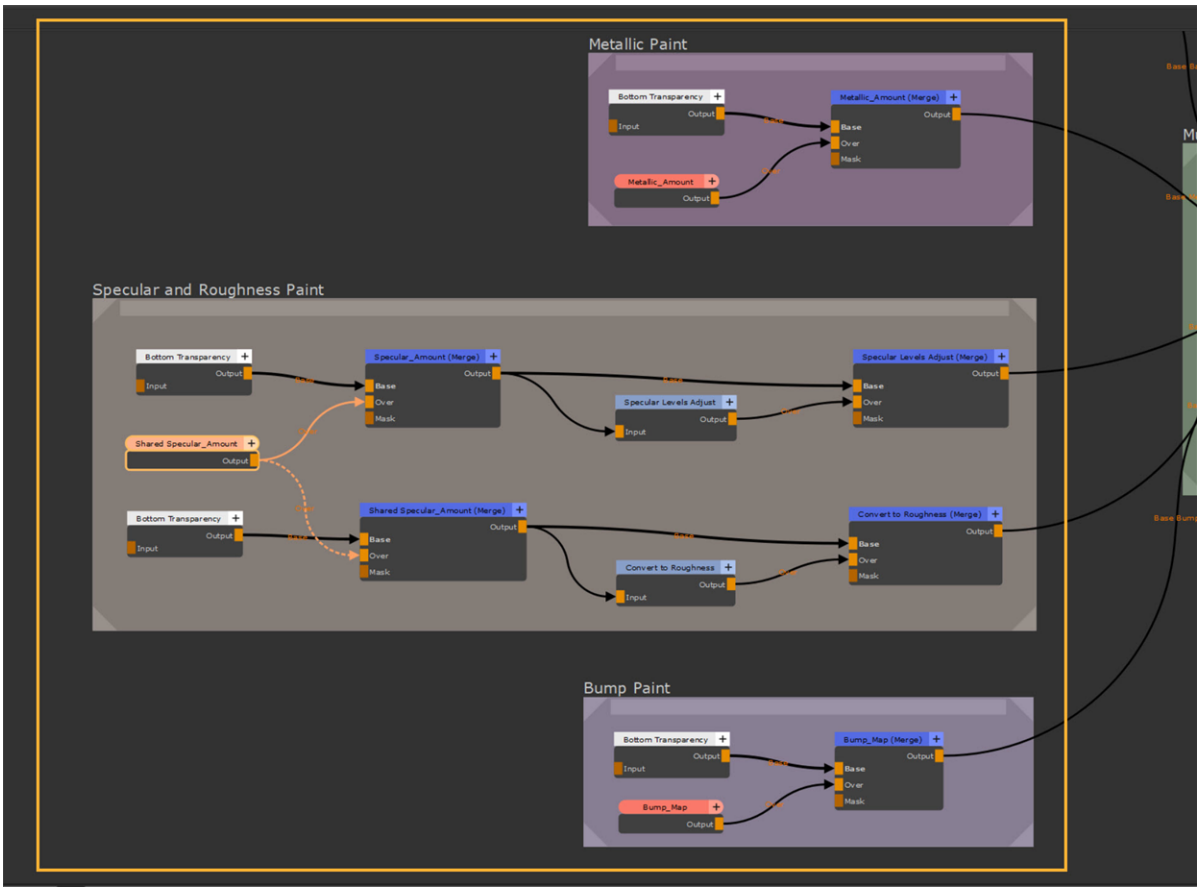
This project is created using an image set that has been imported into the Base Color Paint node. You can view the image UV set in the UV tab.

You can adjust the levels of each of the channels that have been set up to adjust the look of the model.

Head_Geo

In this part of the graph, the different channels are set up. Each channel has a Bottom Transparency Node. The Bottom Transparency node is an empty node with an Alpha channel value of 0 . This node functions to

indicate the bottom of a layer stack. The value of the fill is 0,0,0,0 in the RGBA channels. The node is automatically created if you add **Channels** in the new project creation UI or if you create a new channel inside the object.



This node exists so that you can use the node graph and the layers palette. You don't need a Bottom Transparency node if you only use the node graph workflow.

Base Color Paint

The Bottom transparency node provides the empty base for the Base Color. The **Diffuse_Colour** node is a Paint node that enables you to add color to the model. For example, you could use this node to paint freckles onto the face of the model by changing the colour of the node and painting onto the model.

To add freckles to the model:

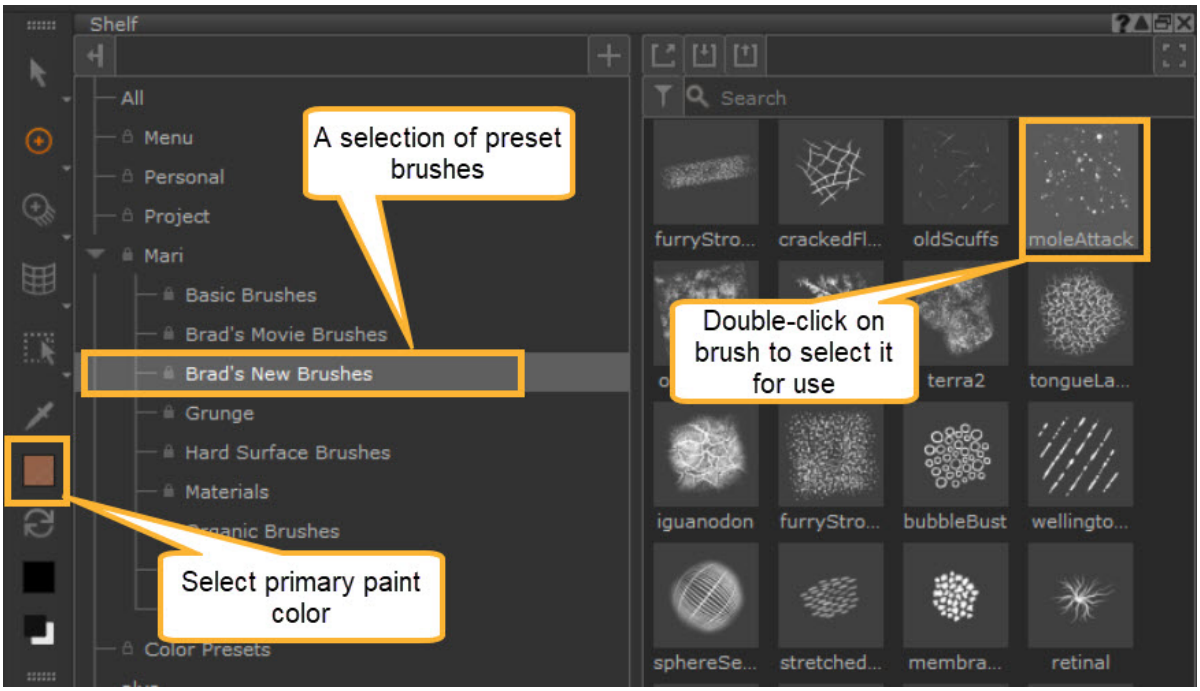
1. In the **Tools** toolbar, select the Painting tool.
2. Select the **Diffuse_Colour** node. This will make the node your active painting target.
3. Choose the primary paint color in the **Tools** toolbar.

Tip: You can also select a secondary paint color and use the swap icon to switch between them easily.

4. Adjust the brush settings. You can use the default brush settings, edit them or choose a preset brush to paint with. To create a freckles effect, you can use one of the preset brushes available in Mari. **Shelf > Mari > Brad's New Brushes > moleAttack** can be used for painting freckles and moles on skin.

Experiment: Mari has over 90 brushes for you to use on your projects.

Tip: You can also edit the brush properties, such as Opacity and Radius, at the Tools Property tab to fully customize your brush for your needs.



Tip: Select the **Diffuse_Colour (Merge)** node and press 1 on your keyboard to view the output of the **Diffuse_Colour (Merge)** node. This will display the output of this node on the model, which can be helpful when working on a specific channel of the project as you can easily see what you are adjusting.

Metallic Paint

The Bottom transparency node provides the empty base for the Metallic texture. The **Metallic_Amount** node is a Paint node that determines whether a metallic texture is applied to the model.

If you view this channel in the default settings, the metallic input is black. This means that there is no metallic texture on the model.

To adjust this, select the Paintbrush tool and click on the **Metallic_Amount** node to make it the primary painting target. Make your paint input color white, and paint on the model.

Change your display to the Shader to see the metallic effect on the model.

Specular and Roughness Paint

The image set that is used for this model has an image that contains both the Specular and Roughness information. These are in the **Shared Specular_Amount** node.

> **Note:** The node is called **Shared Specular_Amount** due to the Sharing Layers workflow. This workflow is important when using the Layers palette but isn't relevant for the node graph, as in the node graph you can connect a paint node to two different outputs (as displayed in this project).

This node is split into two Merge nodes (**Specular_Amount** and **Shared Specular_Amount**).

- **Specular_Amount** output gets plugged into **Specular Levels Adjust (Merge)** node as the base, and the **Specular Levels Adjust** Levels Node which adjusts the value of the **Specular_Amount**.

The output of the Specular Levels Adjust (Merge) node, displaying where the specular information is most prominent (lighter areas).

- **Shared Specular_Amount** output gets plugged into **Convert to Roughness (Merge)** node and **Convert to Roughness** Levels node. In this Levels node, the output levels are adjusted to define the Roughness information for the model.

> **Experiment:** Alter the **Levels > Output Level** values in the **Convert to Roughness** node to see how it changes the final output image.

Bump Paint

The Bump Paint section of the node graph has the Bump_Map paint node that projects the pre-defined bump map onto the model.

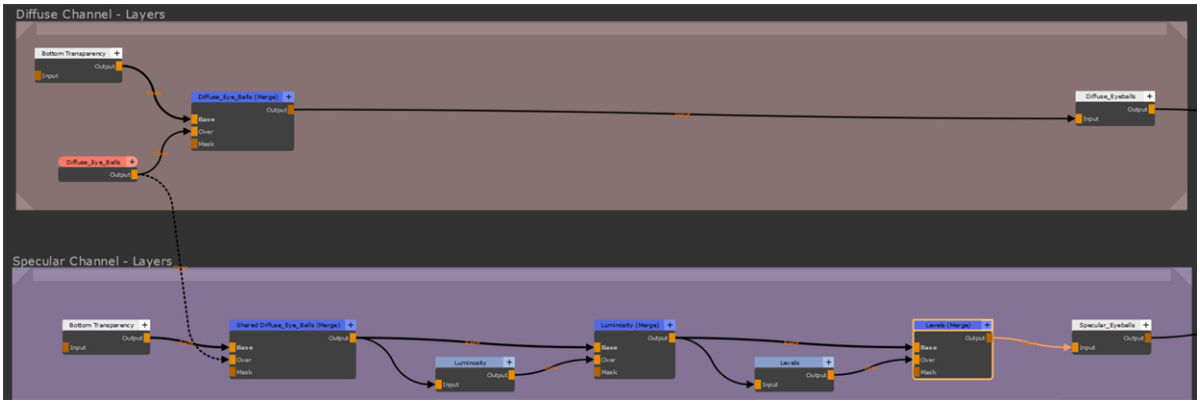
Mud

All of the channels then feed into the **Base** inputs of the **Mud (Merge)** node. This is a Multi-Channel Merge Node, enabling you to merge multiple materials together.

Eyeballs

The Eyeballs object has its own node graph, which can be accessed by going to **Objects** > **Eyes** and double clicking on Eyeballs.

Tip: You can hide the Head_Geo object in order to only see the Eyes object you are working on. In the Objects panel, click the visibility icon ( ) to select which objects are visible in the canvas.



Diffuse Channel - Layers

This part of the Node Graph defines the base color of the eyes.

- The **Bottom Transparency** node provides a transparent base layer to the model.
- The **Shared Diffuse_Eye_Balls** is a Channel Input node that plugs the Eyeball image set into the Over input of the **Diffuse_Eye_Balls (merge)** node.

Specular Channel - Layers

- **Luminosity Node** - a Filter node that defines the amount of luminosity for the eyeballs object.
- **Levels node** - defines the level of luminosity intensity for the object.
- **Specular_Eyeballs** & **Diffuse_eyeballs** nodes - the imported image data for the specular and diffuse channels of the eyeballs object.

Reference Guide

This manual aims to provide a complete reference for all the controls within Mari. It does not give you any instructions on using Mari. For more details see Install Mari.

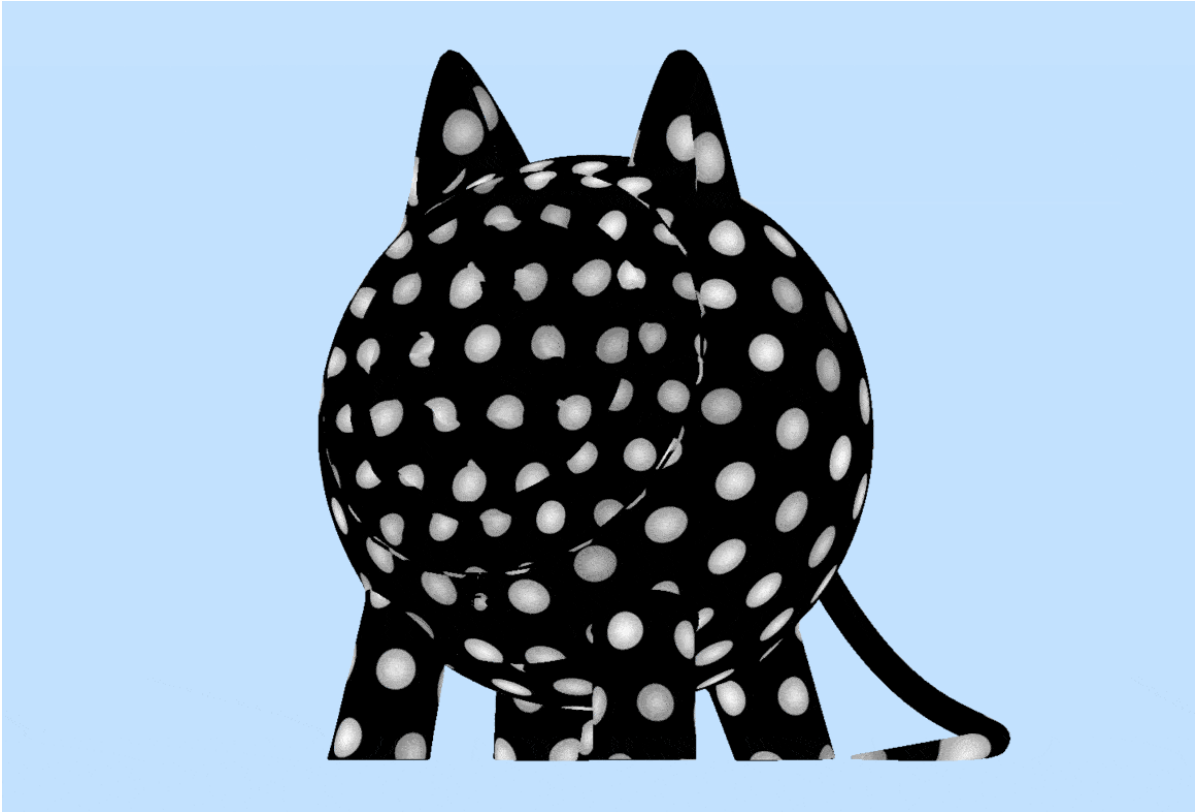
Nodes

This section covers the nodes available to you in Mari, how and why they can be used, and their properties.

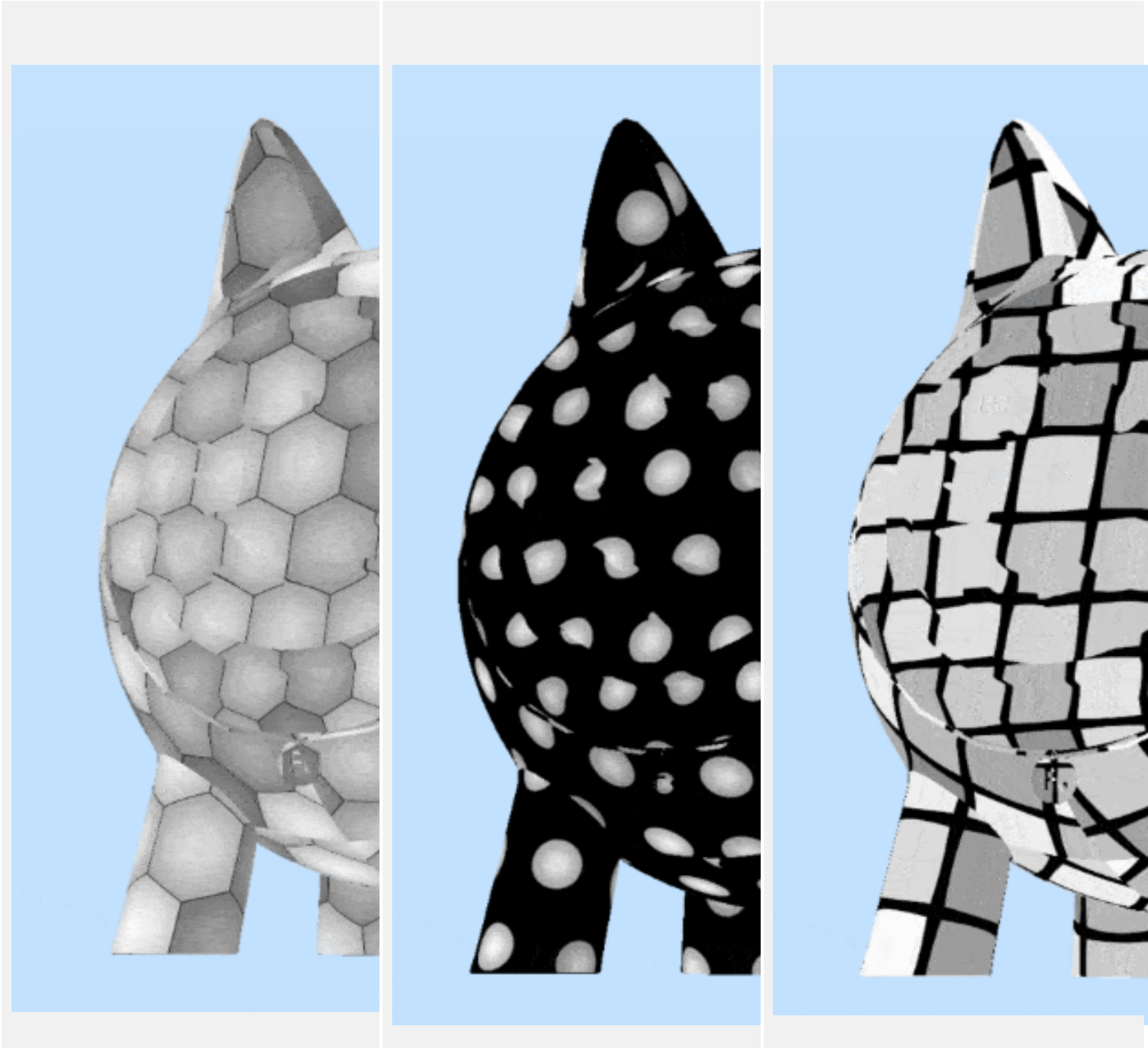
Bubbles Node

The Bubbles node is a new procedural node that allows artists to create a variety of procedural patterns derived from cell or hexagon shapes. You can use random colour values as well as offsetting the image in the X and Y axis.

Use the bubbles node as a mask or in conjunction with other procedural nodes to recreate a number of geometric shapes without relying on flat 2D images.



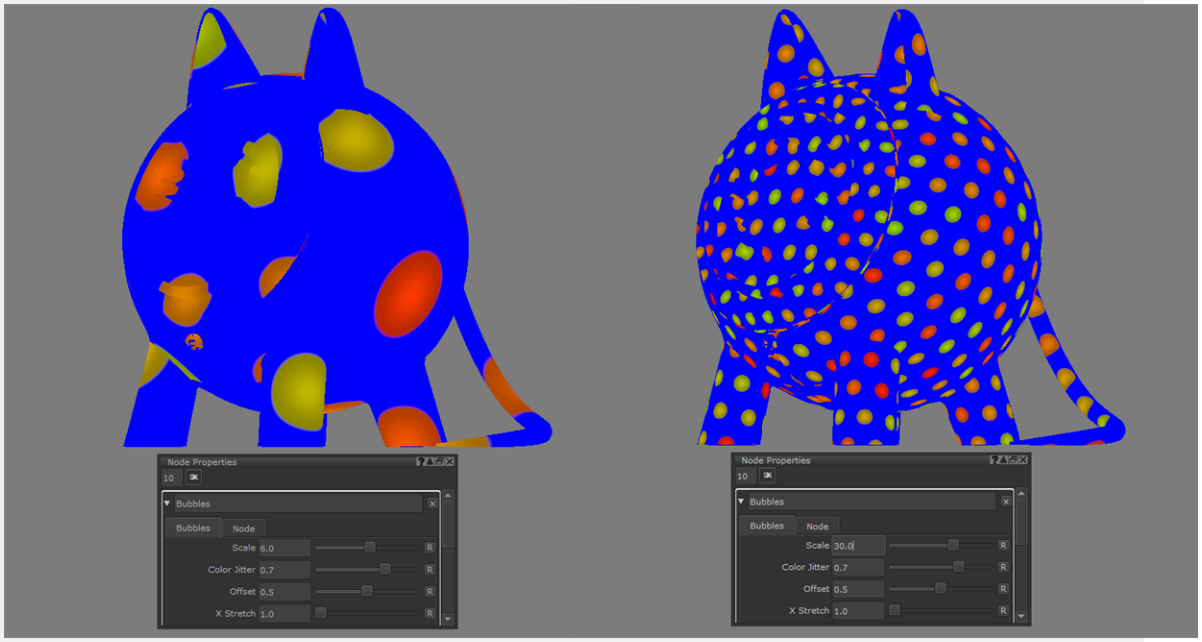
Different variations of shapes and patterns achieved using the Bubbles node.

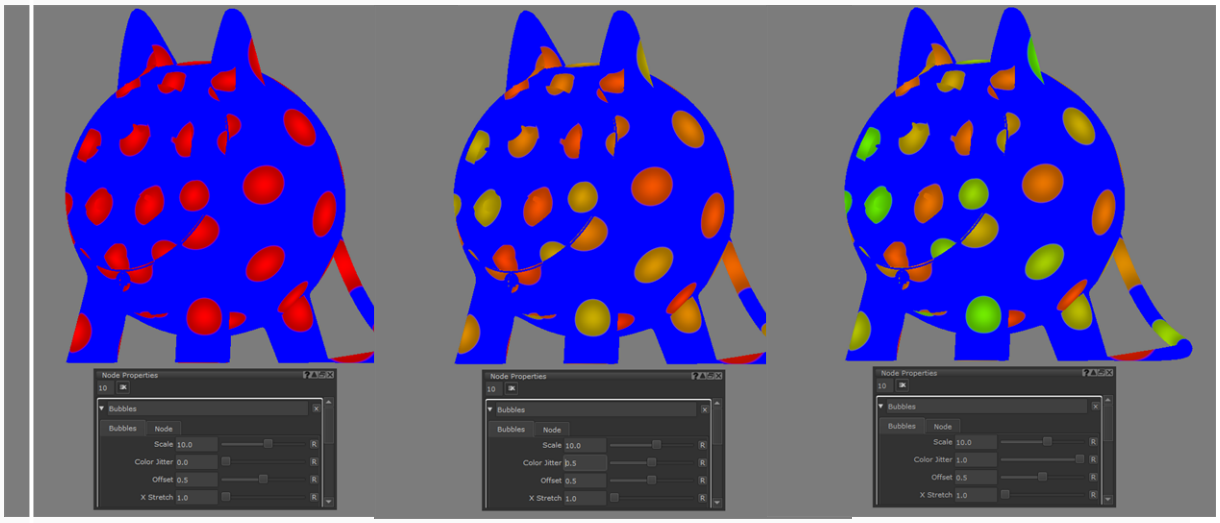
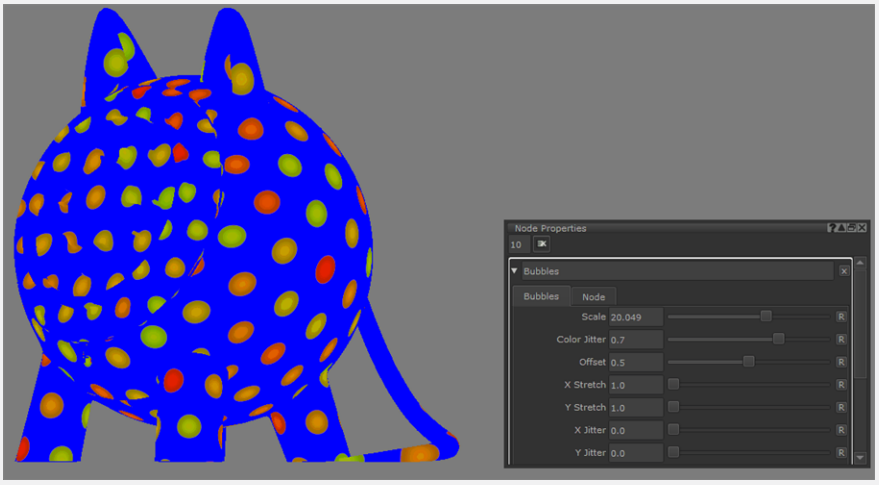


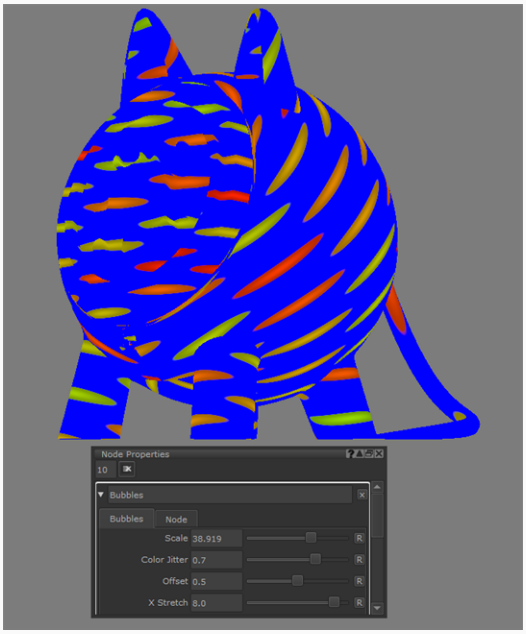
Bubbles node examples.

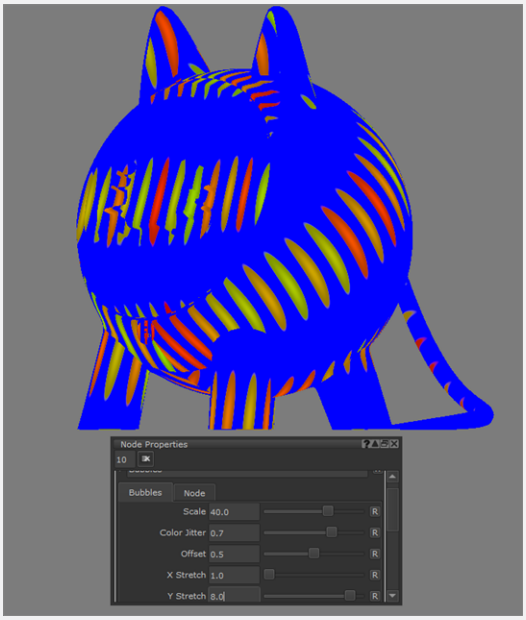
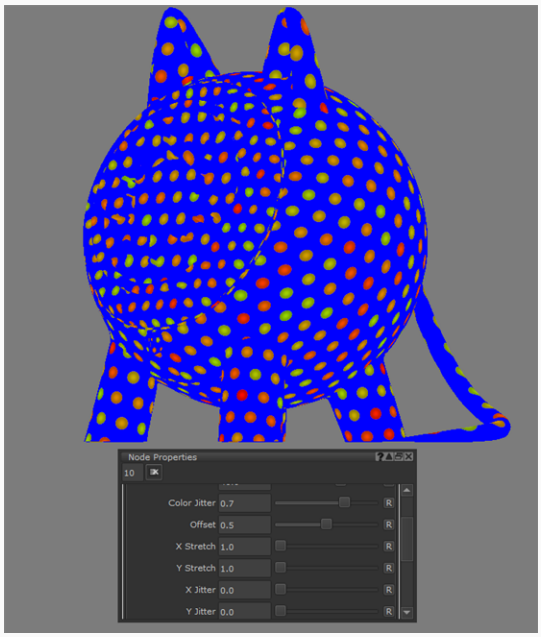
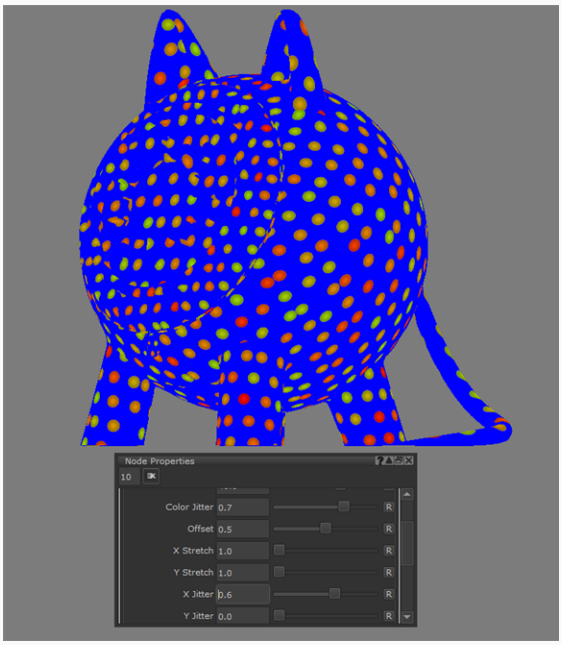
Inputs and Controls

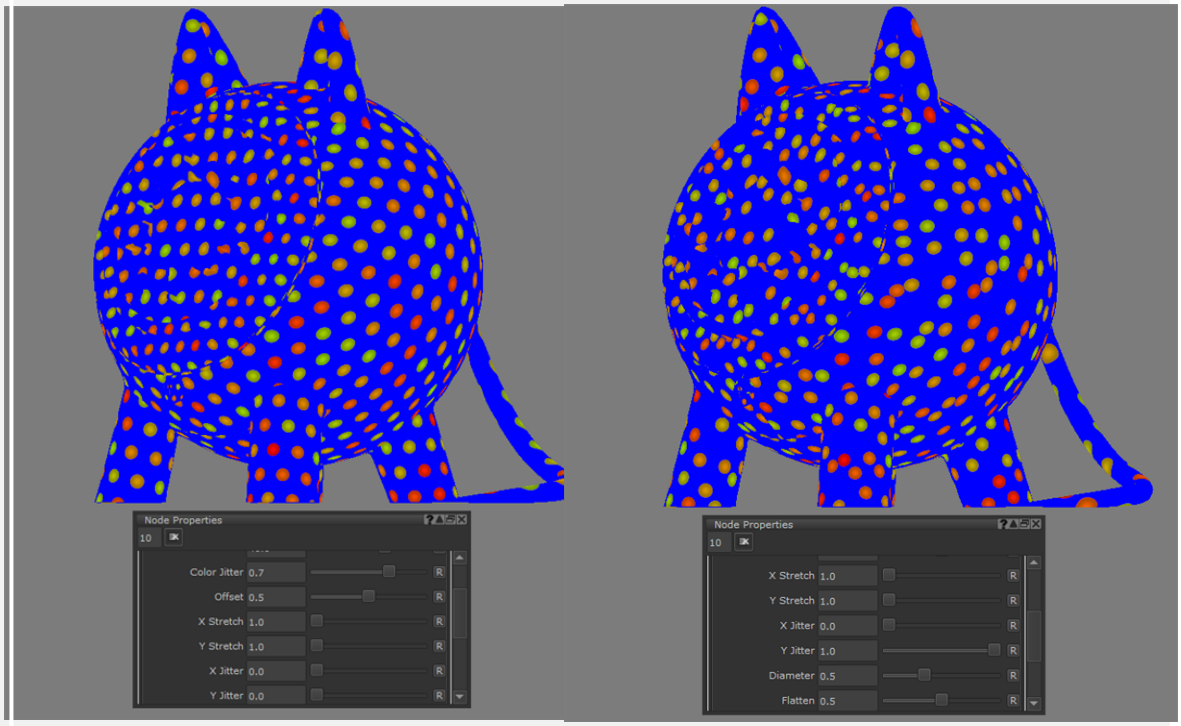
Connection Type	Connection Name	Function
input	Position	Any position data can be connected to the Position input, such as the UV and Position nodes. You can use any type of node to modify the mapping of the pattern. For example, you can use the Scale or Vector node in conjunction with the Position or UV node to stretch or offset the position (data) from the Position or UV node. If no node is connected, the node defaults to UV space. For more information, check the workflow example under Position Node.

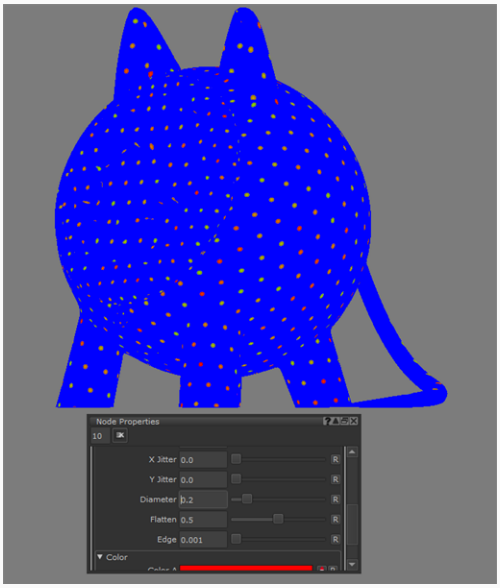
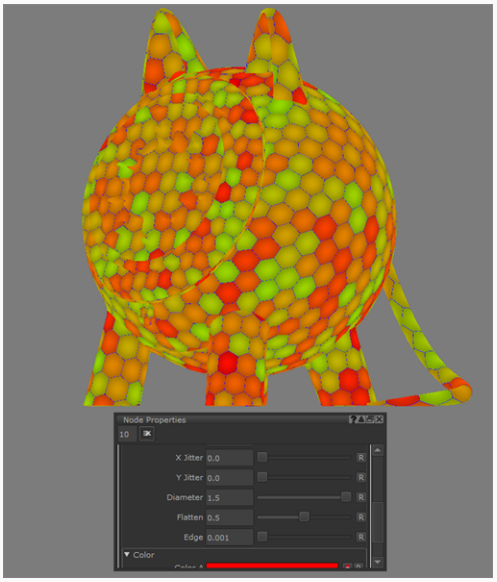
Control (UI)	Default Value	Function
Scale	10.0	Controls the size of each cell.  Scale: 6.0 Scale: 30.0

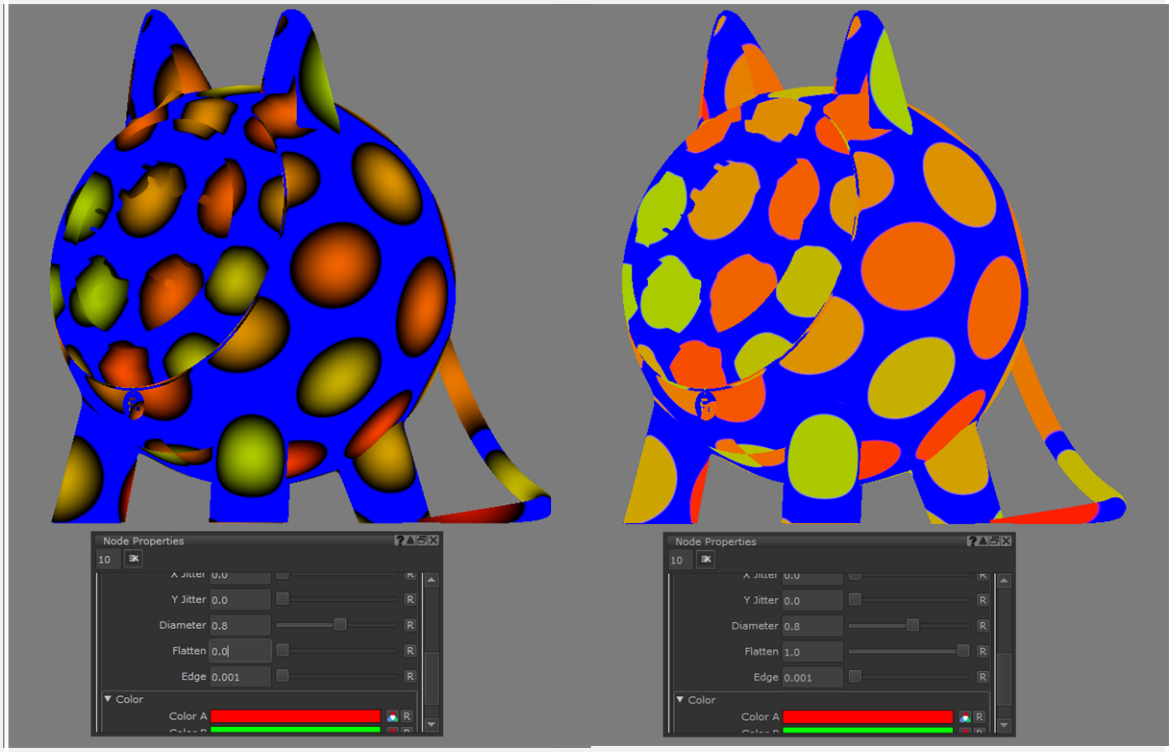
Control (UI)	Default Value	Function
Color Jitter	0.7	Control the variation between **Color A** and **Color B**. Setting it to 0 turns off the effect, while increasing the value gradually blends the colors for each cell.  Color Jitter: 0.0 Color Jitter: 0.5 Color Jitter: 1.0
Offset	0.5	Adjusts the position of every alternate row of cells along the X-axis. 
X Stretch	1.0	Stretches the pattern scale in the X axis.

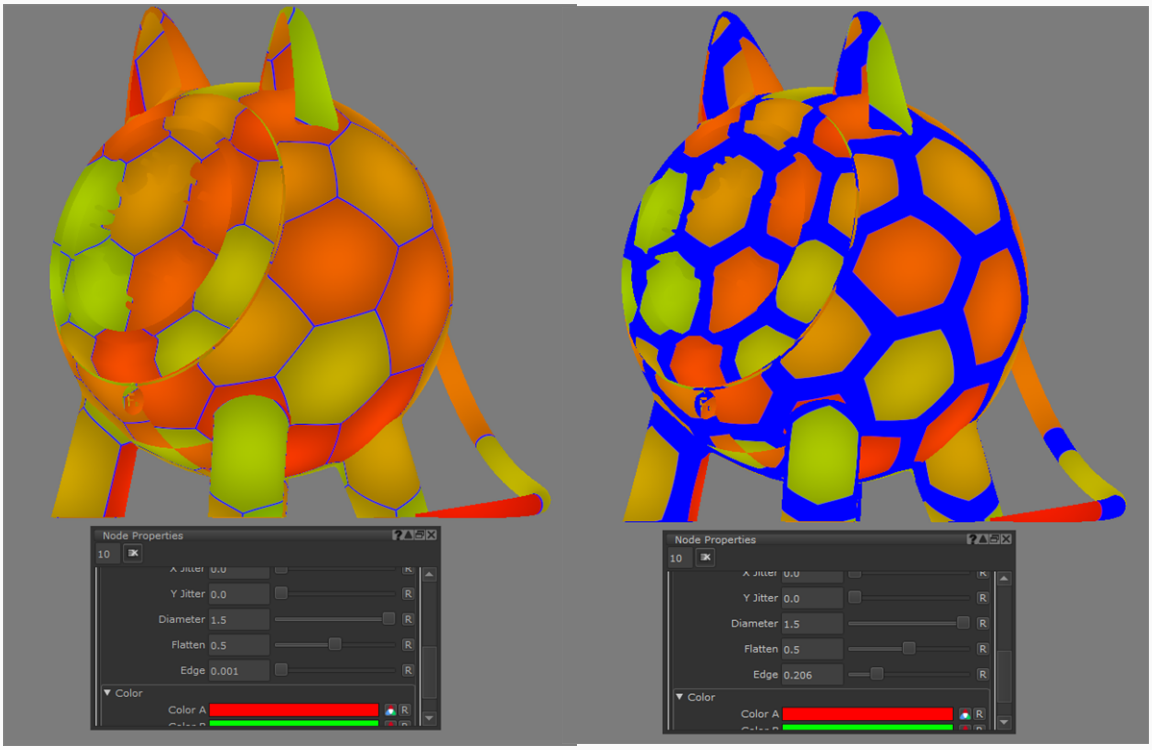
Control (UI)	Default Value	Function
		 **X Stretch**: 8.0
Y Stretch	1.0	Stretches the pattern scale in the Y axis.

Control (UI)	Default Value	Function
		 **Y Stretch**: 8.0
X Jitter	0.0	Randomize cell locations by altering their values on the X axis.  X Jitter: 0.0  X Jitter: 0.6

Control (UI)	Default Value	Function
Y Jitter	0.0	Randomize cell locations by altering their values on the Y axis.  Y Jitter: 0.0 Y Jitter: 1.0
Diameter	0.5	Changes the diameter of each cell.

Control (UI)	Default Value	Function
		 Diameter: 0.2  Diameter: 1.5
Flatten	0.5	Adjusts the contour of cells by either adding or removing a dark gradient.

Control (UI)	Default Value	Function
		 Flatten: 0.0 Flatten: 1.0
Edge	0.001	Changes the width of the intersection between cells.

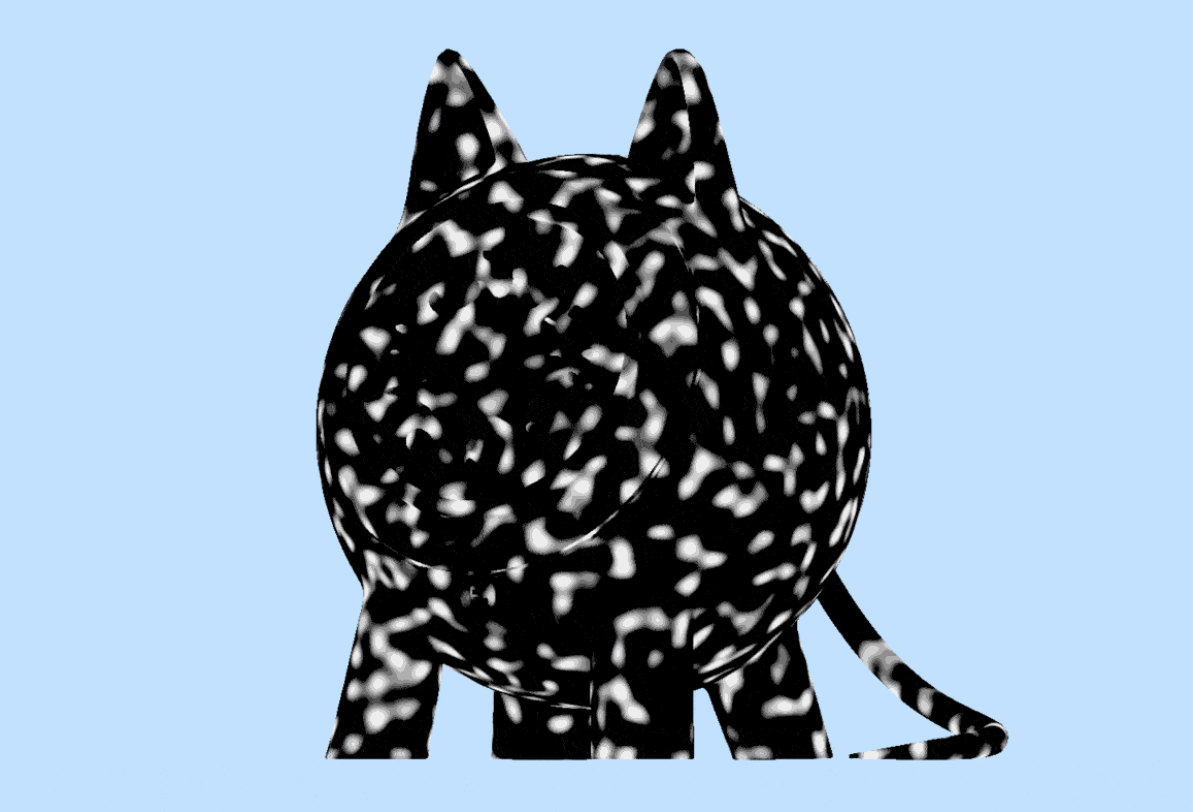
Control (UI)	Default Value	Function
		 Edge: 0.0 Edge: 0.2

Color

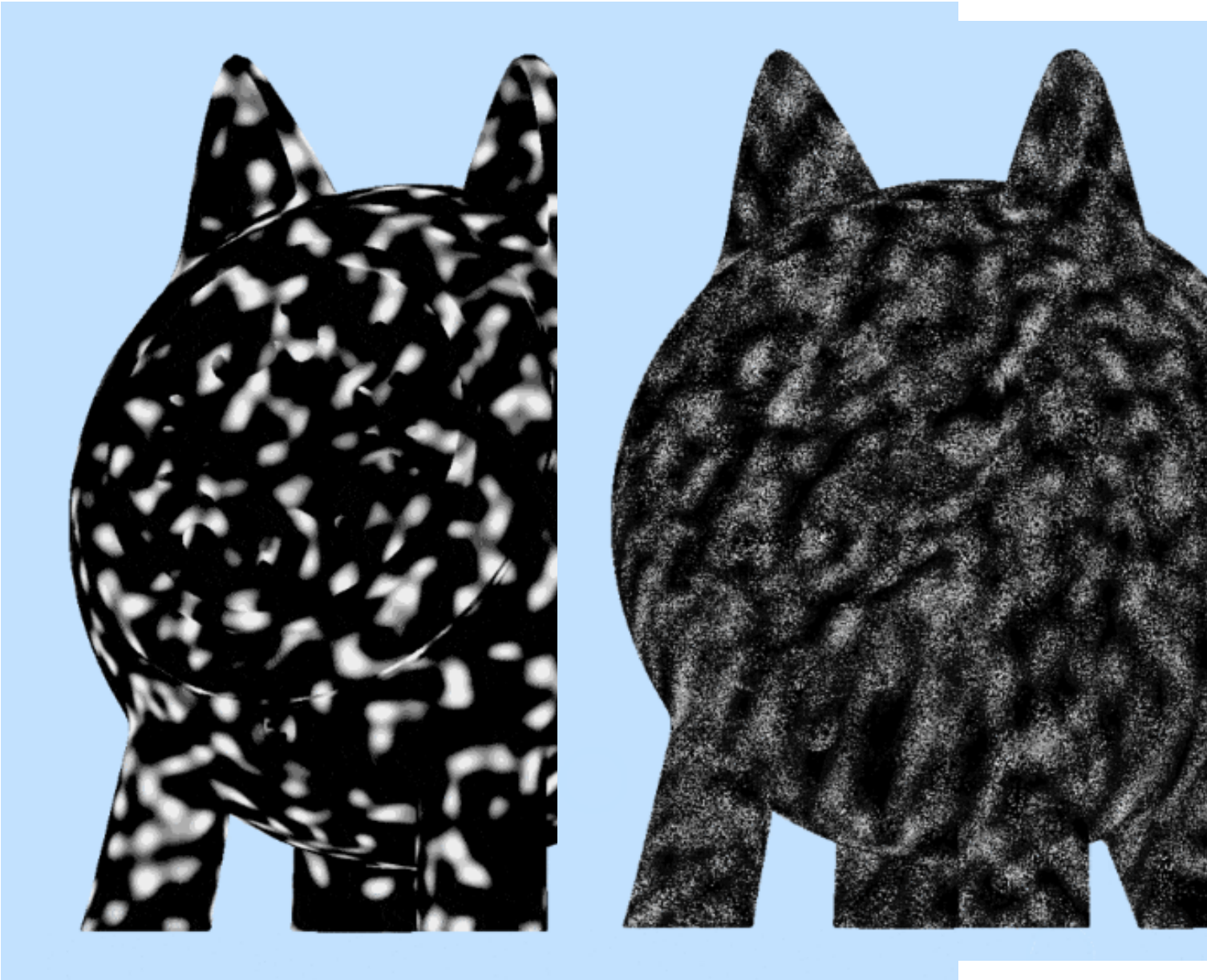
Color A	Red	Changes the color of the initial cells.
Color B	Green	Changes the color of the last cells.
Background Color B	Blue	Changes the color of the background/edges of the cells.

Camouflage Node

The Camouflage node is a procedural noise node that produces a camouflage or splatter effect by mixing together two noise signals, Frequency A and Frequency B, with a distance field, Threshold, to produce a variety of patterns. The Camouflage node can be combined with other procedural nodes to create more complex and organic patterns.



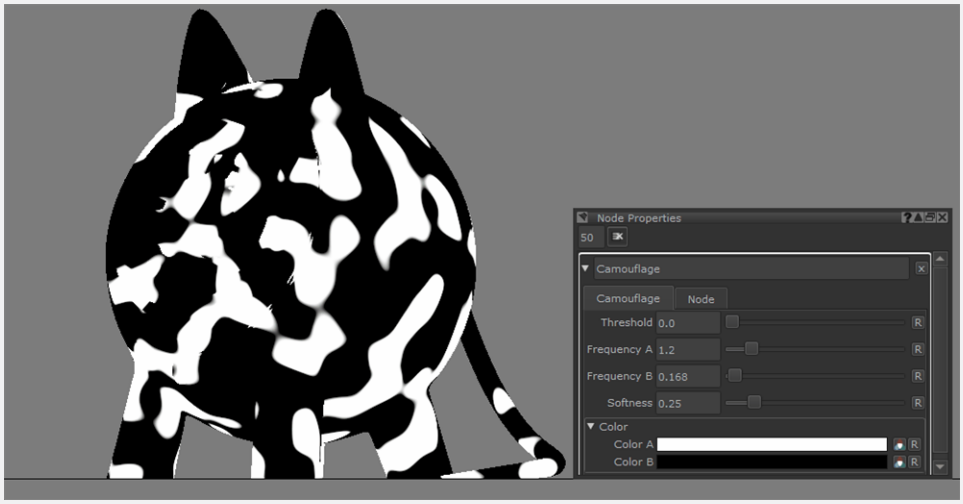
Adjusting the Camouflage node's settings.

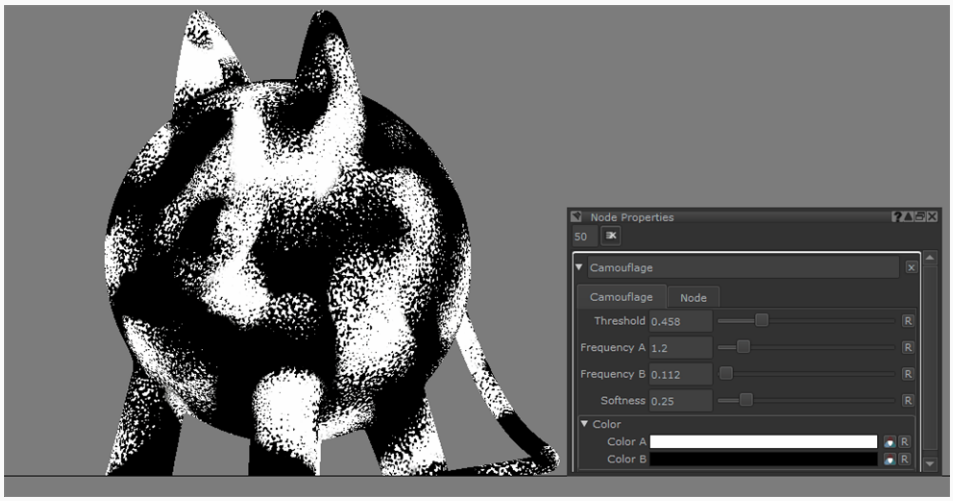
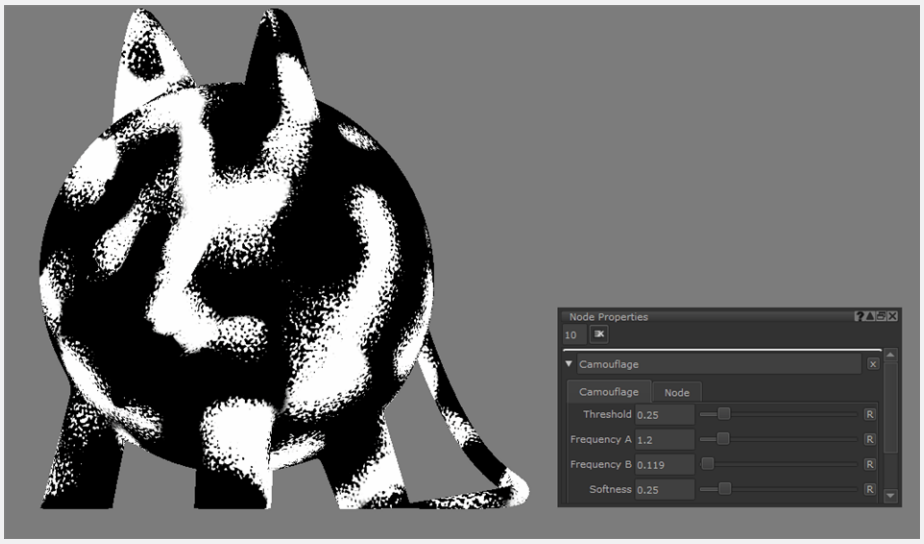


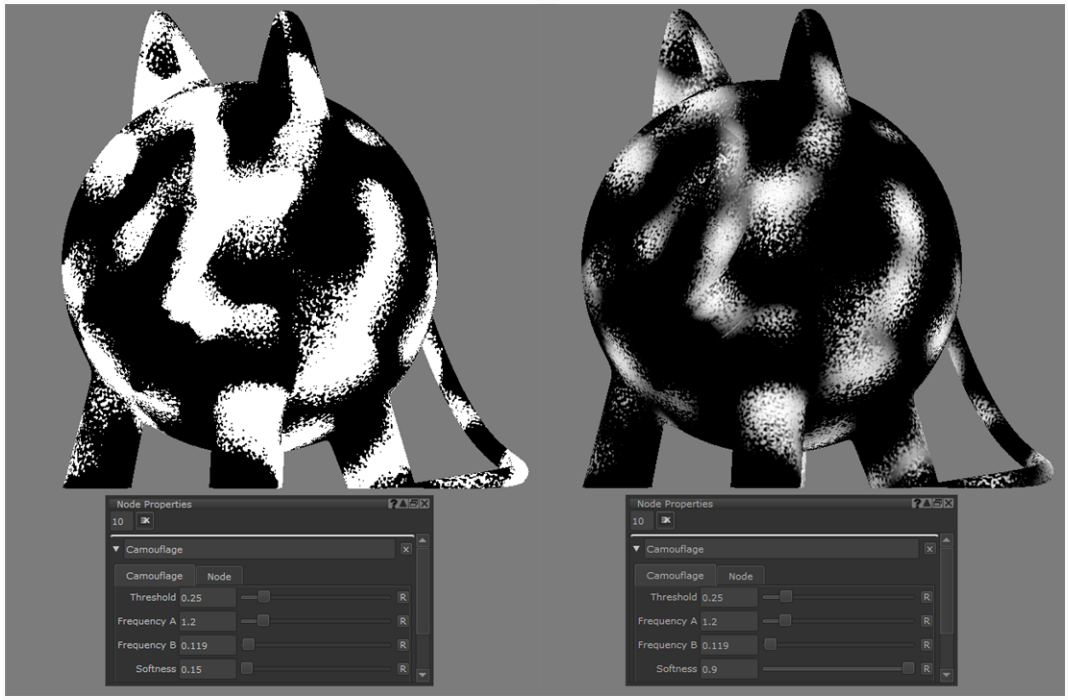
Camouflage node examples

Inputs and Controls

Connection Type	Connection Name	Function
input	Position	Any position data can be connected to the Position input, such as the UV and Position nodes. You can use any type of node to modify the mapping of the pattern. For example, you can use the Scale or Vector node in conjunction with the Position or UV node to stretch or offset the position (data) from the Position or UV node. If no node is connected, the node defaults to UV space. For more information, see Position Node.

Control (UI)	Default Value	Function
Threshold	0.25	Increasing the **Threshold** will result in a more splattered effect. 
Frequency A	1.2	Adjusts noise frequency to be more detailed by layering smaller, textured noise **Frequency A**, to the broader **Frequency B**, enhancing the larger noise shape with finer details.

Control (UI)	Default Value	Function
		
Frequency B	1.1	Adjusts the overall noise frequency on a broader level by increasing or decreasing the main/larger noise pattern. 
Softness	0.25	Controls the edge fade of the camouflage pattern. Lower values make the edges more crisp.

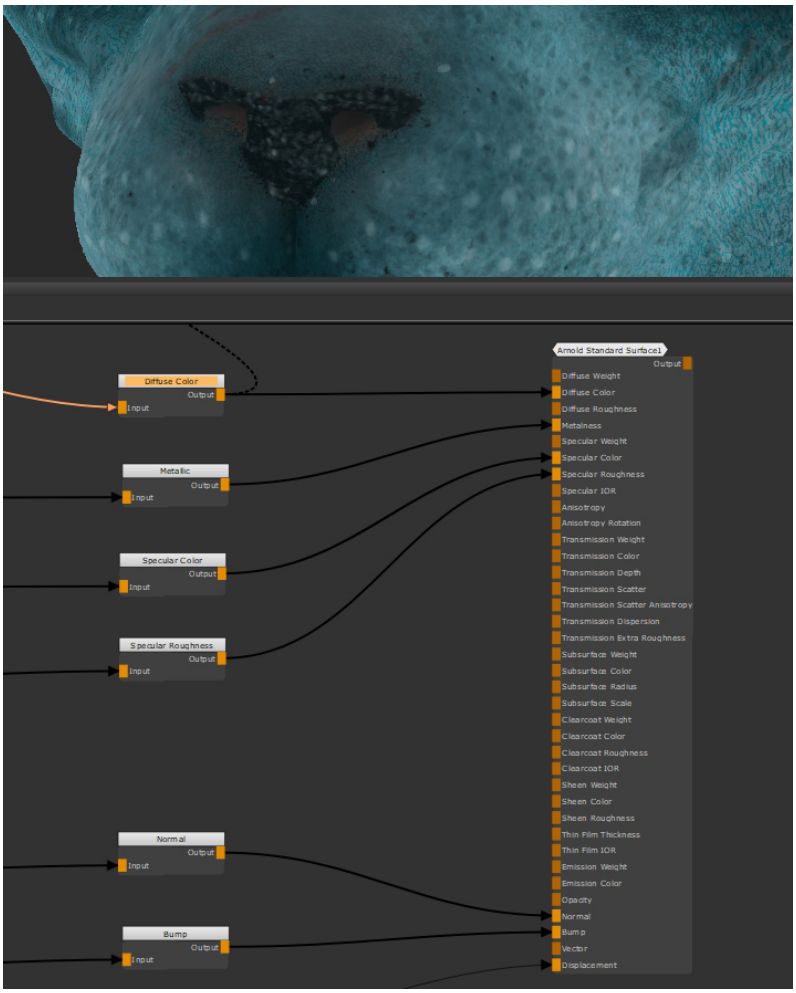
Control (UI)	Default Value	Function
		 Softness: 0.15 Softness: 0.9

Color

Color A	White	Changes foreground color of the camouflage pattern.
Color B	Black	Changes background color of the camouflage pattern.

Channel Node

Access: **Nodes** > **Misc** > **Channel**



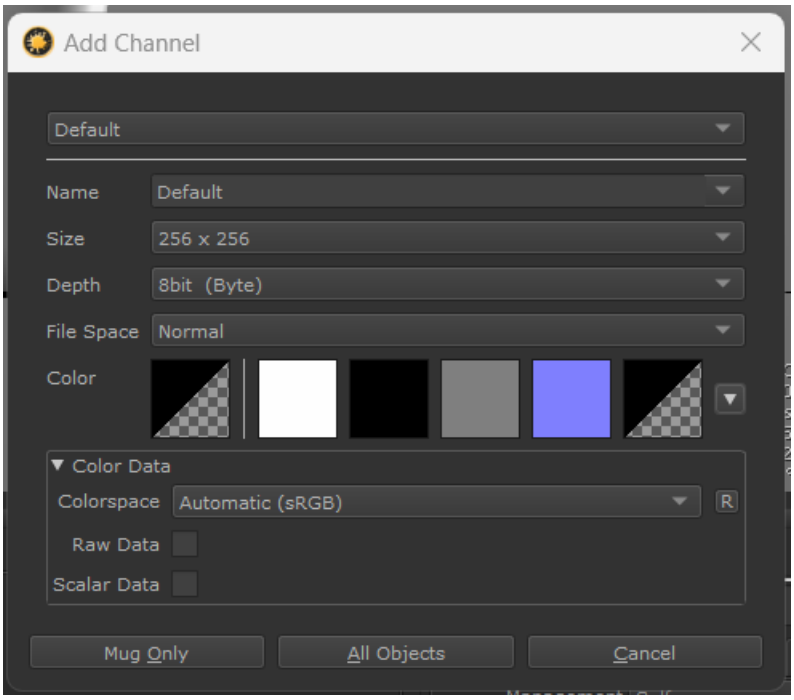
Multiple Channel nodes being used with an Arnold Shader node.

The Channel node is an important node for Mari because it's essential for getting textures out of the software and into the renderer you need to use. The Channel node usually comes at the end of your node tree, where your textures are going to be exported from. It defines the size, bit depth and view transform of your texture maps.

Channel Node Inputs

Input - Defines the textures and values that are be read and ultimately exported as texture channels from your Mari project.

Channel Node Properties



The **Add Channel** dialog.

Size *dropdown list*	Chooses the resolution of the images when exported.
Depth *dropdown list*	Chooses the bit depth of the images that are exported.
File Space *dropdown list*	This defines how values are exported. The default of **Normal** changes nothing. For **Vector**, the data will be treated as vector data and remapped. **Vector_Y_Flip** does the same but inverts the Y component. **Normal_Y_Flip** changes nothing but inverts the Y component.
Color	You can select the leftmost Color square to spawn color picker, or use any of the preset colors (white, black, mid grey, normal, transparent alpha). Any saved Preset Colors can also be accessed here via the button.

Color Data

Colorspace *dropdown list*	If the **Raw Data** checkbox is disabled then you can define the color space of the color data of the channel.

Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.

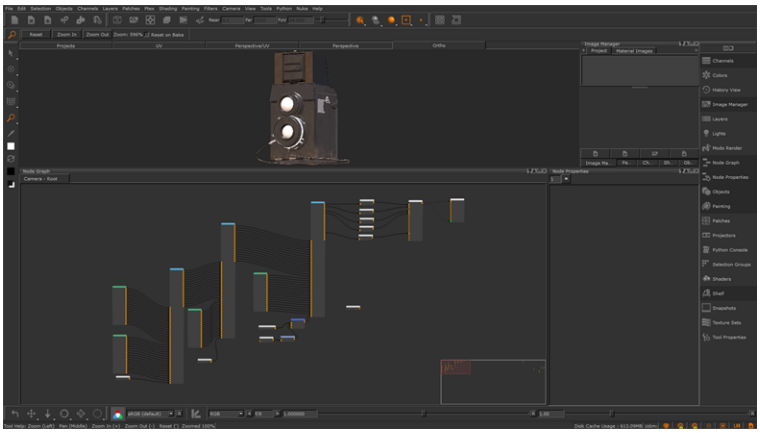
Raw Data *checkbox*	A toggle to define if the data is raw or in a different color space. Disabling this allows you to change the color space dropdown, potentially resulting in a conversion when displayed and exported.
Scalar Data *checkbox*	Defines if values going through the channel are color or **Scalar Data**. If you enable this the view transform will be changed to **None**.

Channel Node Workflow Example

Setting up Channels for Export and Lookdev

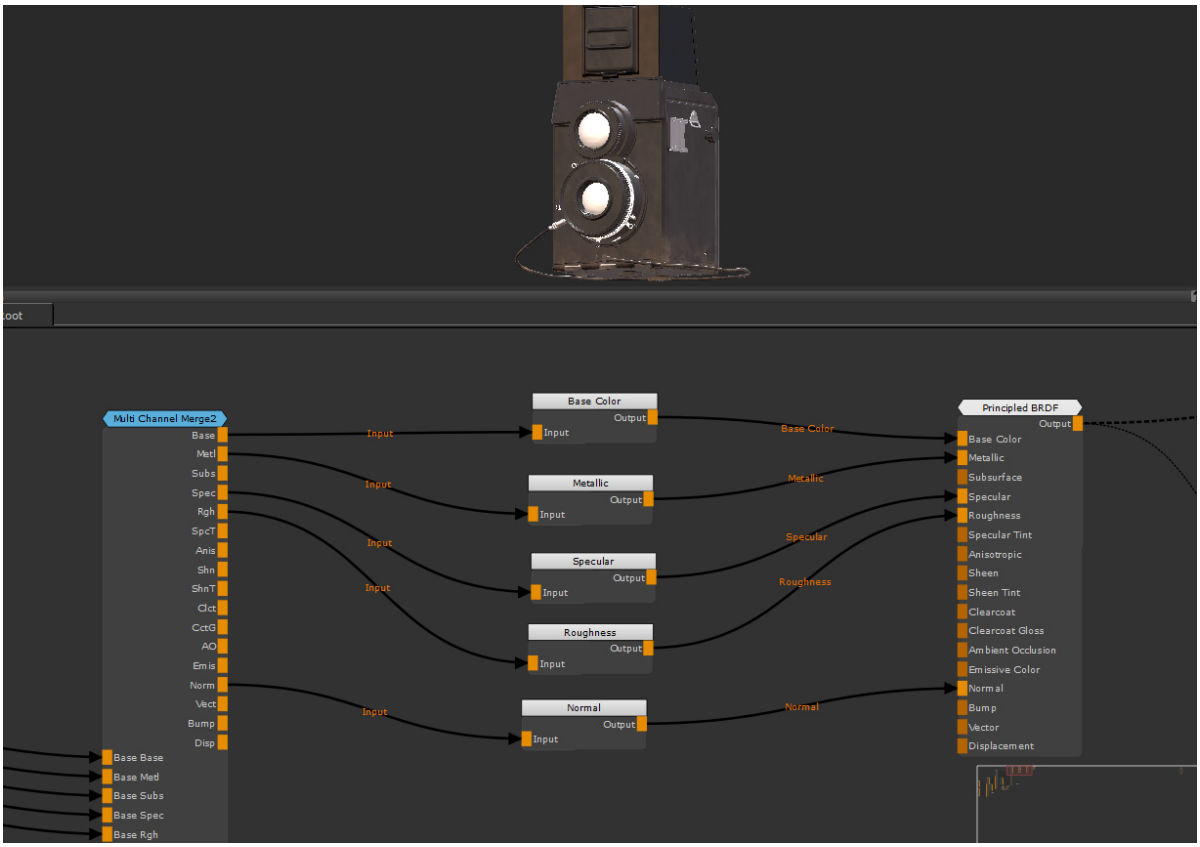
In Mari, it's important to set up channels for every texture map you wish to export, and have them connected so that they are always displayed with the correct view transform. A benefit of having them as Channel nodes is that from there we can hook them up to a shader to show all our material channels at once.

Tip: For a deeper look at material workflows checkout the For some quick video tutorials of the new 4.5 Material System, check out Mari 4.5 Fundamentals video series here. documentation.



The Node Graph of this textured asset using a material based workflow.

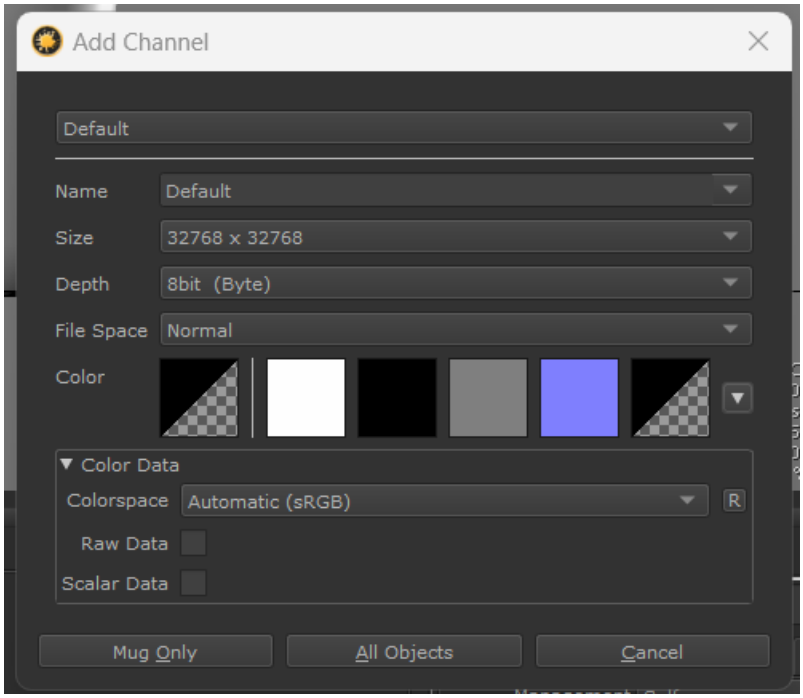
In the example we have our final Multi-Channel Merge node which is outputting multiple ready for export texture channels, however our renderer will not need all of them. We already have most of them connected toChannel nodes, but have yet to set up a channel for our Bump Map. In the viewport, the shader we are viewing is looking quite flat too, so it will be great to get the bump hooked up.



Note: It's important to hook up your Channel nodes before adding them to the Shader so the view transform is correct.

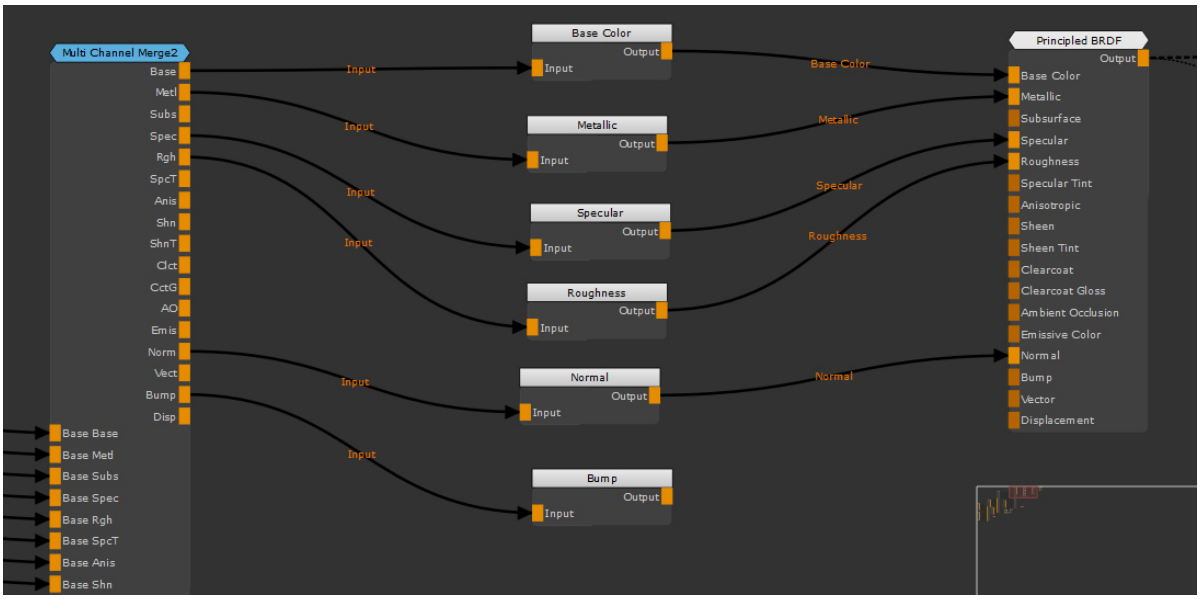
First a Channel node needs to be added. When this is done a dialog pops up requesting the properties of the new channel. You need to make sure when selecting a channel that the settings are correct to your project. Since this is going to a bump channel, the **Depth** has been changed to 16-bit.

Tip: Mari exports the Size you set here by default, so make sure that is set up correctly to whatever size texture you are inputting.



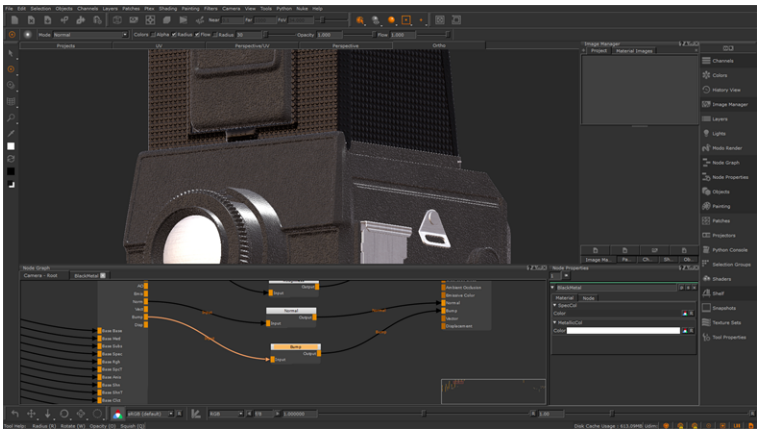
The Channel node creation UI dialog.

Since a Bump map's information is pure data, we don't want it to be affected by the view transform inside of Mari. Clicking the **Scalar Data** checkbox under color data will correctly display the Bump map. Being aware of this is important when creating new channels.



Connecting the Material node's **Bump** output to the **Bump** Channel node.

Whatever is plugged into the input of the **Bump** Channel node will export as a bump map for us when we use the **Export Manager**. Hook up the Bump channel from the Multi Channel Merge node and then plug that into the shader afterwards.



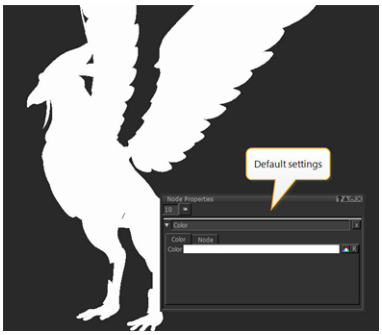
Everything hooked up and working.

Now that's all plugged in we can see the Bump when viewing our shader, which is looking more realistic. We can also export that properly now which is the main purpose for adding Channel nodes.

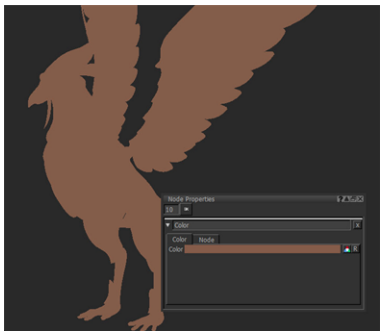
Color Node

Access: **Nodes** > **Basic** > **Color**

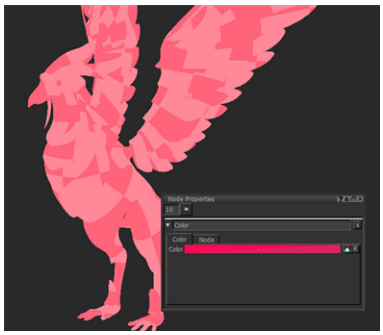
The Color node is a basic node that lets you change the color of your objects at any time in a non-destructive way. It only outputs color and does not have any inputs.



Default ***Color***



Color *set to brown*



Color *set to pink with some transparency*

Color Node Inputs

No Inputs

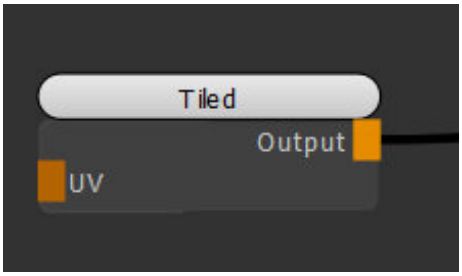
Color Node Properties

Color *swatch*	Changes the color. Click to select a color from the Select Color Dialog.

Color Node Graph Workflow Example

In the example below, let's use an Add node to blend a Tiled and a Color node together.

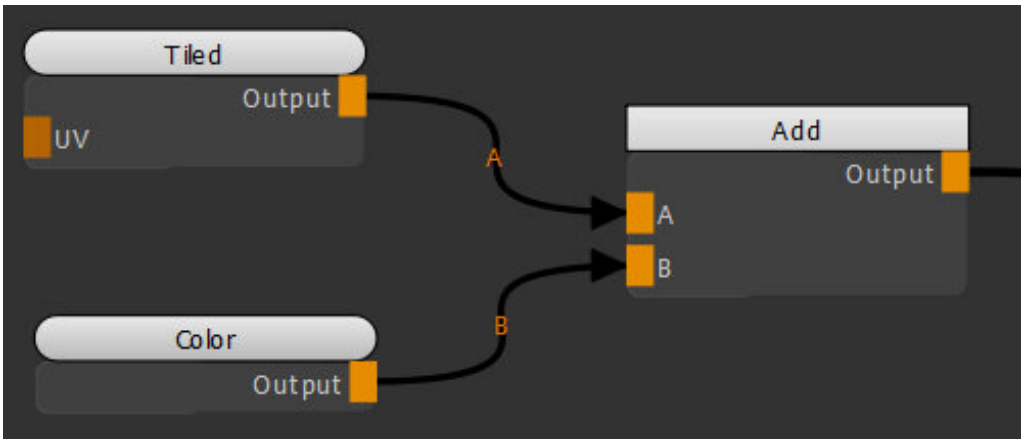
1. Create a Tiled node.



Adding a Tiled node to the scene.

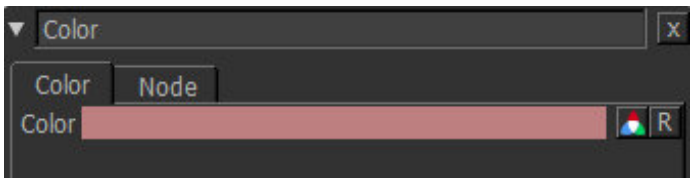
Texture file contained in the Tiled node.

2. Create an Add node. Connect the Tiled node to the **A** input of the Add node. Create a Color node and connect it to the **B** input of the Add node.



Adding a Colornode and Add node to the scene to change the color of the object.

3. Open the node **Properties** of the Color node and select a color.



A new color is selected.

Notice how the Tiled and Color nodes blend together.

The texture has changed color.

Color to Scalar Node

Access: **Nodes** > **Filter** > **Color to Scalar**

The Color to Scalar node converts color RGB values to scalar RGB values, which can be useful for visualizing scalar data with color values.

Color to Scalar Inputs

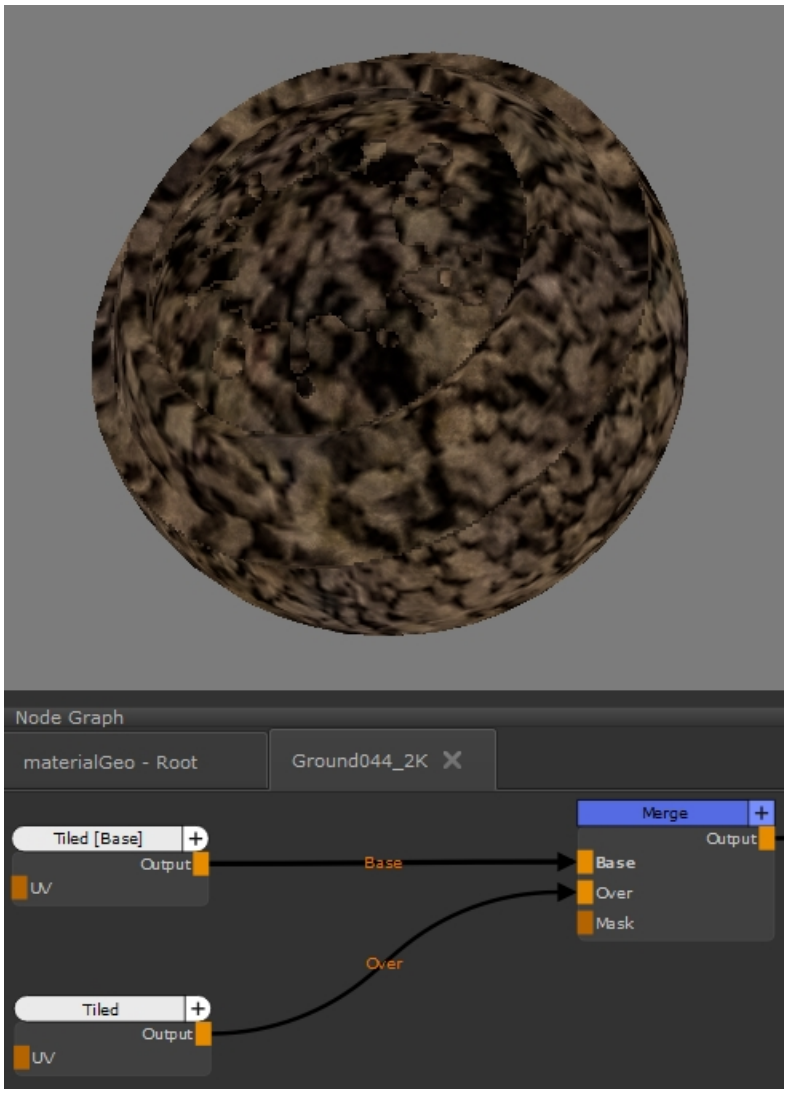
Input: The stream you want to convert to scalar data.

Color to Scalar Node Properties

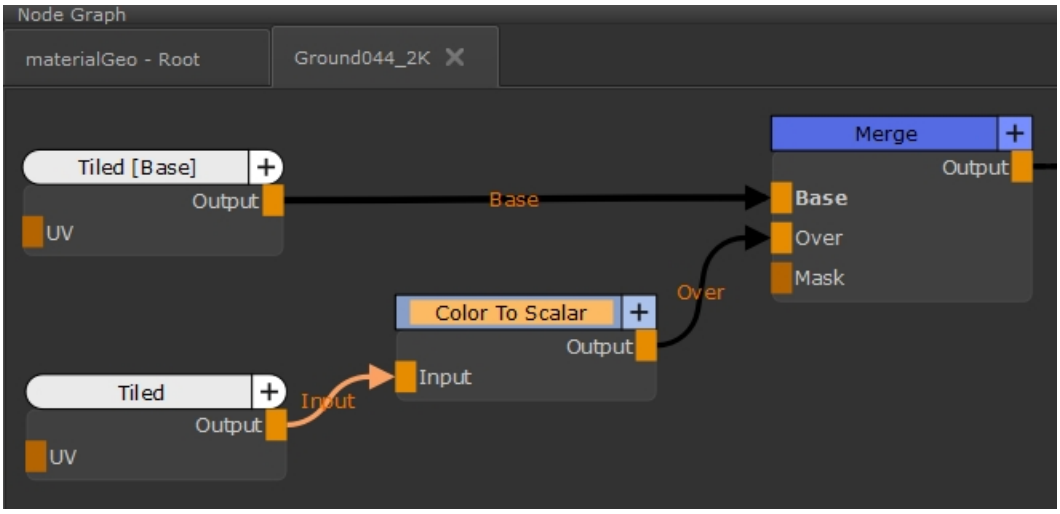
The Color to Scalar node does not include any node properties.

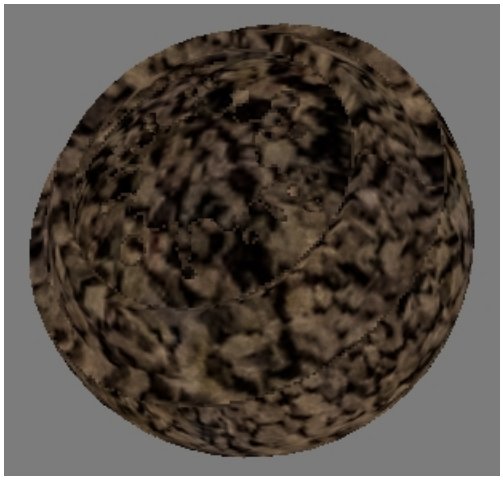
Color to Scalar Node Workflow Example

This example shows the internal node tree in a Material node. In this case, we're blending a tiled color image of some gravel, **Tiled**, with the base color, **Tiled [Base]**, using an overlay blending mode in the Merge node.

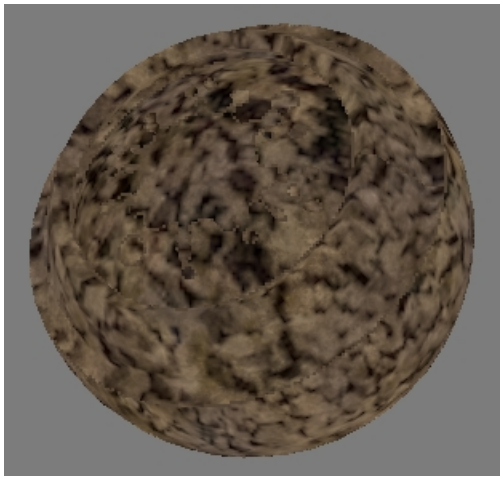


The tiled image is color data, so it must be converted to scalar using a Color to Scalar node before blending to achieve the correct brightness. In the example, the Color to Scalar node is disabled and enabled to show the difference in output.





Color to Scalar disabled.



Color to Scalar enabled.

Compare node

Access: **Nodes** > **Misc** > **Compare**

The compare node is a simple node that enables you to feed in two different visual streams, and compare them side by side using a wipe effect. There are three distinct wiping modes to choose from, so you have full control over the visual comparison.

Compare node inputs

- **A** — The first visual stream to compare.
- **B** — The second visual stream to compare.

Compare node properties

Wipe	*slider*	Controls the position of the wipe boundary between inputs A and B. At -1 only A is visible; at 1 only B is visible.
Space	*dropdown*	Defines how the compare wiper will interact with your object: • **Screen space** — keeps a fixed separation line across your viewport. • **Object space** — ties the wipe directly to your 3D model's local axis • **Object bound screen space** — sweeps across the viewport axis, but scales perfectly to the specific dimensions of your object.

Constant Node

Access: **Nodes** > **Basic** > **Constant**

Under the hood, Mari passes around a set of four numbers along a connection between two nodes. The Constant node is solely used as a set of four numbers grouped under one slider, see Constant Node Properties. You can use the Constant node in conjunction with nearly any node, it depends on your workflow. For example, you can use it to adjust the position or color information, or you can use it as a number to change the value of another node, for instance, a shading node.

Constant Node Inputs

No inputs.

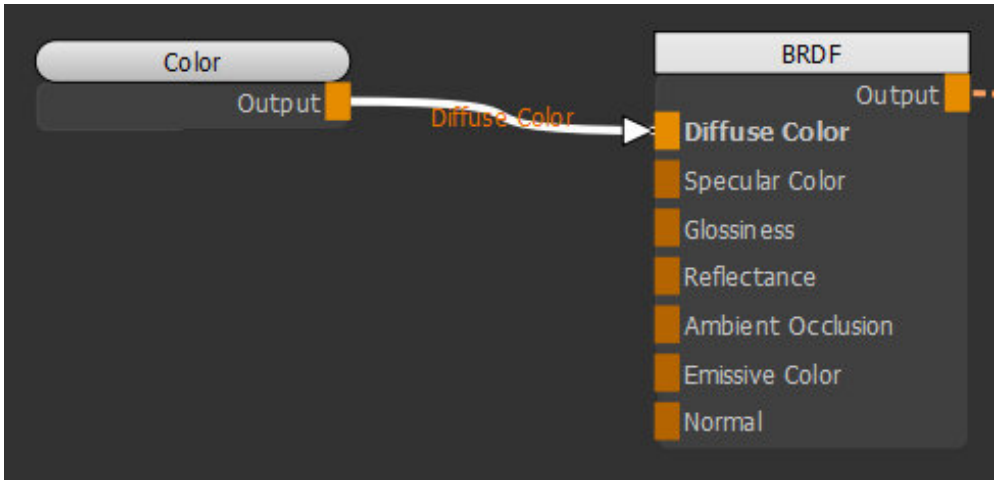
Constant Node Properties

Constant *floating point control*	Changes the value of the RGB or RGBA **Component**.
Component *dropdown*	Select whether the **Constant** value is used for the **RGB** or **RGBA Component**.

Constant: Node Graph Workflow Example

In this example, we are going to use the Constant node to make changes to the Glossiness channel of the BRDF shader.

1. Create a Color node and connect it to the **Diffuse Color** input of the BRDF node.



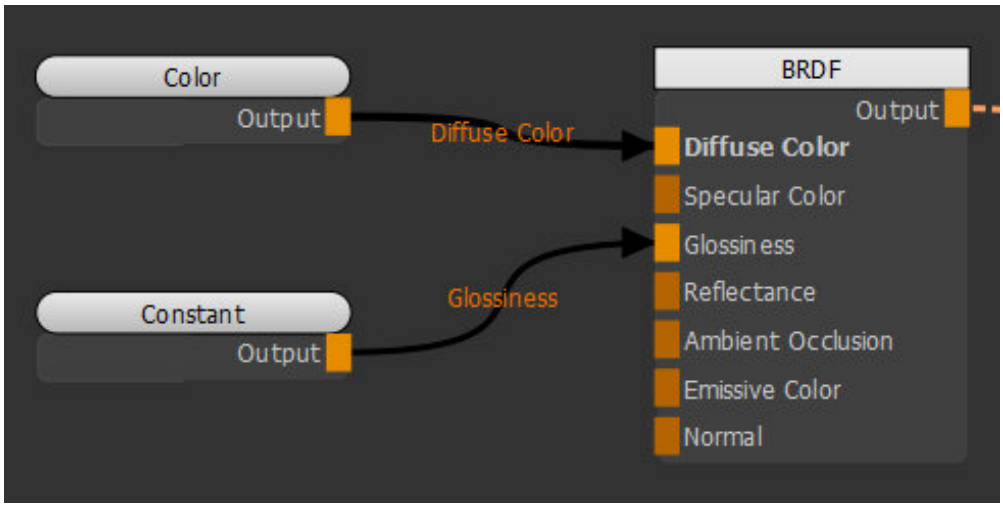
*Connecting a Constant node to the **Diffuse Color** input of the BRDF node to texture the object*

2. In the Color node's node **Properties**, change the color to red.

   The object is now red.

The Color node gives a red texture to the object.

3. Create a Constant node and connect it to the **Glossiness** input of the BRDF node.



*Creating a Constant node and connecting it to the **Glossiness** input of the BRDF node.*

Notice how moving the **Constant** slider of the Constant node (see Constant Node Properties) changes the Glossiness of the surface.

Vector Node

Access: **Nodes** > **Basic** > **Vector**

Under the hood, Mari passes around a set of four numbers along a connection between two nodes. The Vector node is solely used as a set of four numbers (x, y, z, and w). You can use the Vector node in conjunction with nearly any node, depending on your workflow. For example, you can use it to adjust the position or color information, or you can use it as a number to change the value of another node, for instance, a shading node.

Vector Node Inputs

No input.

Vector Node Properties

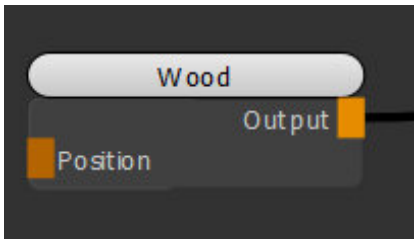
X *floating point control*	First component of the vector. If applied to RGBA, this is the red component.
Y *floating point control*	Second component of the vector. If applied to RGBA, this is the green component.
Z *floating point control*	Third component of the vector. If applied to RGBA, this is the blue component.
W *floating point control*	Fourth component of the vector. If applied to RGBA, this is the alpha component.

Vector: Node Graph Workflow Example

Changing the Position of Noise

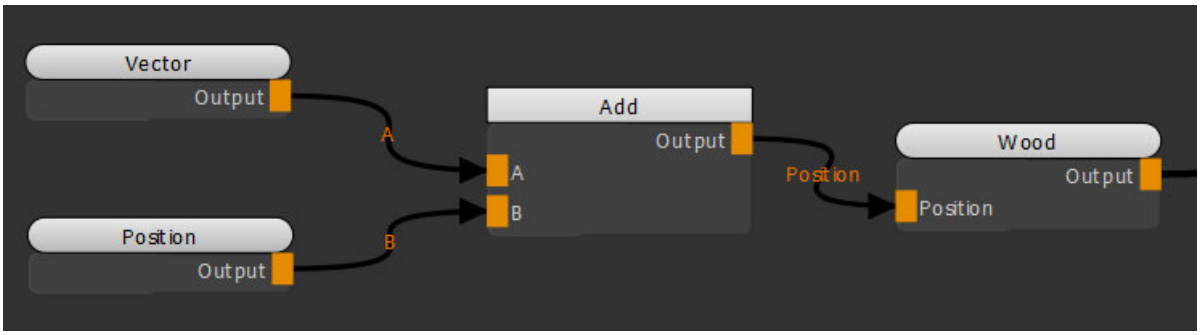
In this example, let's use a Vector node to change the position of a wood procedural applied to the object.

1. Create a Wood node.



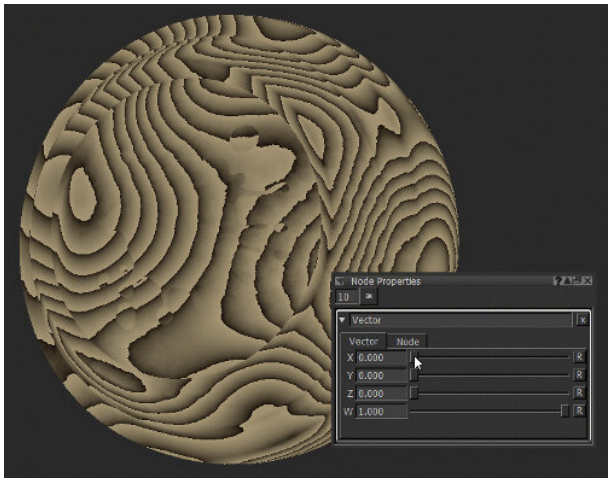
Adding a Wood node to texture the object.

2. Create an Add node and connect it to the **Position** input of the Wood node. Create a Vector and a Position node and connect them respectively to the **A** and **B** inputs of the Add node.



Adding a Vector node in conjunction with a Position node and an Add node to adjust the position of the texture.

Notice how moving the **X** slider of the Vector node (see Vector Node Properties) moves the texture on the **X** axis as the position information is changed.

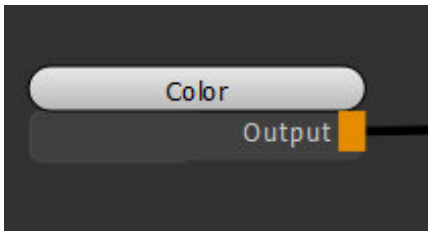


*Adjusting the position of the texture by changing the **X** value of the Vector node.*

Changing the Color of an Object

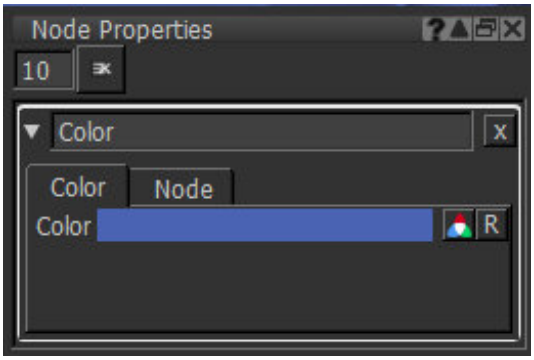
In this example, let's use a Vector node to change the color of the object.

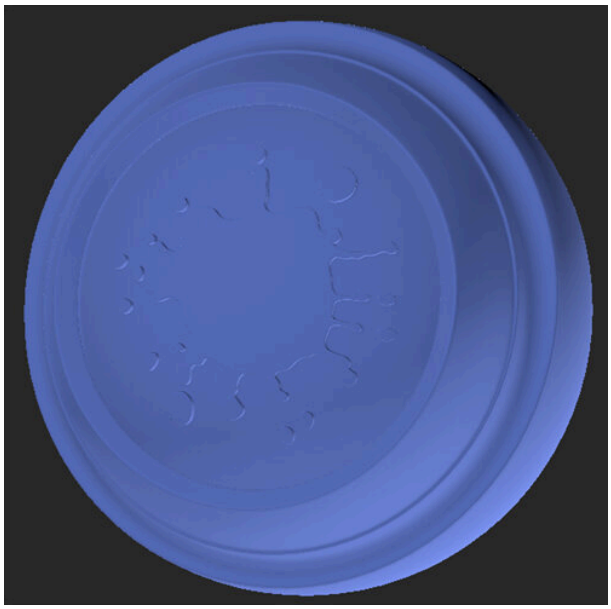
1. Create a Color node.



Original scene contains a Color node.

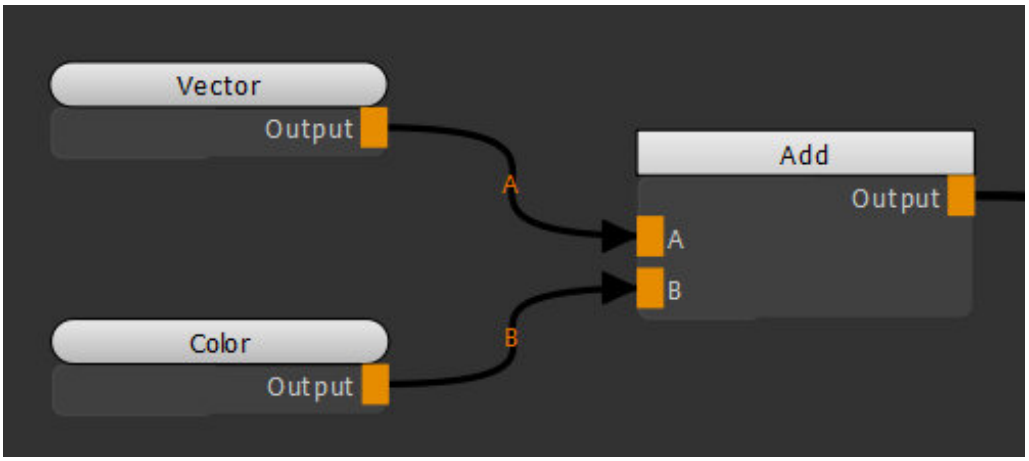
2. Open the Color node's **Node Properties** and change the **Color** to blue.





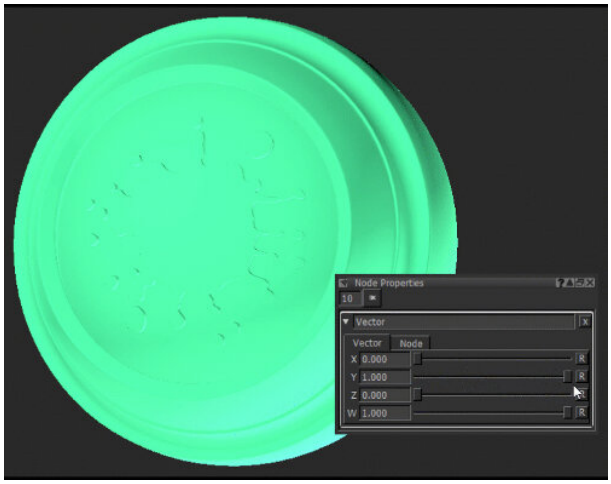
Color node applying a blue color to the object.

3. Create an Add node. Create a Vector node and connect the Vector node and the Color node respectively to the **A** and **B** inputs of the Add node.



Adding a Vector node in conjunction with an Add node to change the color of the object.

Notice how moving the **Y** slider of the Vector node (see Vector Node Properties) changes the green component of the Color node.

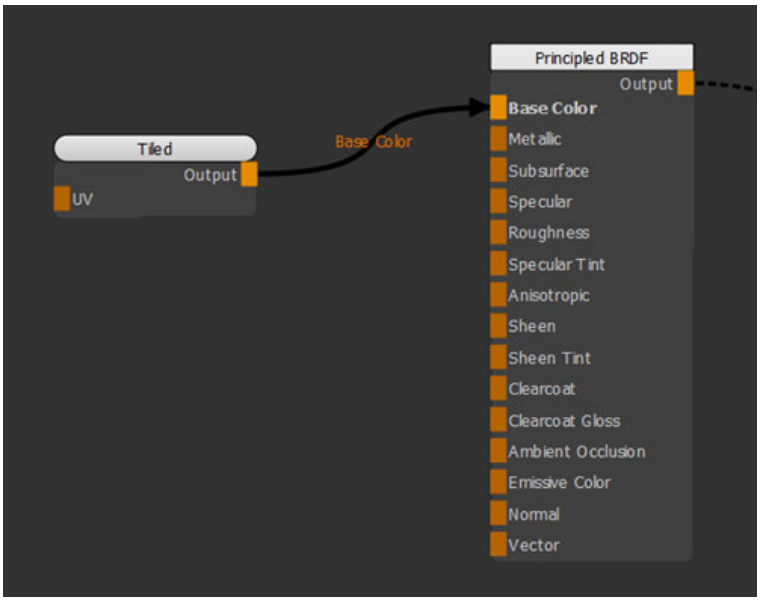


*Changing the green component of the Color node by adjusting the **Y** value of the Vector node.*

Using the Vector Node as a Number

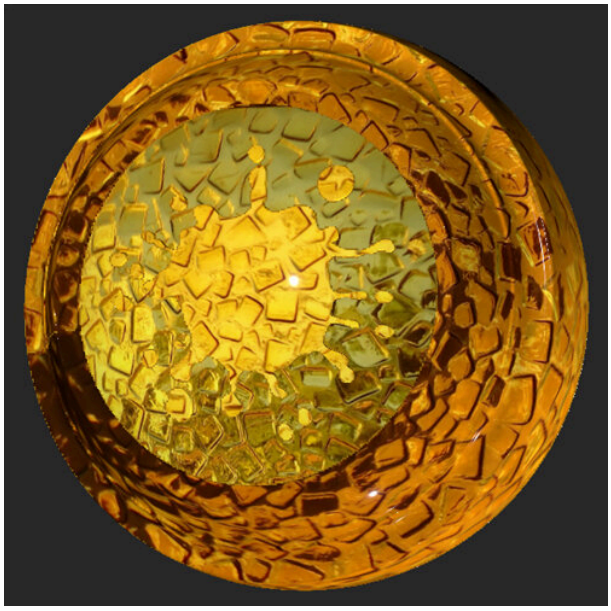
In this example, let's use a Vector node to make changes to the **Roughness** channel of the Principled BRDF shader.

1. Create a Tiled node and a Principled BRDF node. Connect the Tiled node to the **Base Color** input of the Principled BRDF shader.

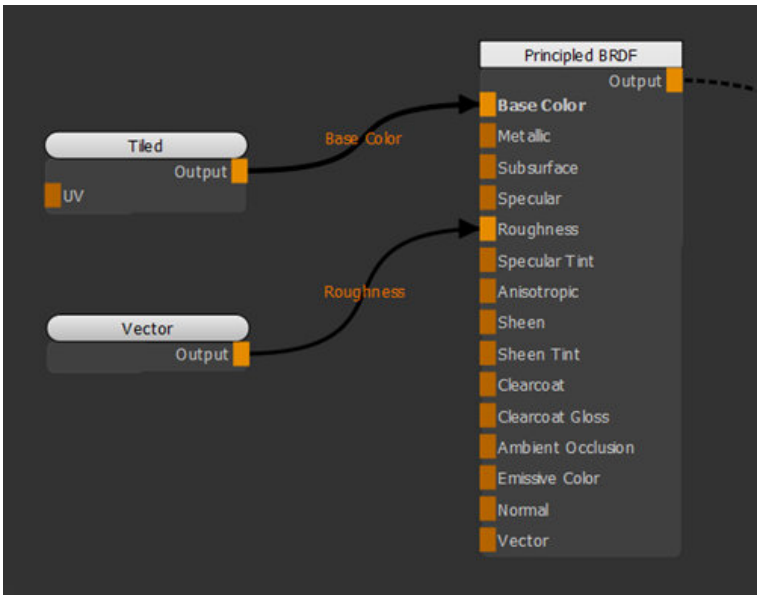


Connecting a Tiled node to the Principled BRDF node.

This is the visual output of the texture applied using the Tiled node.



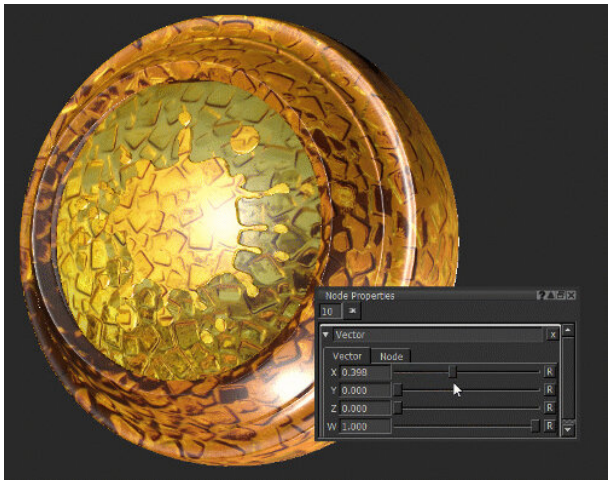
2. Create a Vector node and connect it to the **Roughness** input of the Principled BRDF.



*Adding a Vectornode and connecting it to the **Roughness** input of the Principled BRDF shader.*

Notice how moving the **X** slider of the Vector node (see Vector Node Properties) changes the roughness of the surface.

In this example, the Vector node is used as a number.

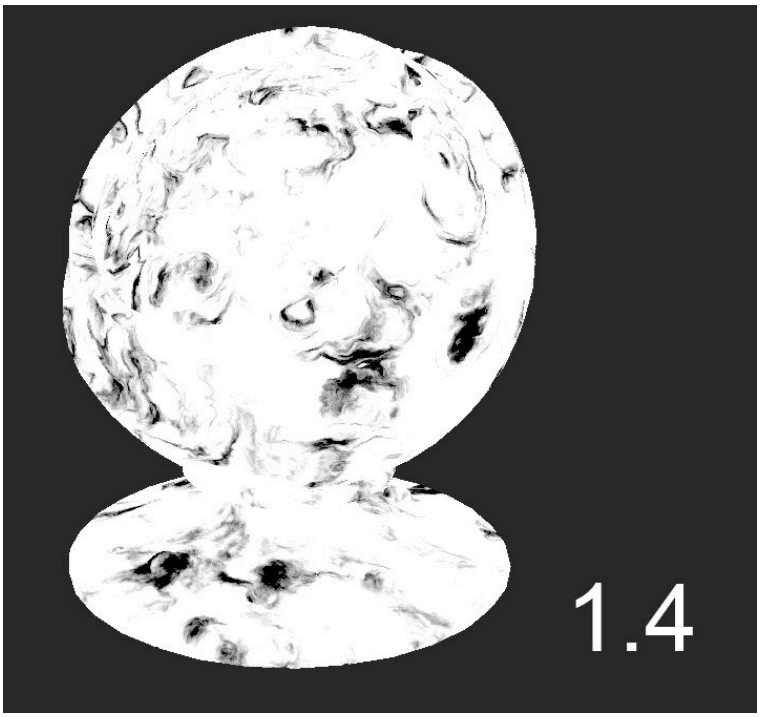


*Changing the roughness of the surface by changing the **X** value of the Vector node.*

Brightness Node

Access: **Nodes** > **Filter** > **Brightness**

The Brightness node is a simple node to brighten or darken your textures. A higher brightness value adjusts the brightness in a non-linear fashion, which means it will not lighten dark and light values the same amount. The Brightness node is a great way to make quick tweaks to your textures if they are too dark.



The Brightness node being used on an Oil node with a Brightness value of 0.1.

Brightness Node Inputs

Input: The data that is to be adjusted by the **Brightness**.

Brightness Node Properties

Brightness *text field, slider*	Adjusts the brightness of the input. Raising the value above 0.5 increases the values of the lighter values of the input but keeps the dark values similar due to the non-linear adjustment. Unlike increasing, when the slider is lowered the values are lowered linearly.

Brightness node with a value of 0.1. | Brightness node with a value of 0.5. | Brightness node with a value of 1.4.

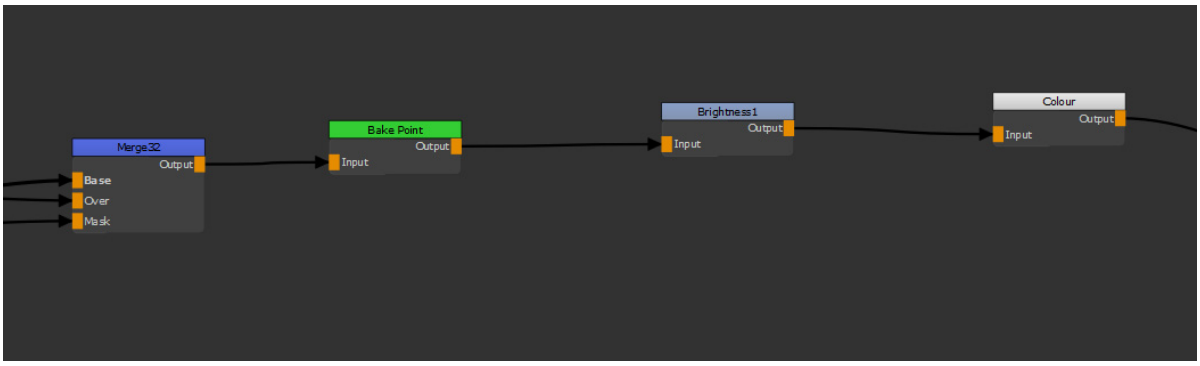
Brightness Node Workflow Example

Addressing Feedback with the Brightness Node

The Brightness node is a great way to make quick tweaks to your textures if they are too dark. Since it's a node with a single slider it's easy to make adjustments. If you only want to make a single change, there is very little ambiguity to what you are doing when you view the node. This can be important if you are sharing assets or handing your archive over to someone else as you want your project to be simple to understand.

The Base Color Map of this asset.

If you have a Base color channel that has been exported for look development, but it needs to be brighter, you can put down a Brightness node at the end of your node tree for a quick change.

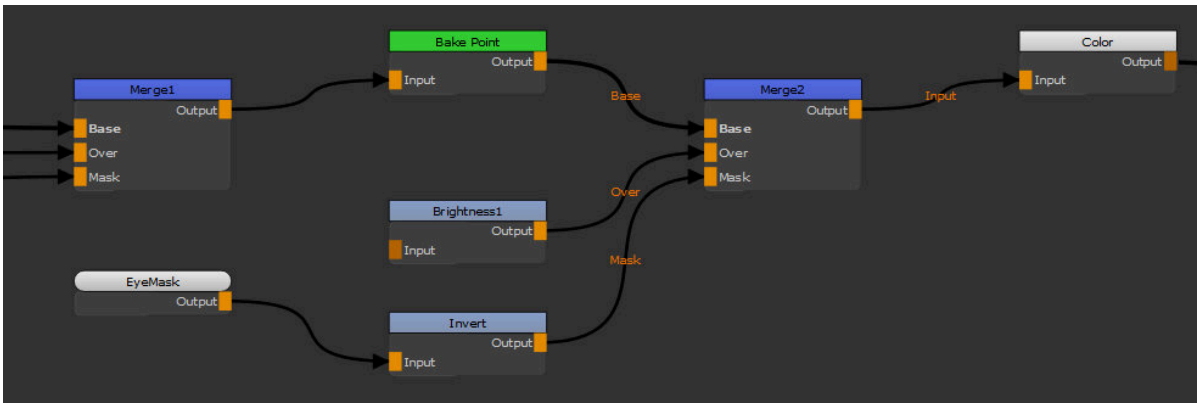


A Brightness node near the end of a node tree.

Just laying down the Brightness node however, would lighten all of the textures. In the example image, there is a light piece of eye geometry in comparison to the rest of the object. The mask which has already been set up for the eye can be used so it is not affected by the Brightness node.

A mask isolating the eye geometry that doesn't need to be brighter.

You can add a Merge node and plug the Brightness node into the **Over** input and the unadjusted texture into the **Base** input. Then by inverting the mask of the eye you can tell Mari to use the brighter version everywhere but the eye.



Setting up the inverted mask and brightness.

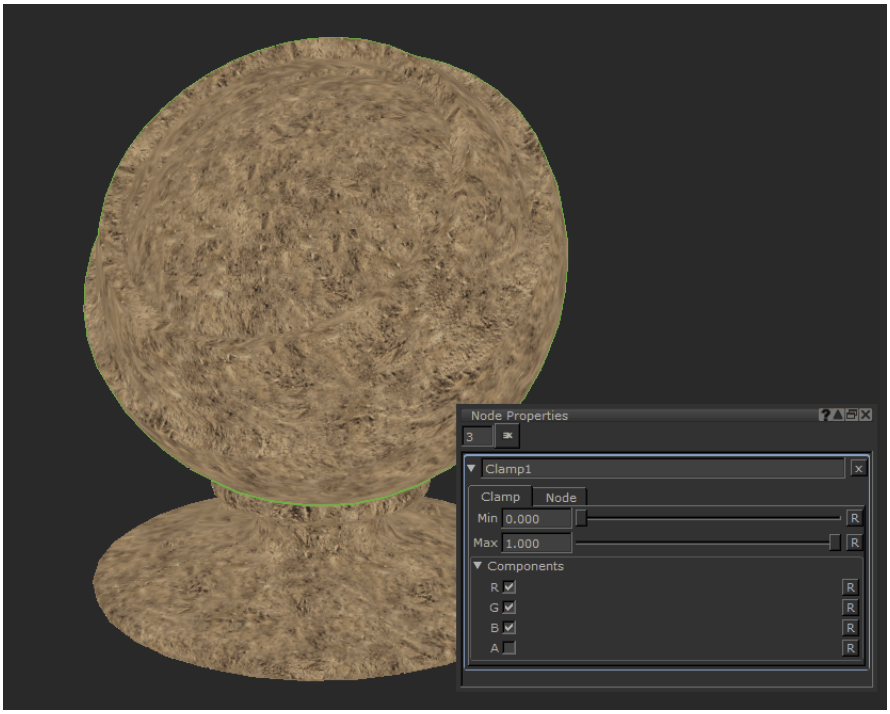
By plugging the Merge node into the **Color** channel, you can export the result and get it to the next person in the pipeline.

Clamp Node

Access: **Nodes** > **Filters** > **Clamp**

The Clamp node is often used for tech fixes and debugging issues in a scene. The node will take values outside of the **Min** and **Max** that you set and clamp them to that range. This is useful if you have erroneous values, for example data going over 1 which are causing you errors.

Tip: Some nodes, such as Levels, include clamping options.



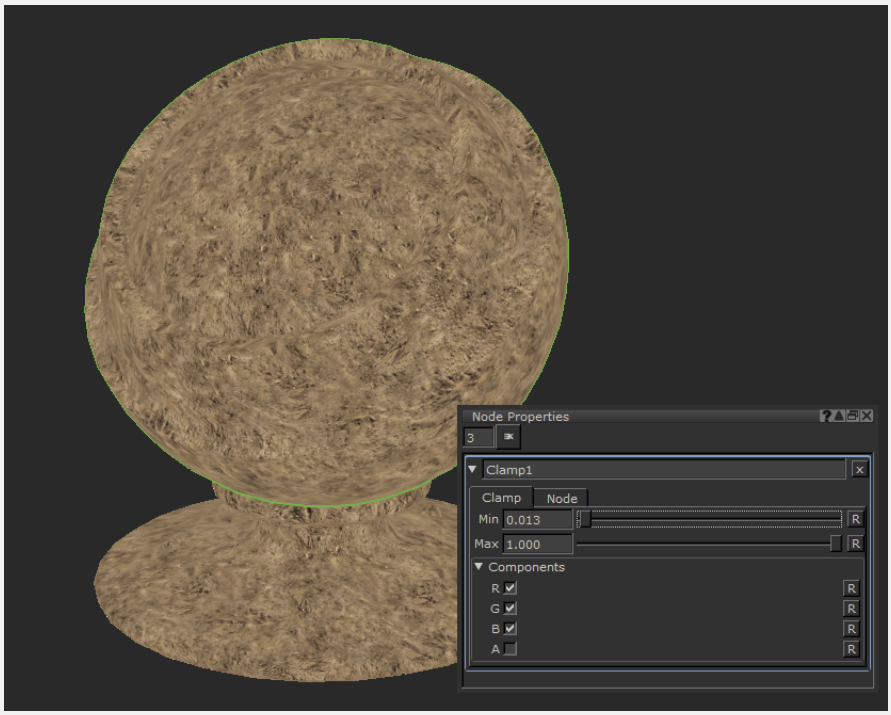
Using a Clamp node to clamp down the max value of this texture

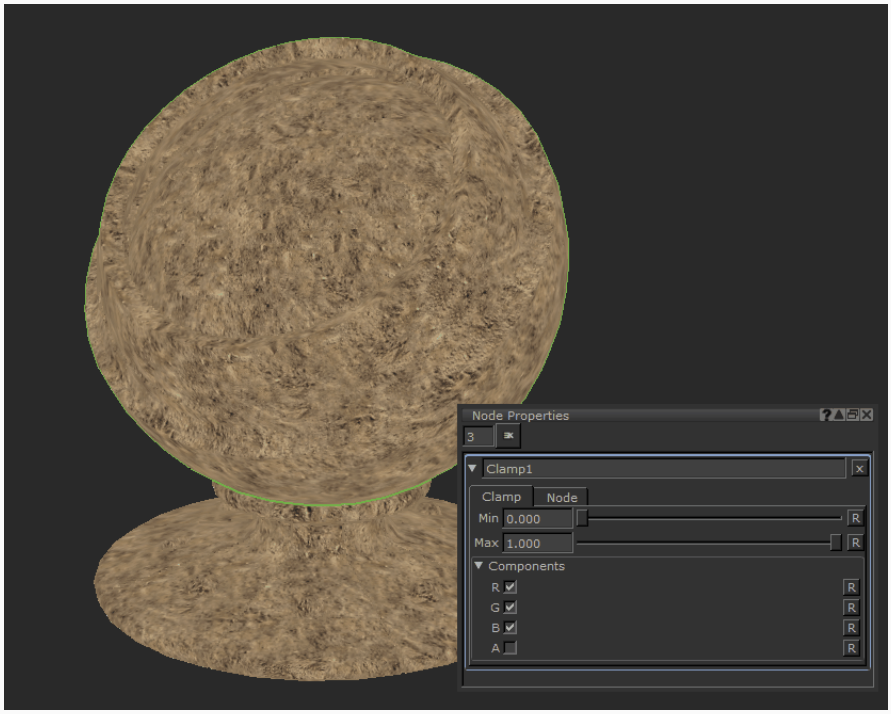
By default the node will clamp from 0-1, visually causing so change, but you can also play with the properties to remove values, say for example you had a bump map with too many peaks, you could use a clamp to bring them down to a more standardized value.

Clamp Node Inputs

Input: Tells the node what the Clamp node will change and adjust.

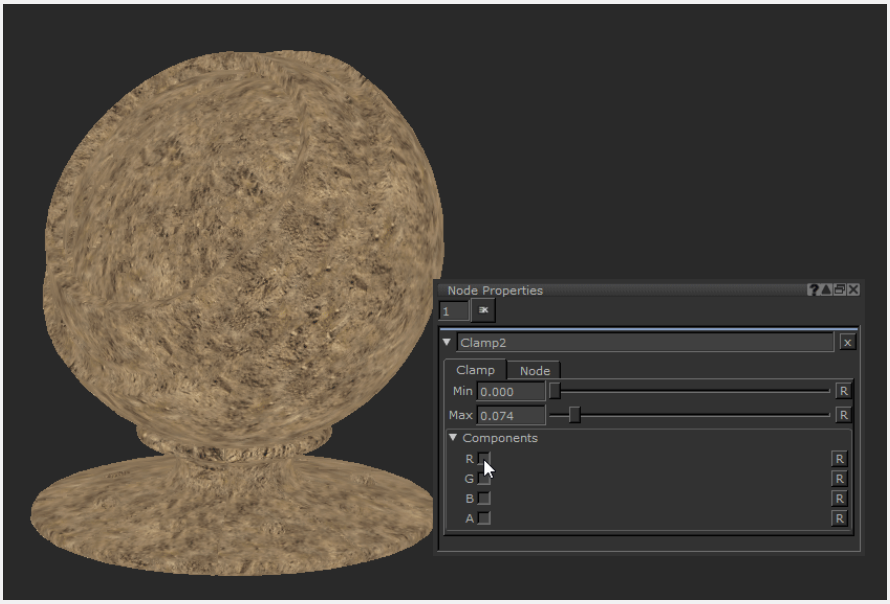
Clamp Node Properties

Min *text field, slider* Default: 0.00	Defines the bottom end value to clamp. Values below the minimum are raised to the **Min** value, causing blacks to be lifted and appear grey. 
Max *text field, slider* Default: 1.00	Defines the top end value to clamp. Values above the maximum will be lowered to the **Max** value, causing whites to be lowered and appear gray.



Components

Red, Green, Blue, Alpha *checkbox*	By enabling or disabling the checkboxes you tell the node which color channels of your Input you want to adjust.



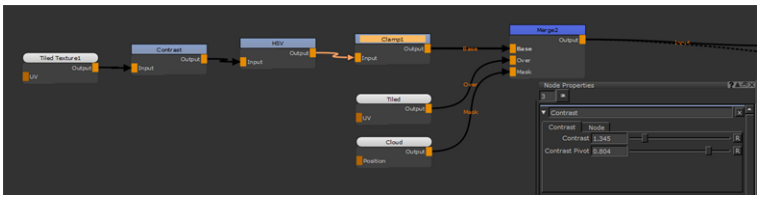
By default, adjustments affect the red, green and blue channel, but if you wanted to include the alpha or disable any of the others you can do so with these checkboxes.

Clamp Node Workflow Example

Example - Fixing Erroneous Values with the Clamp Node

Unexpected results from a Merge node can be the result of data outside of the 0-1 range. Depending on the blend mode math of the merge, you may be seeing results that don't seem right.

In the example, the tiled texture is contrasted to a value of 1.3 using a high pivot of 0.8. This is causing some values to go over 1.



The values overlay another tiled texture using a Cloud node as the mask. However, the merge does not appear to be working properly and still merging in areas that are not white in the mask. There is also unexpected extra contrast added to the image. In the image below, after using a Clamp node, the final output texture is different.

It is not always obvious that your values are going over 1, but if you are doing some major color corrects, placing a Clamp node can tell you if the image is doing any operations incorrectly.

Color Mask node

Access: **Nodes** > **Filter** > **Color Mask**

The Color Mask node enables you to create masks from a specific color in your image. This is useful if you want to only apply an effect or color to a specific part of a model using procedural control.

The node works by calculating how close the input image is to its Mask Color property in Hue, Saturation and Value, and combines them into a single output mask.

Color Mask node inputs

• **Input** — The image/texture you want to apply the color mask to.

Color Mask node inputs

• **Output** — Outputs the mask.

Color Mask node properties

Mask Color	*Color picker*	The color to generate a mask for.
HSV Strength	*Slider, textbox*	How strong the mask is for similarity in hue, saturation and value respectively.
HSV Tolerance	*Slider, textbox*	How close the input color should match the Mask Color in hue, saturation and value respectively.
Invert	*checkbox*	When checked, inverts the final mask output .
Contrast		

Gain	*Slider, textbox*	Adjust the gain of the final mask.
Bias	*Slider, textbox*	Adjust the bias of the final mask.

Color Remap node

Access: **Nodes** > **Filter** > **Color Remap**

The Color Remap node provides color grading from a single space, so you don't have to jump between different nodes to achieve color matching.

Color Remap node inputs

• **Input** — The nodes you want to color grade.

Color Remap node properties

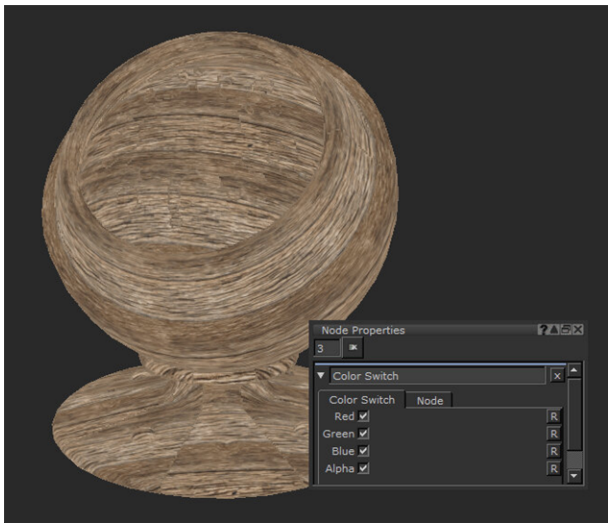
Color		
Gamma	*Slider, textbox*	The amount of Gamma correction to apply to the input. Lowering the slider below 1 darkens the midtones. Raising it above 1 lightens the midtones.
Saturation	*Slider, textbox*	A modifier to the amount of saturation the output has. Moving the slider to the left down to 0 will turn your input into a greyscale texture, where as sliding it right will make the texture more saturated.
Hue Offset	*Slider, textbox*	Changes the hue of the input.
Contrast		

Contrast	*Slider, textbox*	Adjusts the difference in luminosity between light and dark areas.
Contrast Pivot	*Slider, textbox*	Sets the point around which Contrast is adjusted.
Value		
Gain	*Slider, textbox*	Adjusts the gain of the image by using a scale node. Result is a linear multiplier.
Exposure	*Slider, textbox*	Brightens or darkens color in stops. Positive values increase brightness, negative values decrease it. Each stop doubles or halves the color value.
Clamp		
Clamp output	*Checkbox*	Check this to apply a clamp as a safety net for the output.
Min	*Ranged slider*	The min value for the clamp.
Max	*Slider, textbox*	The max value for the clamp.

Color Switch Node

Access: **Nodes** > **Filter** > **Color Switch**

The Color Switch node allows you to switch channels on and off, removing them from the output. This is useful if you have images that you want to display in single channel. For example making a mask just red, green, or blue, so you can combine multiple channels into a single texture.



Enabling all channels on the Color Switch properties with a Tiled node.

Color Switch Node Inputs

Input: Removes color channels from the input.

Color Switch Node Properties

Red, Green, Blue, Alpha *checkbox*	Enabling or disabling the checkboxes determines which color channels you want to keep and remove. By default the all channels are enabled.

Color Switch Node Workflow Example

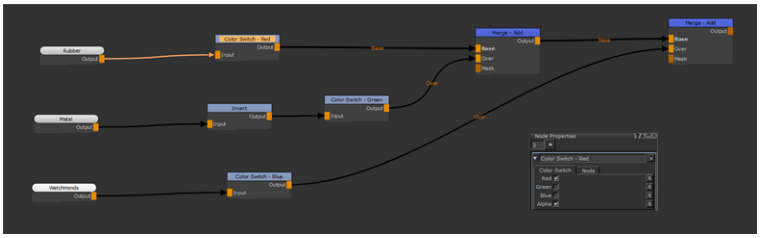
Color Switching Masks into a Single Texture

One use of the Color Switch node is packing multiple masks into a single image. Some pipelines require 'channel packed' masks and other grayscale images to optimize scene performance.

The example shows three masks from a watch asset that need to be output for render IDs. There is a mask for the rubber strap, a mask for the metal of the watch, and the third mask is the clock hands. Masks for different areas enable look development artists to tweak different parts of an object or render.

To pack these masks into a single image, you need to put the three grayscale images into the R, G, and B channels of an image. Using a Color Switch node allows you to do that.

The Color Switch node allows you to strip out the channels of an image you don't need, leaving you with a single color that you can blend together.



Using a Color Switch node means you can turn each mask into pure red, green, and blue, and then merge them together. You need two Merge nodes in order to achieve this, both with the blend **Mode** set to **Add**. The values of the **Base** and **Over** inputs are added together and produce this result:

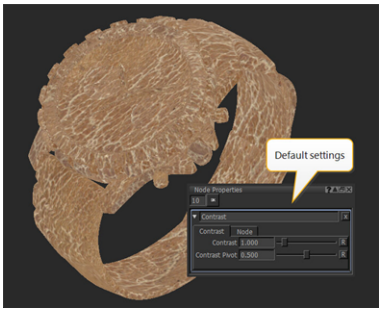
Since the inputs are only in a single channel when this map is exported and used for Render, you can split the channels out in your shader setup, using either the R, G, or B of the image for different masks. You can achieve this with any single-channeled map, such as Bump, Specular Roughness, or Metallic maps.

To see how to deconstruct a channel packed texture, see Shuffle Node.

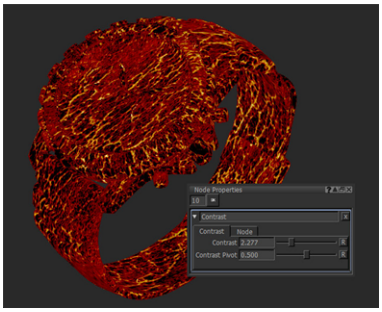
Contrast Node

Access: **Nodes** > **Filter** > **Contrast**

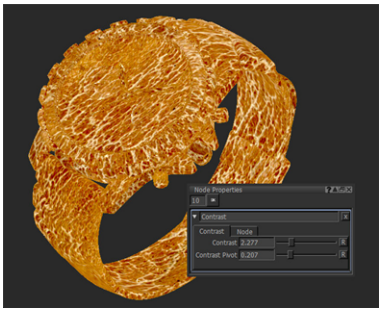
The Contrast node is a filter node that adjusts the input data by linearly adjusting the difference in luminosity between the light and dark areas of a source texture.



Contrast node's default settings



*Contrast node's increased **Contrast***



*Contrast node's increased **Contrast**, decreased **Contrast Pivot***

Contrast Node Inputs

Input - Any nodes containing textures or paint that need to be altered can be connected to the Contrast node, for example, Tiled and Paint nodes.

Contrast Node Properties

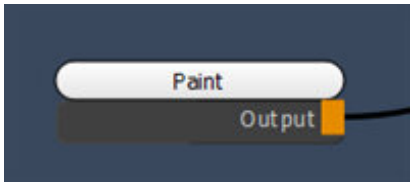
Contrast *floating point control*	Adjusts the difference in luminosity between light and dark areas.
Contrast Pivot *floating point control*	Sets the point around which **Contrast** is adjusted.

Contrast: Node Graph Workflow Example

Example - Achieving Contrast Operation

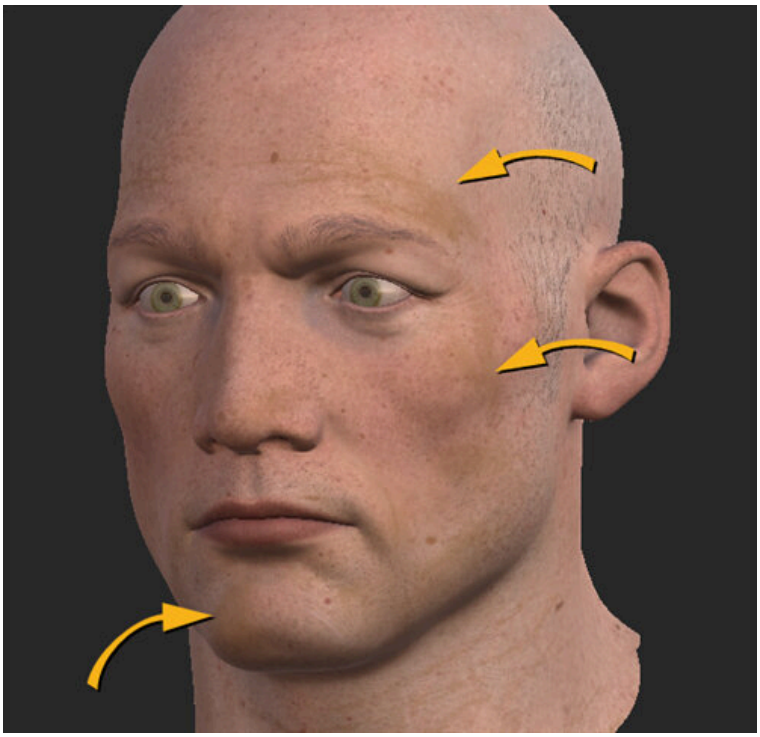
In the example below, let's use a Contrast node to emphasize the dark brown pigments of the model.

1. Add a Paint node.



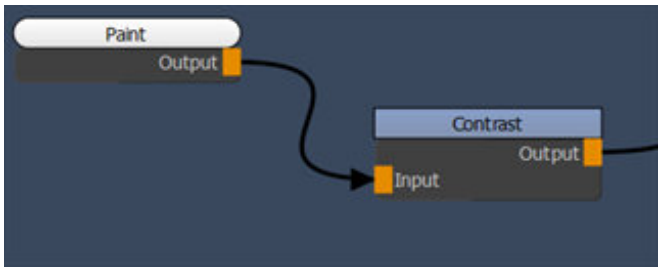
Adding a Paint node.

This is the original texture.



The model has some dark brown pigments on his face.

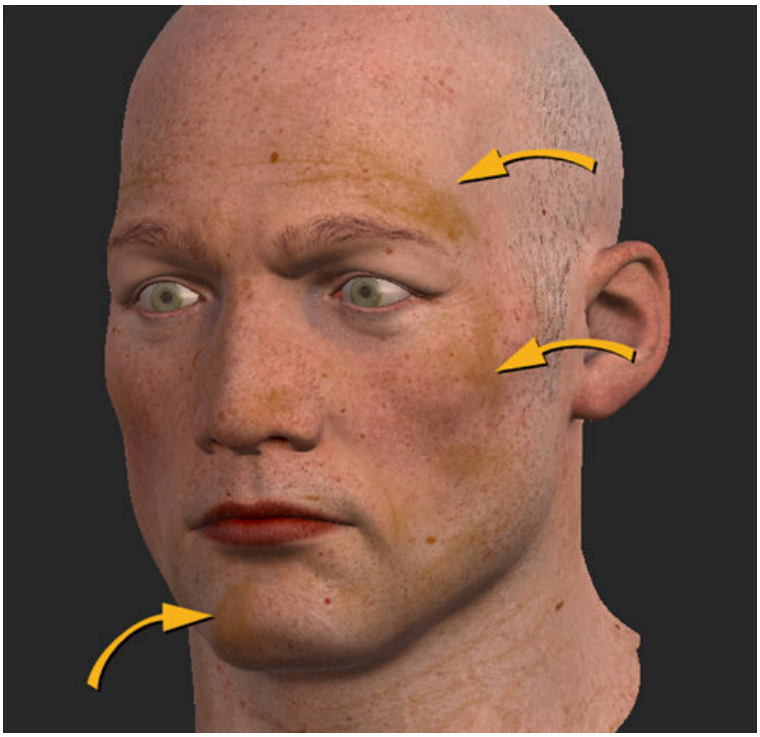
2. Create a Contrast node.
   Connect the Paint node to the input of the Contrast node.



Adding a Contrast node to the Paint node output.

3. Open the node **Properties** of the Contrast node and set the **Contrast** parameter of the Contrast node to **1.200**.

   Notice how the dark brown pigments now look darker.



*Contrast node's **Contrast** set to **1.200**.*
The dark brown pigments now look darker.

Multi Color Mask node

Access: **Nodes** > **Filter** > **Multi Color Mask**

Multiplicative color masks bundled up in a group. Works the same as the Color Mask node.

Multi Color Mask node inputs

• **Input** — The image/texture you want to apply the color mask to.

Multi Color Mask node inputs

• **Output** — Outputs the mask.

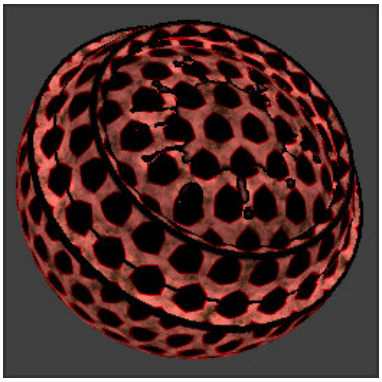
Multi Color Mask node properties

Add Color Mask	*button*	Adds a new additional Color Mask and returns the maximum values of all other Color Masks.
Remove Color Mask	*button*	Removes the Color Mask from the group.
Mask Color	*Color picker*	The color to generate a mask for.
HSV Strength	*Slider, textbox*	How strong the mask is for similarity in hue, saturation and value respectively.
HSV Tolerance	*Slider, textbox*	How close the input color should match the Mask Color in hue, saturation and value respectively.
Invert	*checkbox*	When checked, inverts the final mask output.
Contrast		
Gain	*Slider, textbox*	Adjust the gain of the final mask.
Bias	*Slider, textbox*	Adjust the bias of the final mask.

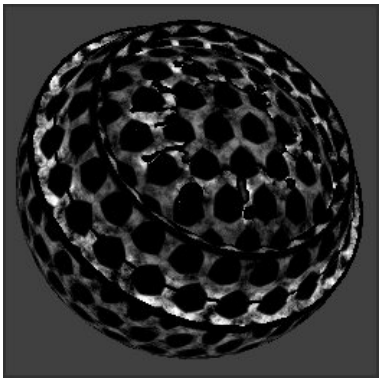
Smart Mask Node

Access: **Nodes** > **Misc** > **Smart Mask**

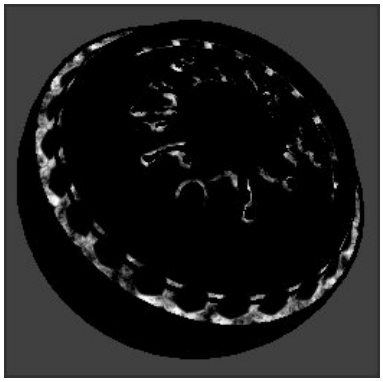
The Smart Mask node allows you to create textures within the node, which can be exported as a procedural texture and added to the **Shelf** for use in other projects, the node graph, or layer stack. The node can be exported as a Color Procedural, a **Scalar Procedural**, or a **Procedural Mask**.



A color procedural in the **Shelf**.



A scalar procedural in the **Shelf**.



A procedural mask in the **Shelf**.

The strength of the Smart Mask node compared to other Group nodes is that it exports all the images used inside the node. This means when the procedural is shared with other artists, they have access to the images used in the textures and don't have to locate them.

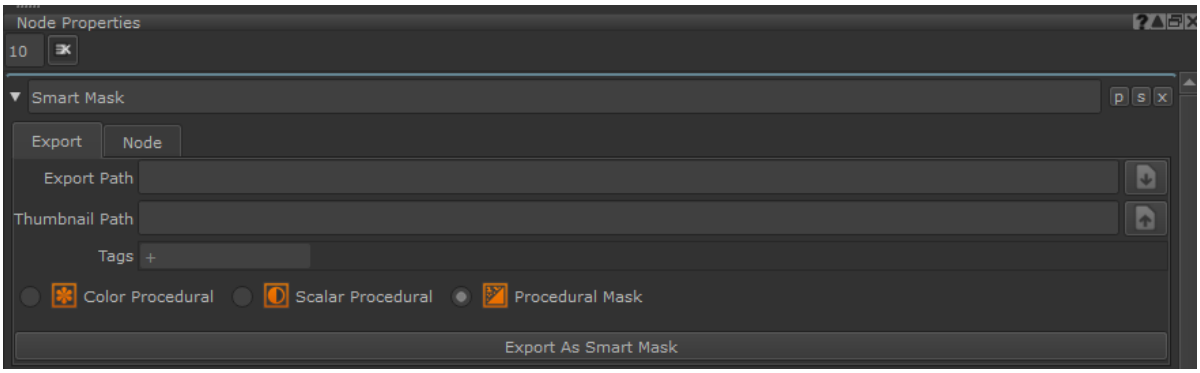
There are a variety of example Smart Masks for you to use in the **Smart Masks** shelf. This includes basic masks which utilize maps such as Ambient Occlusion, and a variety of grunge maps in order to easily achieve looks such as dirt, dust and damage. These can be helpful as a starting point, which you can then adjust to work for you. In the Smart Mask shelf you'll also find a set of utilities, such as warp and spot stainer, which can help with refining your textures. For a full overview of the content that is available, see Ready-to-use Shelf Content.

Note: The Smart Mask node used to be called the 'Custom Procedural' node.

Note: You must use the Node Graph to produce Smart Masks. Smart Masks cannot be created using the **Layers** palette. See Adding New Smart Mask Nodes to the Node Graph.

Tip: The Smart Mask node can also be used to create procedurals that can be used in the layer stack to share with artists who prefer a layer based workflow. The layer stack is one of the most intuitive ways to create your textures, see Layers for more information.

Smart Mask Node Properties



The Smart Mask **Node Properties** panel.

Export Tab

Export Path	Specify a file path where the Smart Mask is saved on export.
Thumbnail Path	Pick a thumbnail image from disk to use with the Smart Mask. This thumbnail represents the procedural in the **Shelf**. If no thumbnail is chosen at export time, a thumbnail is generated automatically using the **Thumbnail** settings in the **Preferences**. See Thumbnail Tab for more information.
Tags	Add metadata tags to your material if required. These tags can be used to display information about the procedural, or for filtering procedurals in the Shelf. Start typing a description into the field to display a list of existing tags or create your own and press **Enter** to add the tag. Once the procedural is exported and added to the **Shelf**.
Export as Smart Mask	Exports the Smart Mask to the **Export Path** as an **.mpc** file.

The Smart Mask node allows you to export three types of procedural for use in other projects:

Color Procedural	Color procedurals are treated as color data when imported into other projects, and are color managed by Mari. This means the procedural can be used in channels which are color managed, such as diffuse, albedo, or base color channels. Select this if, for example, you want to create a tiled color map.

Scalar Procedural	Scalar procedurals are treated as scalar data (non-color data) and are not color managed by Mari when used in other projects. Select this if you want the smart mask to be used in channels using scalar data such as specular, roughness, or normal channels.
Procedural Mask	Procedural masks are similar to scalar procedurals, in that they are also treated as scalar data and won't be color managed by Mari. Unlike the other procedural types, procedural masks can also be applied as a layer mask in the layer stack. Select this if you want your procedural to be used as a mask in other Mari projects.

Any Smart Masks shipped by Mari that you use from the Shelf will have two additional tabs:

Smart Mask Tab

Note: These options will adjust based on which smart mask you are using.

This tab allows you to change the quality of the driver mesh map being used, including: Axis, Debug, Input and Output settings. For example, you can control the contrast of the Ambient Occlusion map used without the need to dive into the node graph.

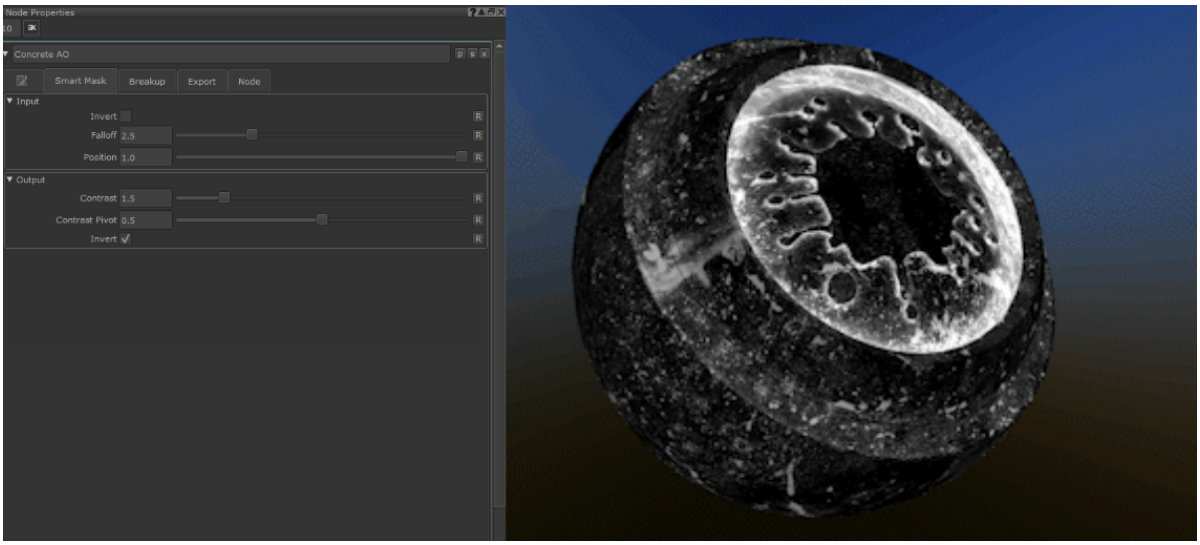
See more in the Smart Masks section of the Ready-to-use Shelf Content page.

Breakup Tab

Note: These options will adjust based on which smart mask you are using.

All of the Smart Masks have a noise or a grunge map assigned. You will find a Texture map file, used to provide breakup for the mesh map. In the Breakup tab, you can control the scale, size, rotation of the texture or even switch to TriPlanar projection.

See more in the Smart Masks section of the Ready-to-use Shelf Content page.



Smart Mask Workflow Overview

Smart Masks are created in the Node Graph using Smart Mask nodes. The node is a container for your own custom node network. You add a network by adding the Smart Mask node to your Node Graph and opening the node. To learn how to add Smart Mask nodes to your Node Graph, see Adding New Smart Mask Nodes to the Node Graph.

You can create any texture you like inside the Smart Mask node, since all Node Graph functions and nodes are available to you. As an example, f you would like to learn how to create an example grunge mask using geo-channels inside a Smart Mask node, see Creating a Smart Mask Grunge Mask with Geo-Channels.

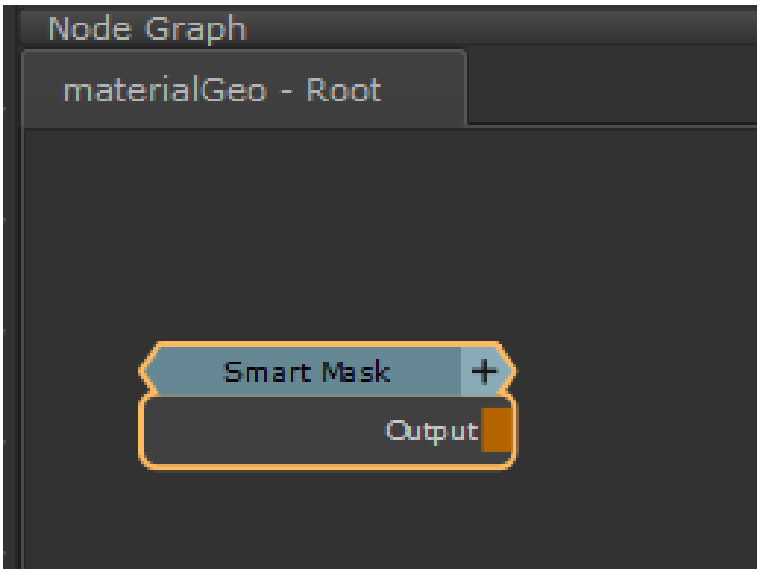
Once you have created a Smart Mask using the Smart Mask node, you can export the procedural using the **Node Properties**. The procedural can then be used in other projects and by other artists when imported to the **Shelf**. See Exporting Smart Masks and Importing Smart Masks to the Shelf.

To use the Smart Mask in other projects, add the Smart Mask from the **Shelf** to any node graph or layer stack. See Adding Smart Masks to the Node Graph and Adding Smart Masks to the Layer Stack.

Adding New Smart Mask Nodes to the Node Graph

1. Right-click anywhere in the Node Graph and select **Nodes** > **Misc** > **Smart Mask**.

   A **Smart Mask** node is added to the Node Graph.



The **Smart Mask** node on the **Node Graph**.

2. With the **Smart Mask** node selected, press **1**.

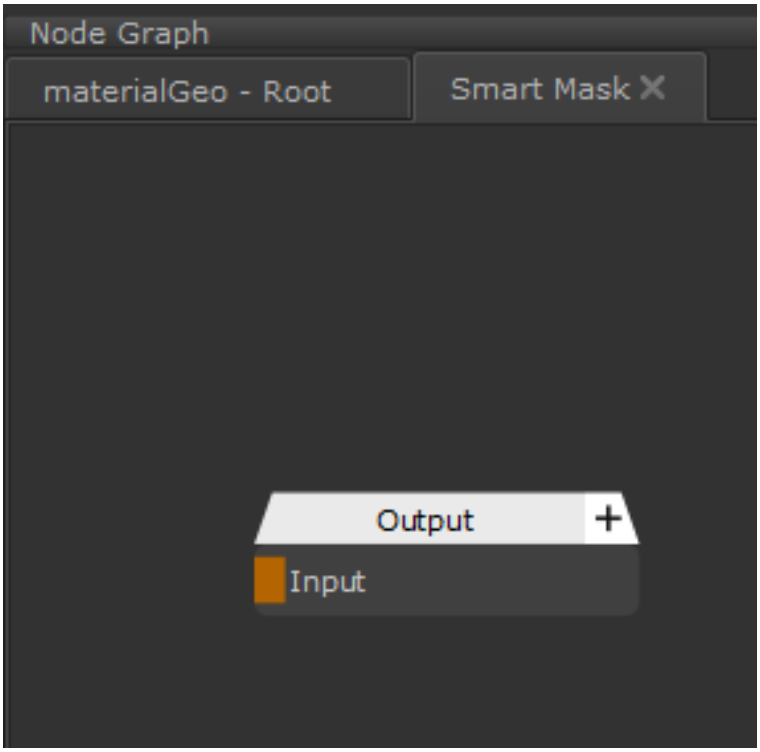
   The node is assigned to the **Viewer**, so you can see the node in the viewport on your model.

3. Hold **Ctrl** and double-click the **Smart Mask** node,

   or

   press **Ctrl** + **Enter** with the Smart Mask node selected.

   A new tab opens in the Node Graph and the contents of the **Smart Mask** node are displayed. The Smart Mask is authored in this tab.



Inside the Smart Mask node in the **Node Graph**.

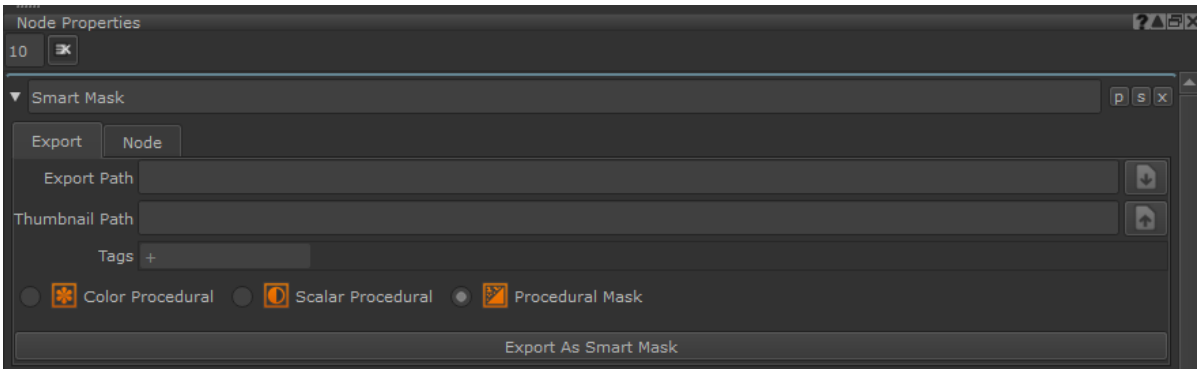
Exporting Smart Masks

Once you have authored a smart mask using the Smart Mask node, you can export it as an **.mpc** file to be used in other projects through the **Shelf**.

To export your Smart Mask:

1. Double-click the Smart Mask node containing the node network you'd like to export.

   The **Node Properties** open.

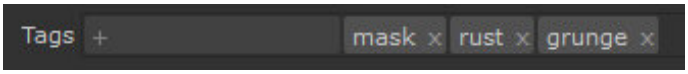


The Smart Mask **Node Properties**.

2. Next to **Export Path**, click the export button to choose an export location for your Smart Mask.
3. Click the choose thumbnail button next to **Thumbnail Path** to choose a thumbnail image from disk.

   If you don't specify a **Thumbnail Path**, Mari creates one for you before the Smart Mask is exported.
4. If required, add metadata tags to your Smart Mask into the **Tags** field. Start typing a description into the field to display a list of existing tags or create your own and press **Enter** to add the tag. See Tags for more information.

   Once the Smart Mask is exported and added to the **Shelf**, these tags can be used to display information about the procedural, or for filtering procedurals in the **Shelf**.



5. Select **Color Procedural**, **Scalar Procedural**, or **Procedural Mask** to specify a type for your Smart Mask.

   The type determines how your Smart Mask is color managed in the project. See Smart Mask Node Properties for more information.
6. Click **Export as Smart Mask**.

   Your Smart Mask is exported as an **.mpc** file to the specified **Export Path**.

Importing Smart Masks to the Shelf

Once your Smart Masks are exported, you can import them to the Mari Shelf for use in other projects, node graphs, and layer stacks.

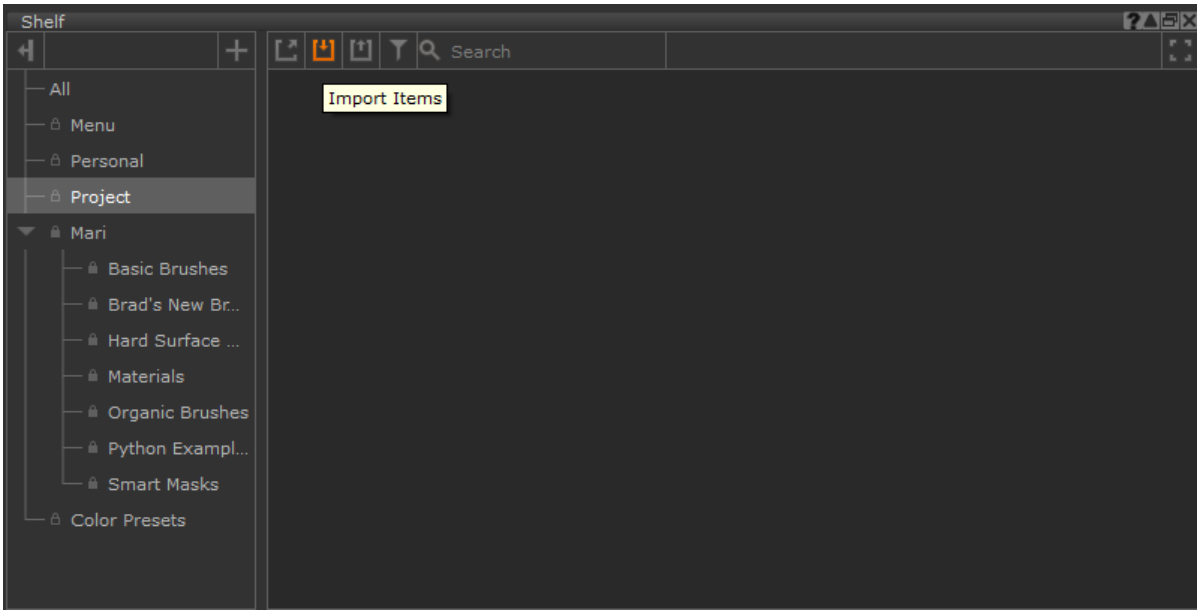
To import your Smart Mask:

1. Click **Shelf** on the palette toolbar.

   This opens the **Shelf**.
2. In the left panel of the **Shelf**, select a folder where you would like to add your procedural. **Note:** The Smart Masks shelf is locked, so you will need to use the Personal, Project or a custom shelf.
3. Drag the **.mpc** file from your file browser into the **Shelf**,

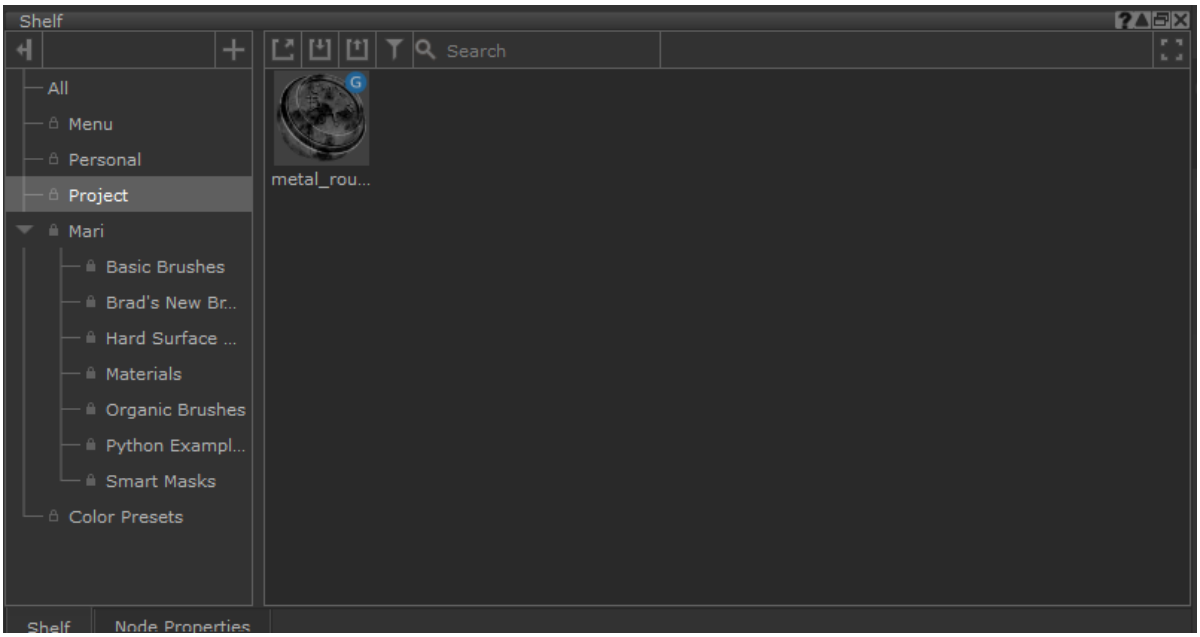
   Or

   Select the **Import Items** button:



4. Select the **.mpc** file from the **Import Items** dialog and click **Open**.

   The Smart Mask is imported to the specified folder in the **Shelf**.



Imported procedural in the **Shelf** palette.

Note: If your Smart Mask shows a geo-channels icon on the thumbnail, it means the Smart Mask contains geo-channel nodes.

Tip: Hover over the thumbnail in the **Shelf** to see more information about the Smart Mask.

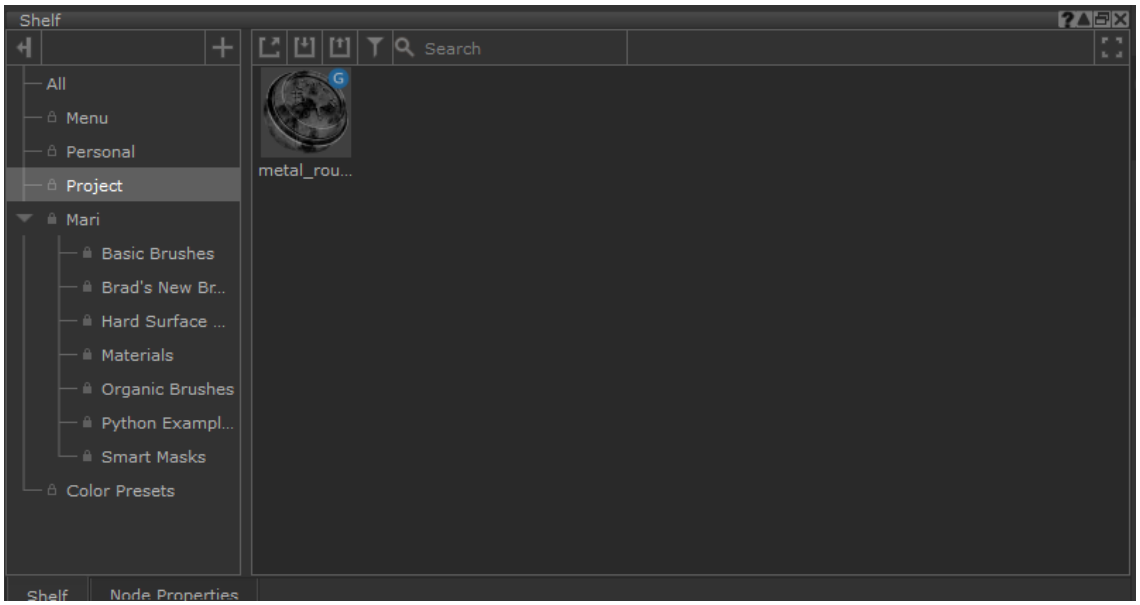
Adding Smart Masks to the Node Graph

Once a Smart Mask has been added to the **Shelf**, you can add it to your Node Graph.

1. Click **Shelf** on the palette toolbar.
2. Click the Smart Masks folder to find Smart Masks shipped by Mari

   or

   select your own procedural from another shelf, eg. Project.

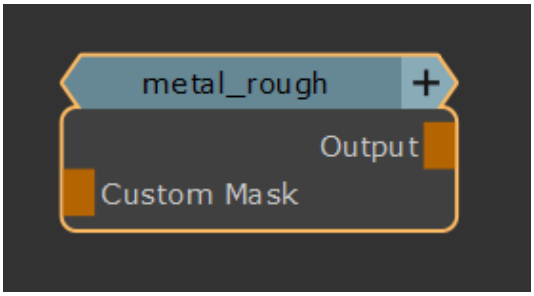


Imported Smart Mask in the **Shelf**.

3. Hover over Smart Masks to see more information.

The information displayed when hovering over an item in the **Shelf**.

4. Drag the chosen Smart Mask from the **Shelf** onto your Node Graph.
   The procedural is inserted into the Node Graph.



Tip: Hold **Ctrl** and double-click the Smart Mask to view the its node network.

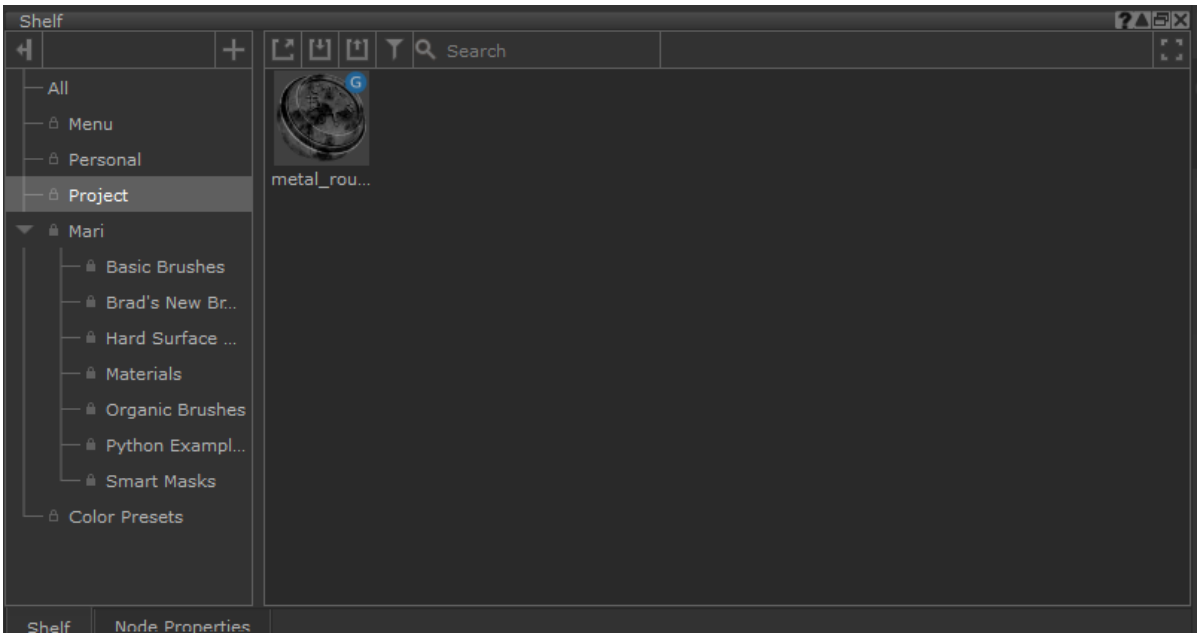
Adding Smart Masks to the Layer Stack

Once a Smart Mask has been added to the **Shelf**, you can add it to your layer stack.

1. To open the Shelf, click **Shelf** on the palette toolbar.
2. Click the Smart Masks folder to find Smart Masks shipped by Mari

   or

   select your own procedural from another shelf, eg. Project.



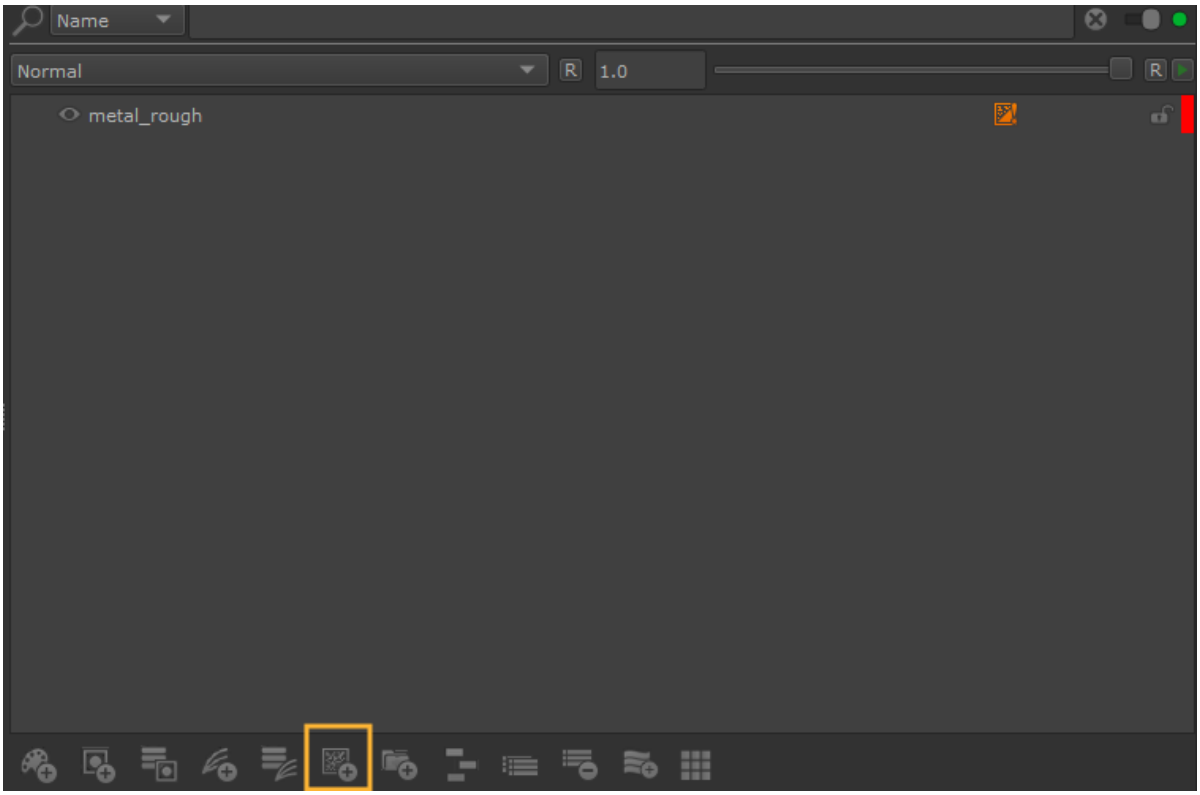
Imported Smart Mask in the **Shelf**.

3. Hover over Smart Mask to see more information.
4. Drag the chosen Smart Mask from the **Shelf** into your layer stack,

   or

   on the **Layers** palette, click the add procedural layer button > **Smart Mask** > select the Smart Mask you want to add to the layer stack.

   The Smart Mask is added to the layer stack.



A smart mask in the layer stack.

5. Click the procedural icon next to the procedural layer to view the procedural's node network.

Smart Mask Node Workflow Example

Creating a Smart Mask Grunge Mask with Geo-Channels

A procedural mask is a mask that can be authored with the Smart Mask node and then imported into projects and applied in the node graph or layer stack. It allows artists to save the time it would take to create a mask manually, and gives them the option to use a pre-generated mask instead. To learn more about masks, see Adding Layers and Masks.

You can export a procedural mask that be used by any object with geo-channels in Mari by taking advantage of Geo-Channel nodes while authoring your textures with the Smart Mask node. The exported procedural can be used in any node graph, or added as a layer mask on any layer providing you have Geo-Channels setup for your object. See Adding Geo-Channels to Objects and Geo-Channel Node for more information.

In the following example, you can create a grunge mask that could be used to show and hide grunge on a layer or node, for example if you wanted to mask off some metallic flooring to apply a rust layer around the edges.

Tip: Use the **Example Material Design Project** inside the **Help** menu to follow the below example, as this project has geo-channels already setup for you to use.

Stage 1 - Creating a Procedural Cloud Mask

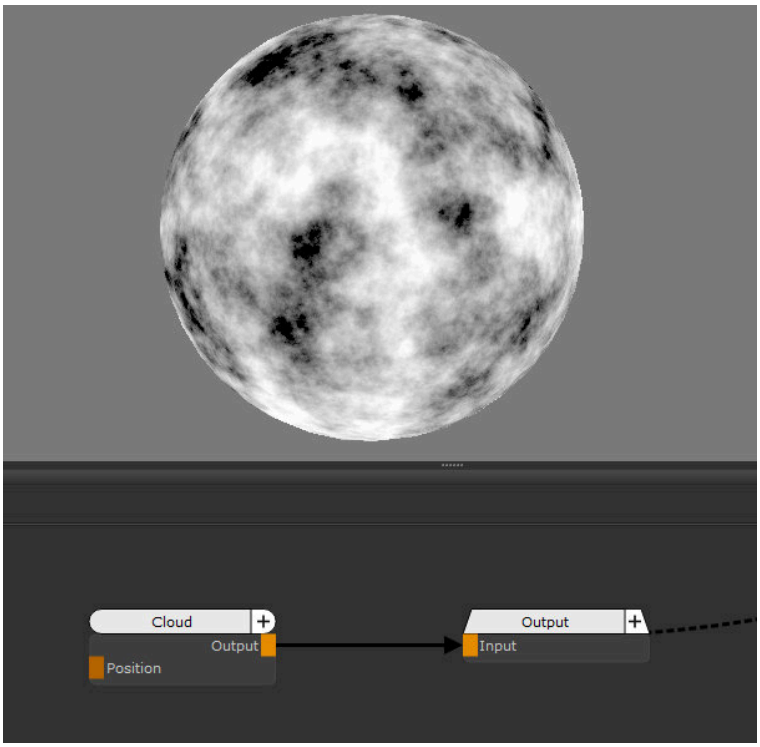
Adding Cloud nodes to your node network is a quick and useful way to create grime-like textures. The Cloud node is going to be the base of this procedural texture. To create a procedural cloud mask using a Cloud node, while inside the Smart Mask node:

1. Press **Tab** and type **Cloud**, then select the resulting **Cloud** node.

   A Cloud node is added to the Node Graph. The noisy cloud texture forms the basis of the grunge texture.

2. Drag the Cloud node's **Output** to the Output node's **Input**.

   The Cloud node is output to the model.



The **Cloud** node displayed on the model.

3. Press **Tab** and type **Color**, then select the resulting **Color** node.

   A Color node is added to the Node Graph. This Color node is going to be merged with the Cloud node and a geo-channel mask. The Cloud node shows in any areas of the geo-channel mask that are white. The Color node shows in any areas of the geo-channel mask that are black.

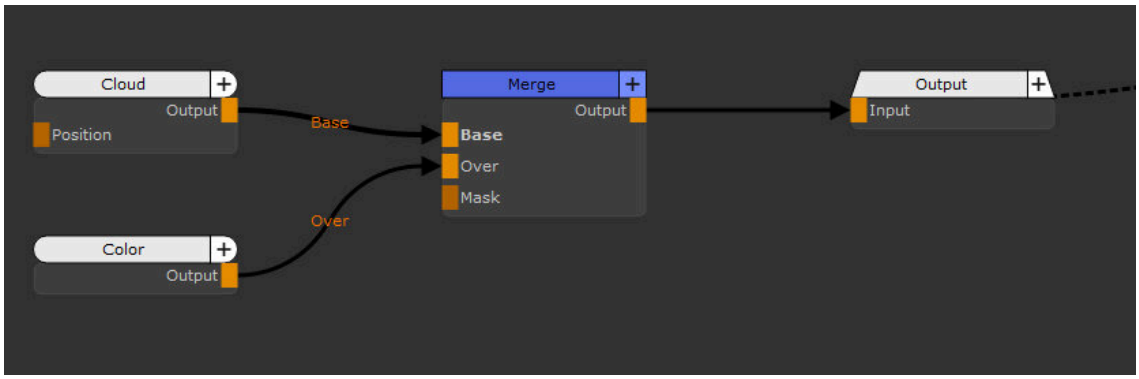
4. Double-click the **Color** node and set the **Color** in the **Node Properties** to black.

The Color node must be set to black so the black areas in the final mask texture are transparent and show any layers or nodes below the current layer.

5. Press **M** over the Node Graph to create a Merge node, and input the **Cloud** node into the **Base** input and the **Color** node into the **Over** input.

   The Cloud and Color nodes are merged together.

6. Drag the Merge node **Output** to the **Input** of the Output node.



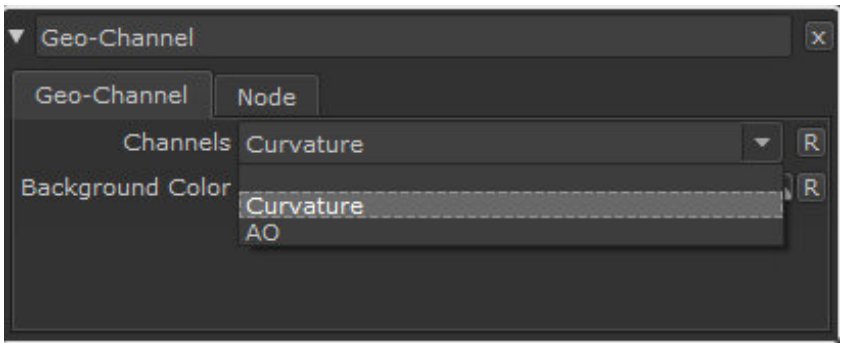
Cloud, Color, and Merge node inside the Smart Mask node.

Stage 2 - Adding Geo-Channels to your Procedural Cloud Mask

The next stage of the workflow deals with adding geo-channels to the node network. Geo-channel nodes point the texture at geo-channels on your object. In this example workflow, the geo-channel is used to drive curvature data from the model into the texture, so the grime appears on any sharp edges of your model.

1. Press **Tab** and type **Geo-Channel** and select the resulting **Geo-Channel** node.
2. Double-click the Geo-Channel node and pick any geo-channel to use with the node from the **Channels** dropdown in the **Node Properties**. If you're using the Example Material Design Project, select the **Curvature** channel.

Note: If you need to add geo-channels to your object, see Adding Geo-Channels to Objects.



The **Channels** dropdown of the Geo-Channel node.

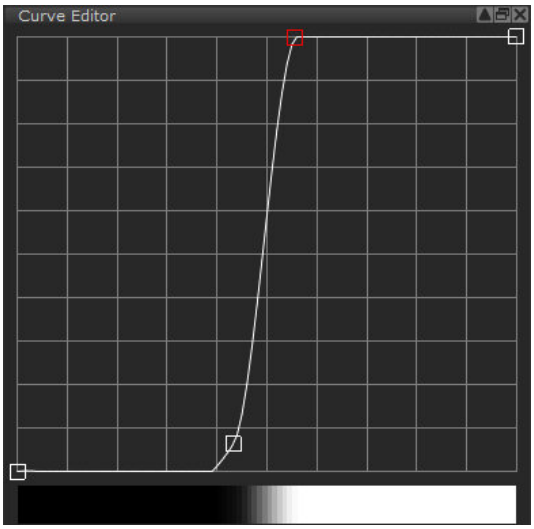
3. Press **Tab** and type **Brightness Lookup** and select the resulting **Brightness Lookup** node.

   The Brightness Lookup node is going to be used to tighten the black and white values from the Geo-Channel node and reduce the mid-tones. This produces a mask that is restricted to the darker areas in the geo-channel.

4. Double-click the Brightness Lookup node and click the curve editor button in the **Node Properties** to open the **Curve Editor**.

   The **Curve Editor** allows you to use the curve to edit the brightness of the input.

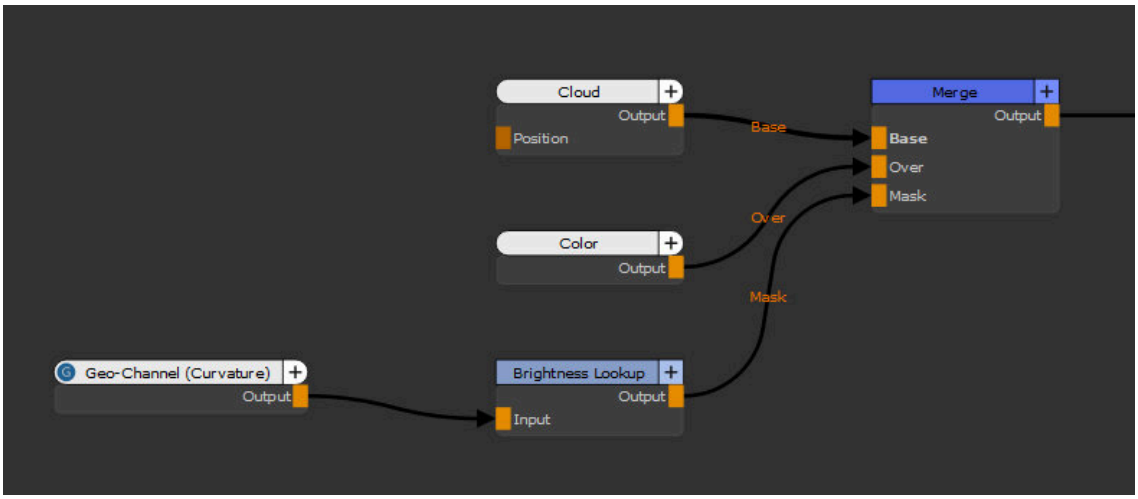
5. Drag the handles in the **Curve Editor** to create a curve which reduces the mid-tones of the input. Use the image below as a reference.



The Brightness Lookup **Curve Editor**.

6. Drag the **Output** of the Geo-Channel node into the **Input** of the Brightness Lookup node, then drag the **Output** of the Brightness Lookup node into the **Mask** input of the Merge node.

   The Geo-Channel node is now used as a mask in the Merge node to determine where in the Merge node the Cloud node is shown, and where the Color node is shown.



A Cloud node, Color node, and Geo-Channel node input to the **Base**, **Over**, and **Mask** inputs of the Merge node, to create a grunge mask.

A grunge mask created with geo-channels, which can be used in other projects.

Exposure Node

Access: **Nodes** > **Filter** > **Exposure**

Brightens or darkens the color of the input texture in stops.

Exposure node inputs

Input — The image/texture you want to apply the color mask to.

Exposure node outputs

Output — The image/texture with the exposure altered.

Exposure node properties

Exposure *slider*	Adjusts the exposure of the input, brightening or darkening the color in stops. Positive values increase brightness, negative values decrease it. Each stop doubles or halves the color value.

Gamma Node

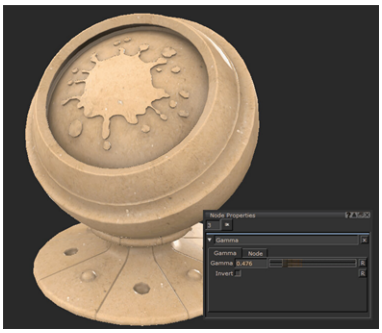
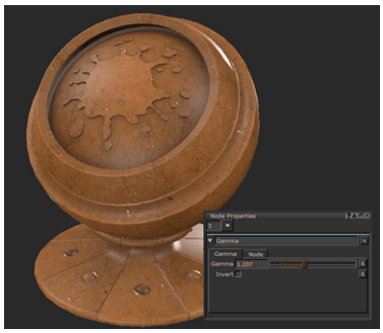
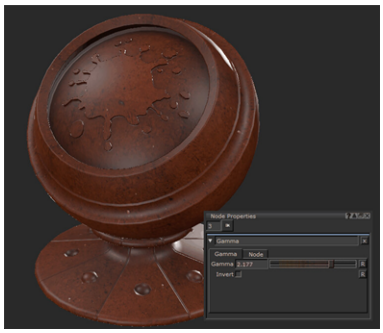
Access: **Nodes** > **Filter** > **Gamma**

The Gamma node is applies a gamma correction to your input. It is a non-linear adjustment, meaning midtones are affected more than the highest and lowest values.

The Gamma node can also be a useful debugging node since color space is linked to gamma. This is because gamma can tell you if incorrect values are due to importing an image in the wrong color space.

A **Gamma** value of 1 does not change the input. By default the Gamma node has a value of 2.2, which is roughly the same as a color space conversion from Linear to sRGB. For more on this, see the Gamma Node Color Space Debugging.

Note: The Grade node contains a **Gamma** slider too.

Gamma adjustments of 0.4, 1.2, and 2.1 on a texture.

Gamma Node Inputs

Input: Informs the node what the Gamma will adjust.

Gamma Node Properties

Gamma *text field, slider* Default: 1.0	The amount of **Gamma** correction to apply to the input. Lowering the slider below 1 darkens the midtones. Raising it above 1 lightens the midtones. The Gamma node properties.
Invert *checkbox* Default: Off	The **Invert** checkbox reverses the gamma operation using the value from the **Gamma** slider.

Gamma Node Workflow Example

Gamma Node Color Space Debugging

You can use the Gamma node for color correction, but it may be more useful to add a Grade node for that instead, since it also has a **Gamma** option as well as more color correcting options.

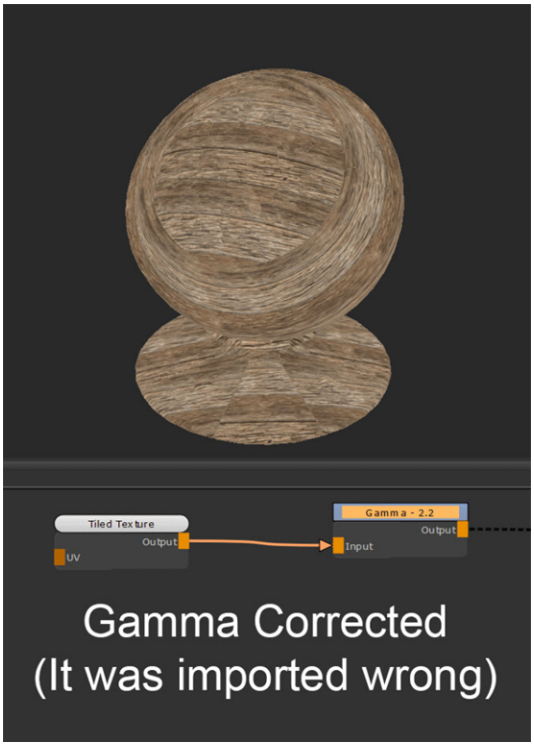
Tip: You can use the Gamma node for debugging.

Imported textures often aren't set to the correct color space when imported into Mari. Adding a Gamma node can tell you if incorrect values are due to importing an image in the wrong color space.

If you have imported a texture into the **Image Manager**, or some images into a Paint node and they look too light, you can add a Gamma node to check if it is a color space issue. If you leave **Gamma** at the default of 2.2 and it looks correct then you know it has imported an sRGB image as RAW or Linear.

Note: A **Gamma** value of 2.2 is roughly the gamma curve of an sRGB image.

An example Tiled Texture after import. It appears to be too light.



The Tiled Texture running through a default Gamma of 2.2 to check color space.

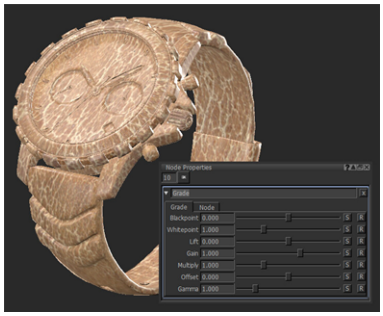
If you import an image and it looks darker than you were expecting you can enable the **Invert** checkbox, with a **Gamma** of 2.2. If that brings the values up to what you were expecting, Mari may have imported a Linear color space image as sRGB. You can fix this through the **Image Manager** or the Paint node properties.

Tip: Ensure you remove the Gamma node after debugging. It is important to make sure the import is corrected after you have debugged the issue.

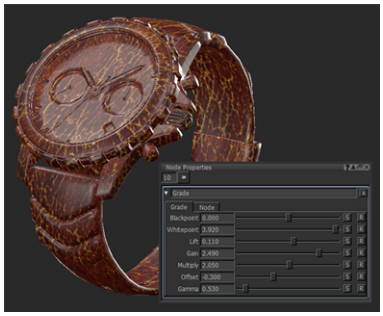
Grade Node

Access: **Nodes** > **Filter** > **Grade**

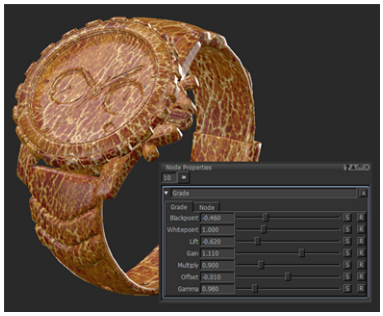
The Grade node is a filter node that adjusts the input data by letting you change the grade of a texture.



Grade node's default settings



*Grade node's increased **Whitepoint**, **Lift**, **Gain**, **Multiply**, and **Gamma**, decreased **Offset***



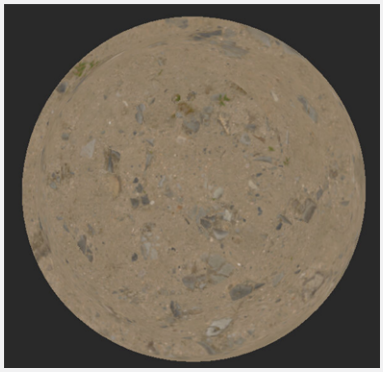
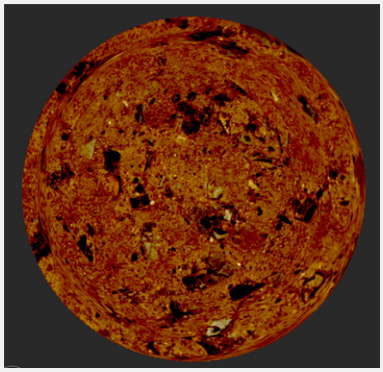
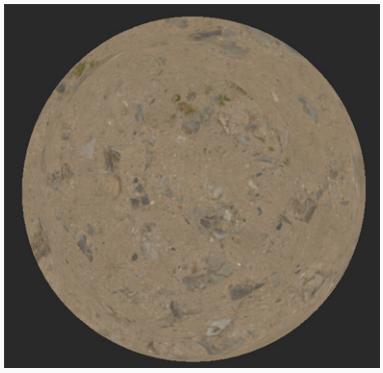
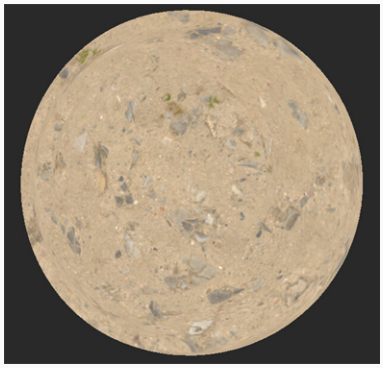
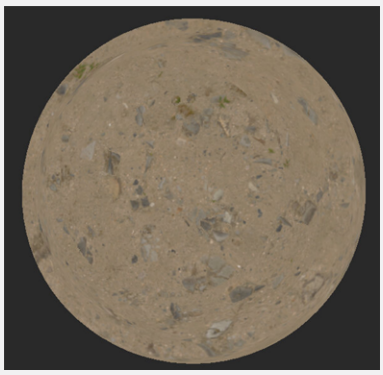
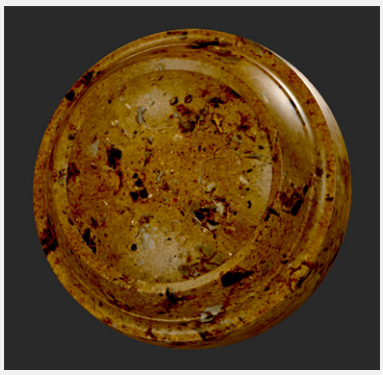
*Grade node's increased **Gain**, **Multiply**, and **Gamma**, decreased **Blackpoint**, **Lift**, and **Offset***

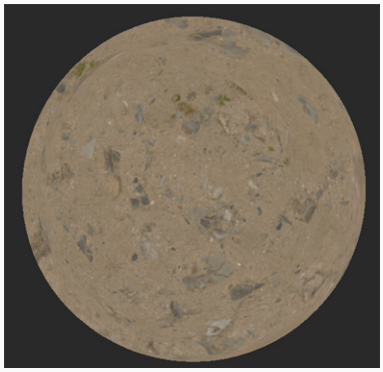
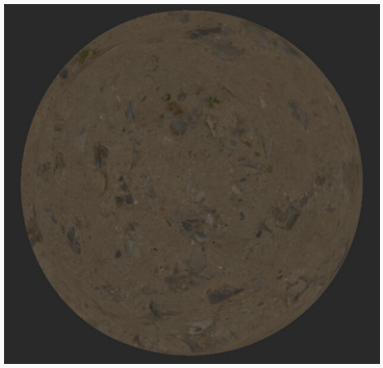
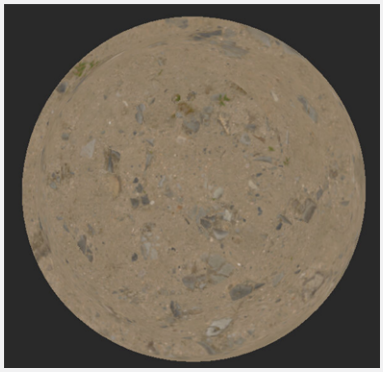
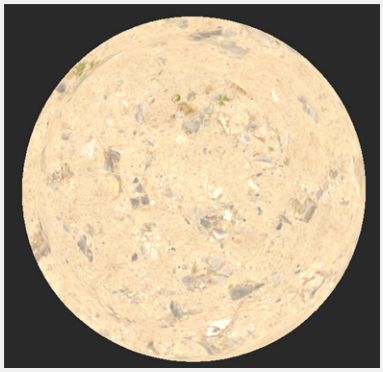
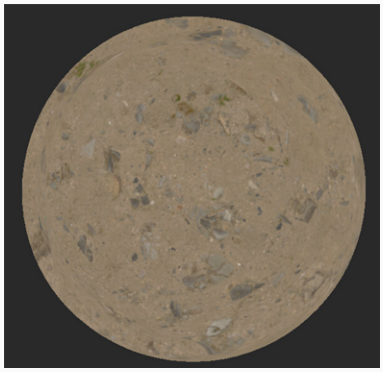
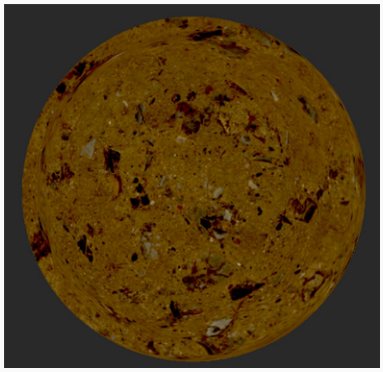
Grade Node Inputs

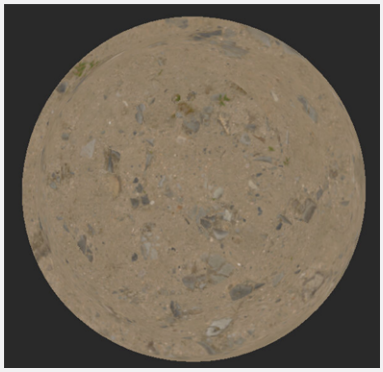
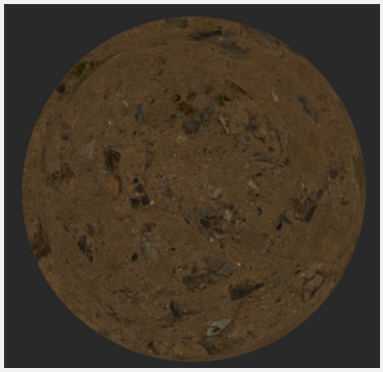
Input - Any nodes containing textures or paint that need to be altered can be connected to the Contrast node, for example, Tiled and Paint nodes.

Grade Node Properties

Blackpoint *floating point control*	Sets the blackpoint of the RGB values. Blackpoint is the color value at which the input is considered to be 100% black.

	 Default: 0.000  0.200
Whitepoint *floating point control*	Sets the whitepoint of the RGB values. Whitepoint is the color value at which the input is considered to be 100% white.  Default: 1.000  0.480
Lift *floating point control*	Lifts the blackpoint, while keeping the whitepoint the same.  Default: 0.000  0.240
Gain	Lifts the whitepoint, while keeping the blackpoint the same.

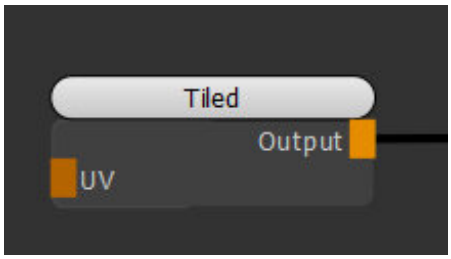
floating point control	 Default: 1.000  0.300
Multiply *floating point control*	Multiplies the value to lighten or darken the texture while preserving the blackpoint.  Default: 1.000  3.500
Offset *floating point control*	Specifies a fixed value to add in order to lighten or darken the texture.  Default: 0.000  -0.160
Gamma *floating point*	Adjusts the midtones, while keeping the whitepoint and blackpoint the same.

control	 Default: 1.000	 0.500

Grade: Node Graph Workflow Example

Let's use a Grade node to change the whitepoint value of the texture.

1. Create a Tiled node.

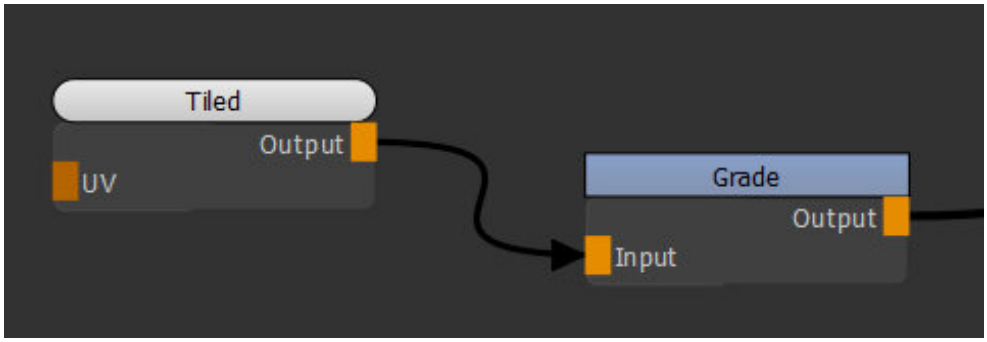


Adding a Tiled node the scene.

This is how the texture of the Tiled node looks:

Tiled node texture.

2. Create a Grade node.
   Connect the Tiled node to the input of the Grade node.



Adding a Grade node to the scene.

3. Open the node **Properties** of the Grade node and adjust the **Blackpoint** value.
   Notice the change of color value.

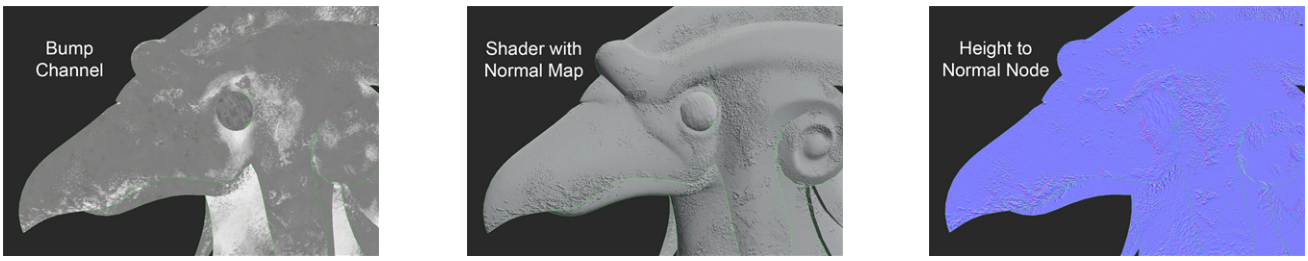
Height As Normal Node

Access: **Nodes** > **Filter** > **Height As Normal**

The Height As Normal node is used to convert a greyscale texture into a normal map. The node needs a **Height** input so it is best to plug in a correctly balanced Bump channel. The Height As Normal node interprets values over 0.5 as bumps, and values below 0.5 as troughs.

Unlike a bump map, a normal map fakes directional data as well as height. Because of this, using your bump map through a Height As Normal node can often give better results in your shader.

Note: Height As Normal creates an approximation, and results may differ once the node is baked or the results are exported.



Using the Height As Normal node to convert a Bump into a Normal map.

Height As Normal Node Inputs

Height: The texture from which the normal output will be calculated. Values over 0.5 are interpreted as bumps, and values below 0.5 are interpreted as troughs.

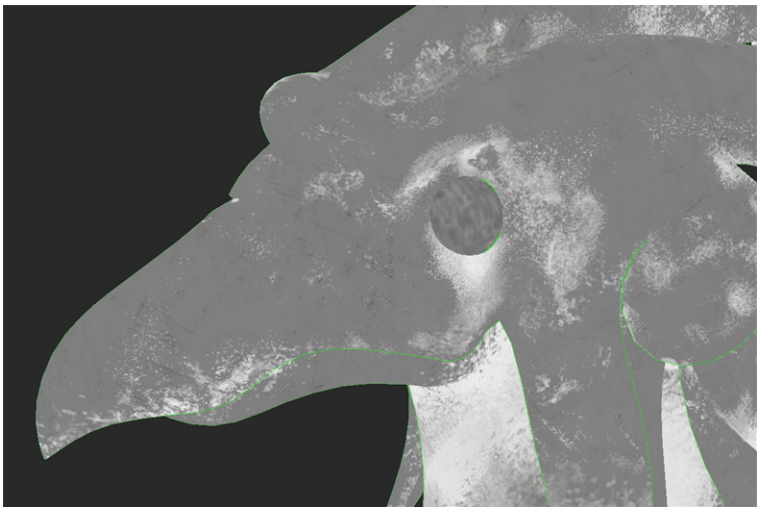
Height As Normal Node Properties

Bump Weight *text field, slider* Default: 0.1	The strength of the output Normal map. Normal maps can be hard to visualize when just viewing the map, so plugging it into a shader can give the best representation of the weight and strength.
Bump Mode *dropdown* Default: Fast	Defines the calculation method of the normal. The options are **Fast**, **Accurate** or **Fastest**. • **Accurate** gives you the smoothest calculation method but takes longer to calculate which can slow down your scene and shader. • **Fastest** is the quickest to calculate but you can get odd results and anomalies on a per pixel basis giving you an inaccurate result. For the best scene and shader response, use **Fastest** and then switch to **Accurate** to export your textures.
Bump Space *dropdown* Default: UV	Changes the Bump's space between UV and World Space.

Height As Normal Node Workflow Example

Creating a Normal Map from your Bump

In the image below a Bump channel has already been set up.

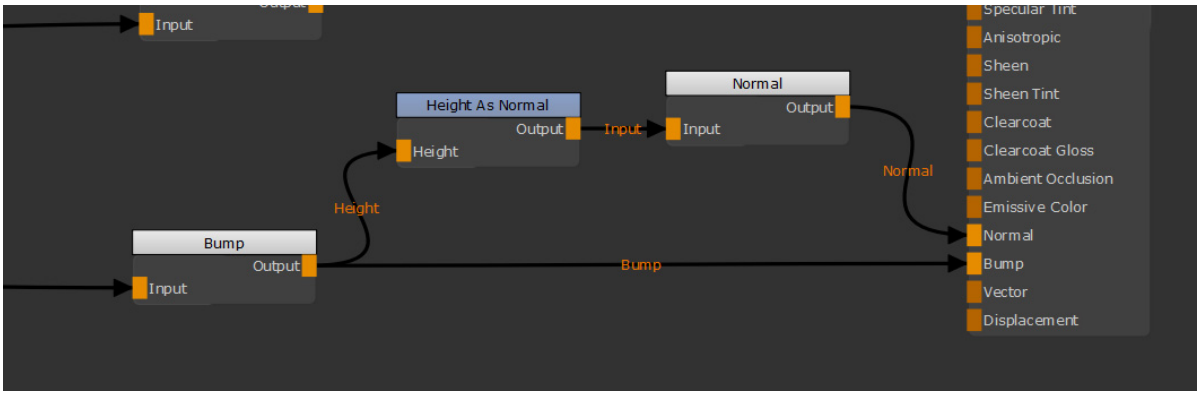


A Bump map for this mechanical bird asset ready to convert into a Normal map.

Using a Height As Normal node you can convert the height data in the Bump map to a Normal Map.

1. Input the **Bump** channel into the Height As Normal node.
2. Input the Height As Normal node into the **Normal** channel of your shader.

In this example, the **Normal** channel is being plugged into a Principled BRDF shader:



Plugging a Bump into a Height As Normal node, and then into the **Normal** channel on the shader.

Changing the **Bump Mode** to **Accurate** gives you the best results, and by adjusting the **Bump Weight** slider you can decide the strength of your normals.

> **Tip:** It is recommended to have a shader set up with the normal when adjusting the **Bump Weight** slider, as viewing the flat texture does not give you the most accurate view of how the normal affects your model at render time.

For more on shader workflows check out the documentation on Principled BRDF nodes.

In the example at the beginning of this topic, the Bump channel that was set up informs the mesh where the areas around the beak and eyes should raise. Converting the Bump map to normal data creates a 'facing' as well as height, which produces more realistic specularity breakup when the light hits the geometry.

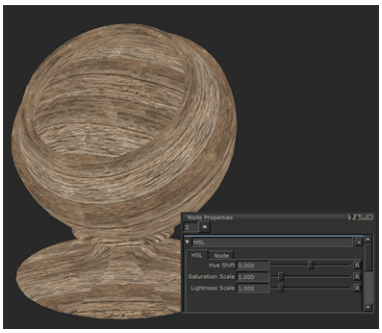
HSL Node

Access: **Nodes** > **Filter** > **HSL Node**

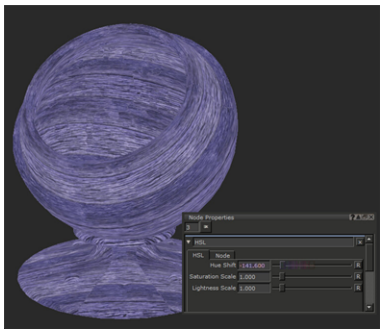
The HSL node manipulates three texture input settings: **Hue**, **Saturation**, and **Lightness** making it ideal for color corrections on your textures. This is a really crucial feature if you're trying to balance reference images with others that you have gotten from elsewhere.

Manipulating the **Hue Shift** and **Saturation Scale** allows you to adjust images to match the look of the rest of your scene.

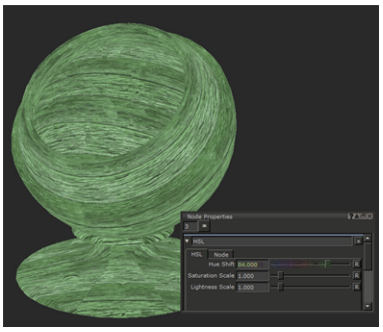
The HSL node is very similar to the HSV Node, which lets you manipulate **Value Scale** rather than **Lightness Scale**. **Lightness Scale** has a maximum of a value of 1, whereas using the HSV node's **Value** property gives you values over 1.



HSL node with a **Hue Shift** value of 0.0.



HSL node with a **Hue Shift** value of -141.0.



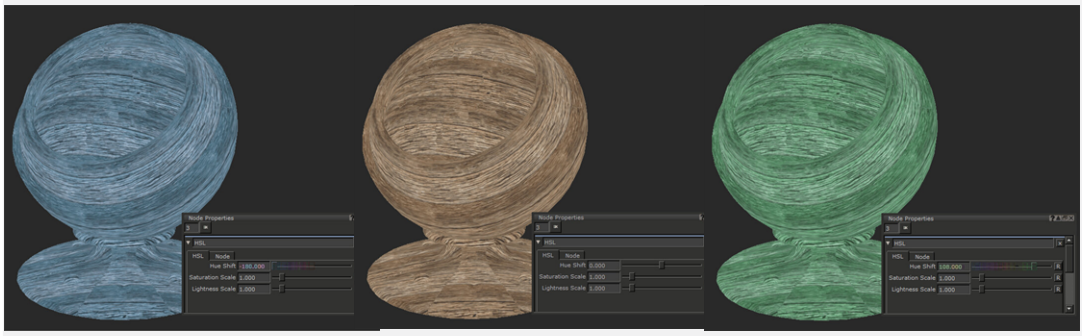
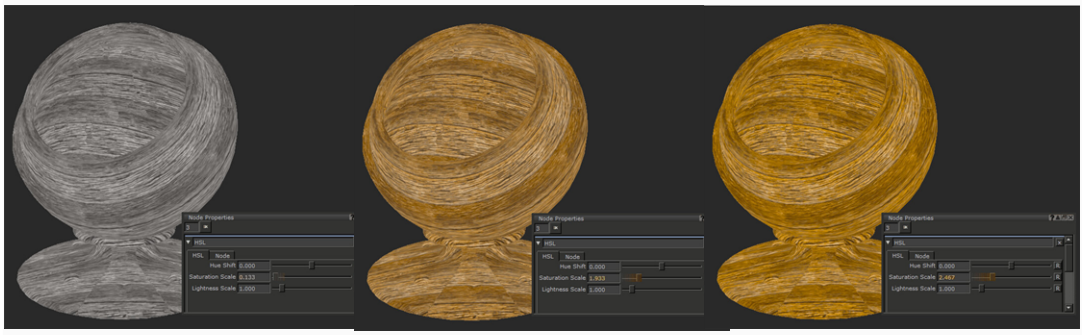
HSL node with a **Hue Shift** value of 84.0.

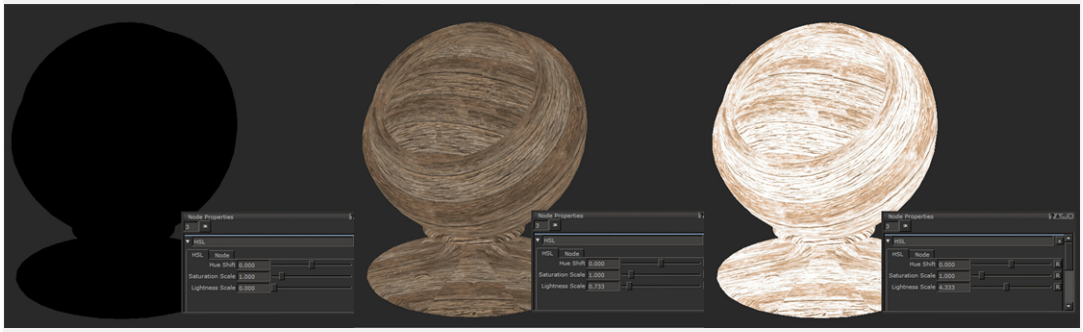
Note: The HSL's **Lightness** scale increase highlights and midtones at the cost of saturation, so sometimes it may be useful to try the HSV node instead.

HSL Node Inputs

Input: What the HSL node adjusts.

HSL Node Properties

Hue Shift *text field, slider*	**Default**: 0.000	Changes the hue of the input. Moving the **Hue Shift** slider cycles through the color spectrum.  **Hue Shift**: -180. **Hue Shift**: 0. **Hue Shift**: 108.
Saturation Scale *text field, slider*	**Default**: 1.0	Multiplies the output saturation. For example, a value of 2 gives you twice saturated an output. A value of 0.5 results in half as much saturation. Setting the slider to 0 outputs a grayscale result.  **Saturation Scale**: 0.1. **Saturation Scale**: 1.9. **Saturation Scale**: 2.4.
Lightness Scale *text field, slider* Default: 1.0		Multiplies the amount of lightness the output has. **Lightness Scale** increases the color value at the cost of hue and saturation For example, a value of 2 gives you twice as light an output. A value of 0.5 results in half as much saturation. Setting the slider to 0 outputs a grayscale result.



Lightness Scale: 0. **Lightness Scale**: 0.7. **Lightness Scale**: 4.3.

HSL Node Workflow Example

Color Correcting Reference with the HSL

The HSL node is great for changing the color of images that don't match the rest of your reference. This is a speed boost to your workflow, rather than having to make changes in photo editing software.

The base color map.

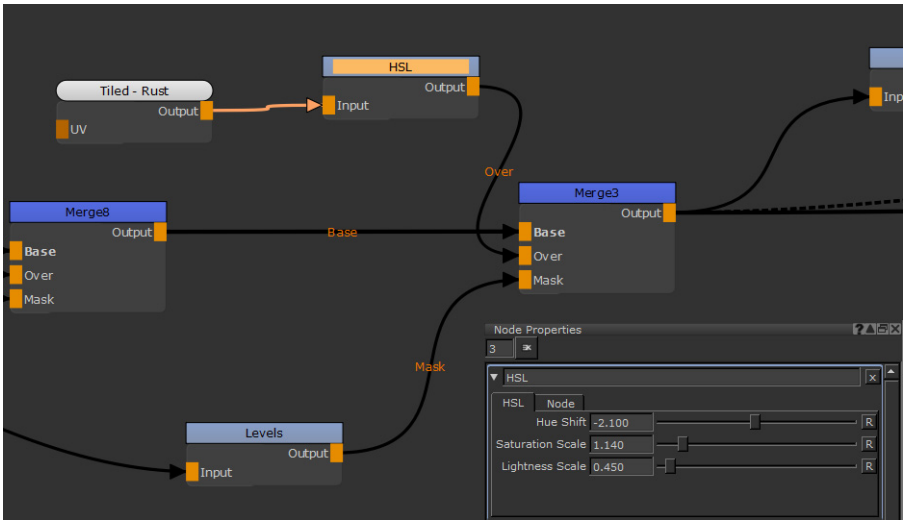
In the example, the base color map is fairly plane. There is a tiled rust texture to merge on top, but the metal base already built up has warm orange and yellow tones.

In the following example, there is very little variation between the base metal and the warmth of the rust when adding the rust on top. Using a HSV node allows you to add some contrast.

There is very little variation in the base color map.

The default tiled texture blends too well and doesn't look like rust compared to the current base color.

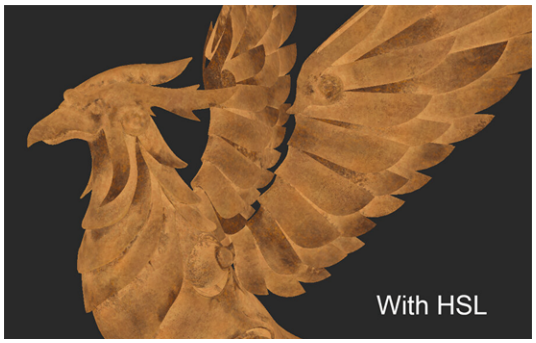
To rectify this, you can add an HSL node to bring the **Hue Shift** down to make it warmer and look like a rustier version of the base. **Saturation Scale** is then increased slightly and the **Lightness Scale** is reduced to make it darker.



A HSL node with **Hue Shift** reduced slightly, **Saturation Scale** increased, and **Lightness Scale** reduced.

To rectify this, you can add an HSL to bring the **Hue Shift** down to make it warmer and look like a rustier version of the base. **Saturation Scale** is then brought up slightly and the **Lightness Scale** is brought down to make it darker.

The mask texture of the merge node can then be hand painted, letting you decide where the rust is.



Without the HSL node, the darks of the rust image do not fit well as they have very little contrast to the rest. Using the three sliders of the HSL node, you can create something that represents a real world reference.

Creating a Secondary Channel from a Color Channel with the HSL

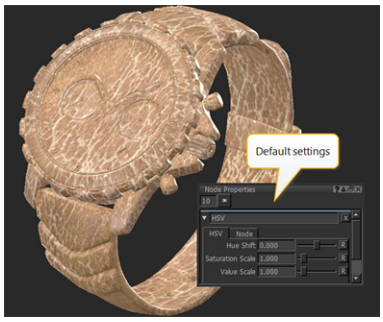
Because an HSL node can desaturate a color texture and create a grayscale version, it is really useful for creating secondary maps from your **Base Color** channel. Secondary maps such as Specular roughness or Bump, should line up with your painted texture mask details. You can create a secondary map by adding an HSL node and setting the **Saturation Scale** to 0.

Then you can start grading parts with masks to get a secondary map correctly balanced to how you would want that channel to look. For more on how to do that, see the Levels Node topic.

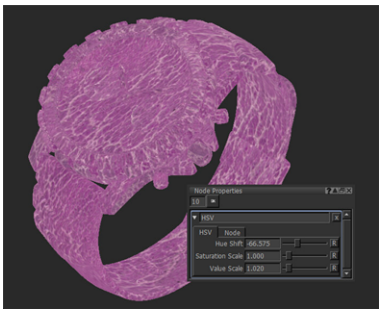
HSV Node

Access: **Nodes** > **Filter** > **HSV**

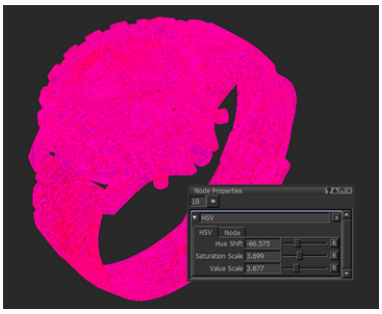
The HSV (Hue, Saturation, Value) node is a filter node that adjusts the input data by letting you change the hue, saturation, and value of the colors that are fed into it.



HSV node's default settings



*HSV node's **Hue Shift** decreased, **Value Scale** increased*



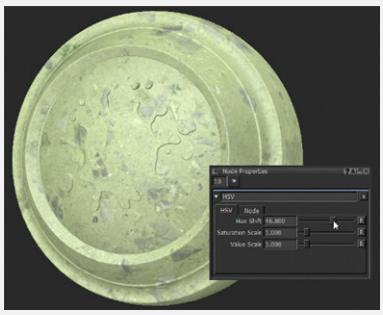
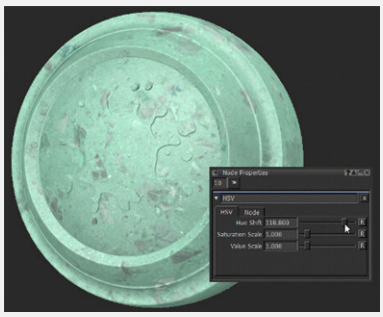
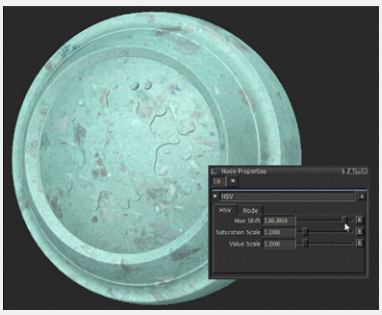
*HSV node's **Hue Shift** decreased, **Saturation Scale** and **Value Scale** increased*

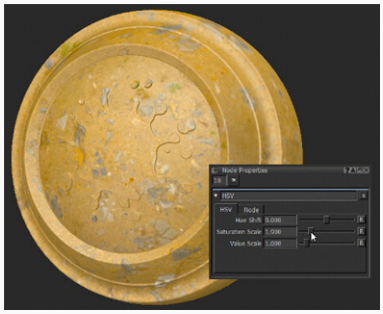
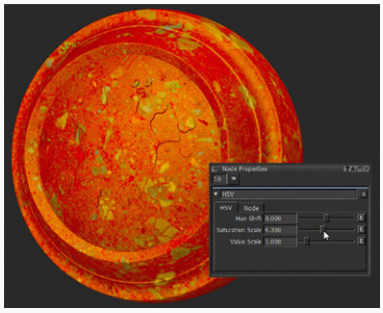
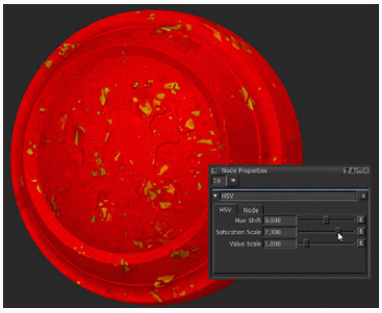
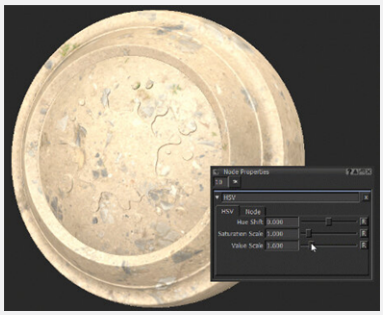
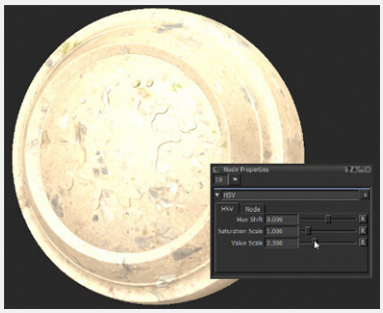
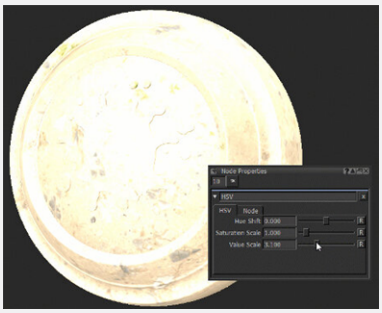
Note: The HSV node is similar to the **HUE** filter. Unlike the HSV node the **HUE** filter is destructive. See Available Filters for more information.

HSV Node Inputs

Input - Any nodes containing textures or paint that need to be altered can be connected to the HSV node, for example, Tiled and Paint nodes.

HSV Node Properties

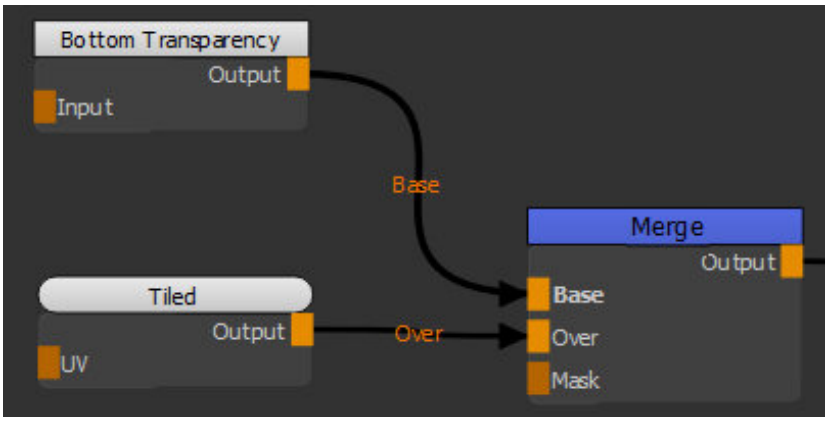
Hue Shift *floating point control*	Changes the hue (or tint) of the colors. This moves the colors around the color wheel.  A **Hue Shift** value of 46.  A **Hue Shift** value of 118.  A **Hue Shift** value of 136.
Saturation Scale	Increases or decreases the depth of color of the hue.

floating point control	 A **Saturation Scale** value of 1.  A **Saturation Scale** value of 4.  A **Saturation Scale** value of 7.
Value Scale *floating point control*	Increases or decreases the intensity of the hue.  A **Value Scale** value of 1.  A **Value Scale** value of 2.  A **Value Scale** value of 3.

HSV: Node Graph Workflow Example

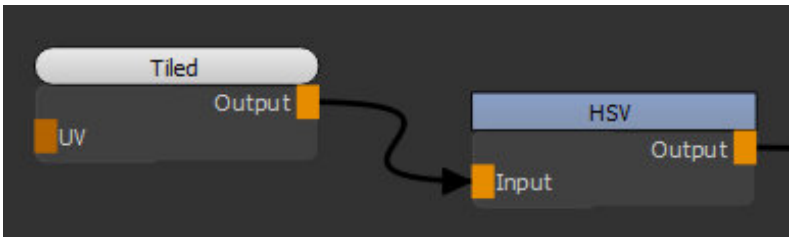
In the example below, let's use an HSV node to change the saturation of the texture.

1. Create a Tiled node.



Adding a Tiled node to the scene.

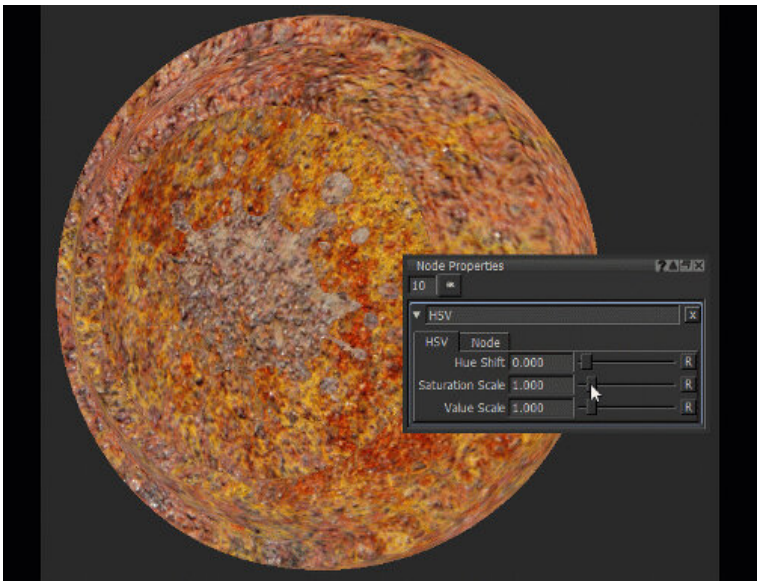
2. Create an HSV node. Connect the Tiled node to the **Input** input of the HSV node.



Adding an HSV node to the output of the Tiled node to adjust the saturation of the texture.

3. Open the **Node Properties** (see HSV Node Properties) of the HSV node and change the value of the **Saturation Scale**.

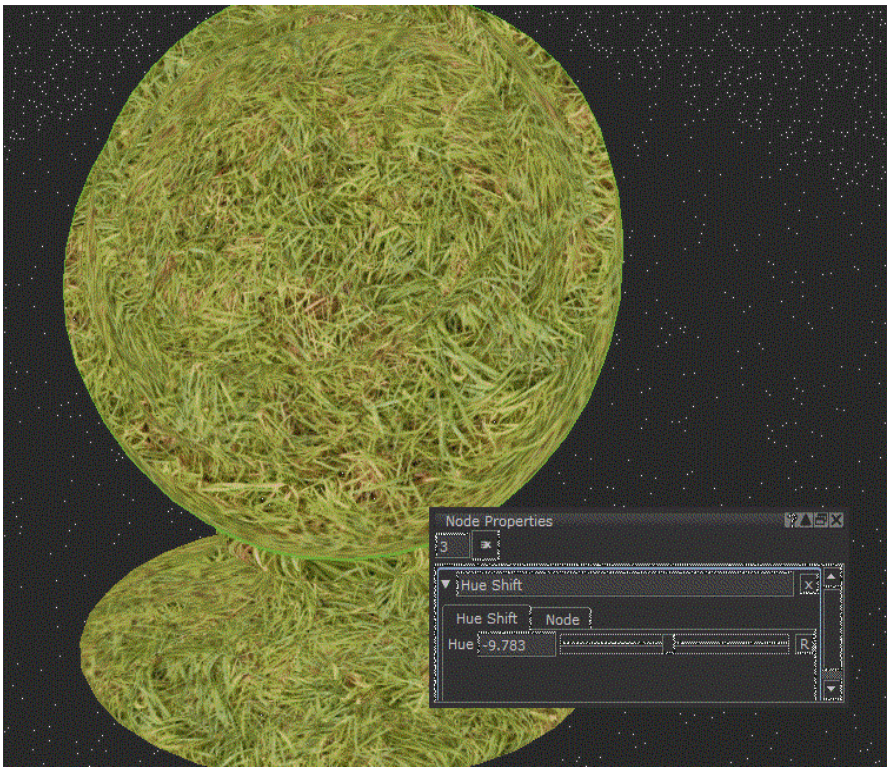
   Notice how the texture becomes more saturated as the **Saturation Scale** parameter is adjusted.



*Changing the **Saturation Scale** value to adjust the saturation of the texture.*

Hue Shift Node

Access: **Nodes** > **Filter** > **Hue Shift**



The Hue of a Tiled node being adjusted with the Hue Shift node.

The Hue Shift node allows you to edit the color of your input. It does not change the saturation or value of the input so it has little to no effect on an image with mostly grayscale values.

For more color correction options, an HSV Node or HSL Node may be useful too.

Tip: Color correcting tiled textures and images in your resource manager is a great way to re-use textures and keep your scene tidier.

Hue Shift Node Inputs

Input: The Hue Shift node adjusts this input.

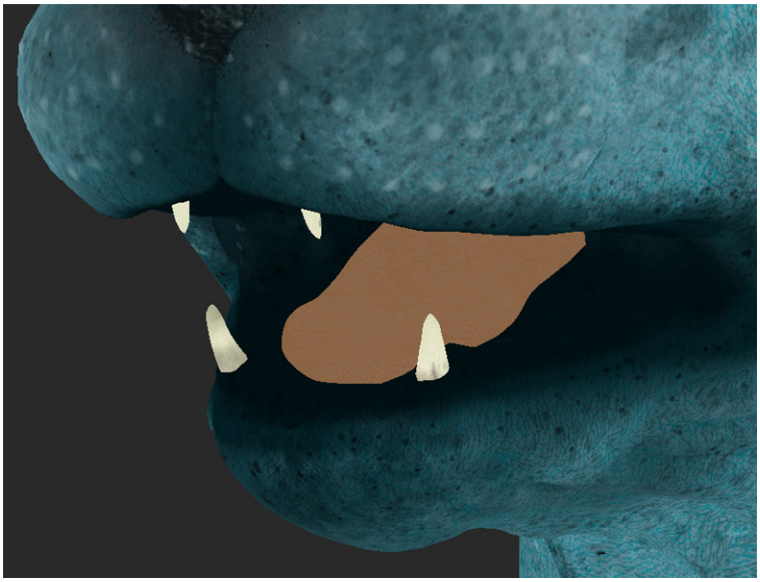
Hue Shift Node Properties

Hue Shift *text field, slider*	Changes the hue of the input. By moving the slider either left or right, the hue cycles through the color spectrum. A value of 0 is the hue that your input is. Default is 0.

Hue Shift Node Workflow Example

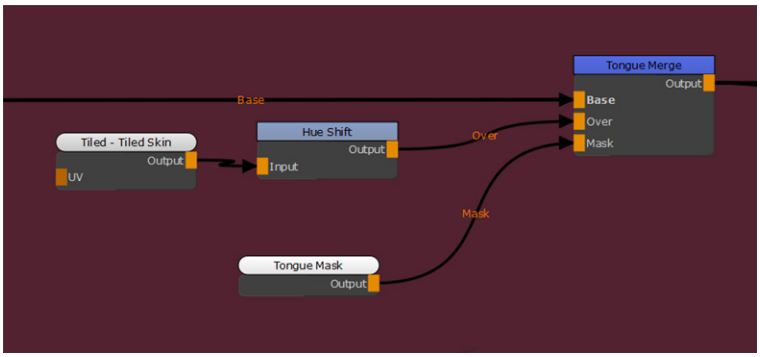
Reusing Textures with Hue Shift

Here we have a creature that was textured using a base tiled skin texture. However, while the tongue a similar texture, it's more of a pinky tone. It would be great if we can reuse the texture rather than finding another one.



While it's a good base for our textures, the color of the tongue is wrong.

To reuse that texture for that part of the mesh, a Hue Shift node can be added. By merging it on top of the Node Graph so far and masking off the tongue, a redder version of that texture can be used to correct the issue. As you can see below with a few simple nodes, it can be sorted and we have a great base for the tongue.



A simple node setup to add the corrected tongue texture into the Node Graph.

Below is an example of the mask being used.

This mask makes sure the hue-shifted texture is only added to the isolate tongue.

This is now a good base to start adding extra texture detail on top of. If you had a grass tiled image, you can use a Hue Shift node to make it more brown, and very quickly you have a desert grass variation.

Image Write Node

Access: **Nodes** > **2D** > **Image Write**

The Image Write node is a 2D node that allows you to export/write your image or edits to your image to the Image Manager. If the image is used in any procedural nodes in 3D mode, rebaking the Image write node will update all of them with the new changes.

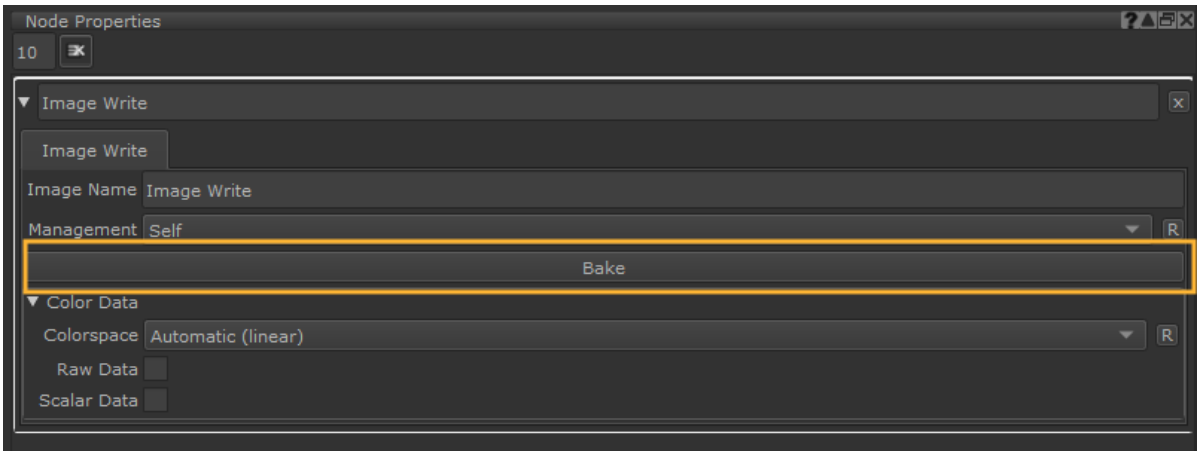


Image Write Node Inputs

Input: The Input allows you to specify which node stream to be baked and exported to the Image Manager.

Image Write Node Outputs

Output: If the Image Write node is unbaked, the Output shows the live input. Once an Image Write has been baked, it'll show the baked image; upstream changes will not be shown live.

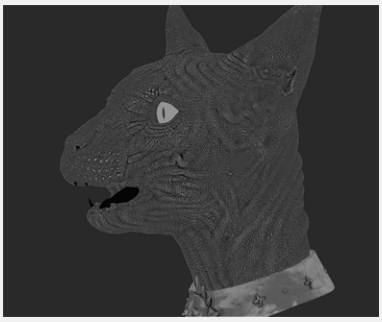
Image Write Node Properties

Image Name	text	The name of the image that will be written to the Image Manager.
Management	dropdown	This option decides where the node's properties come from.

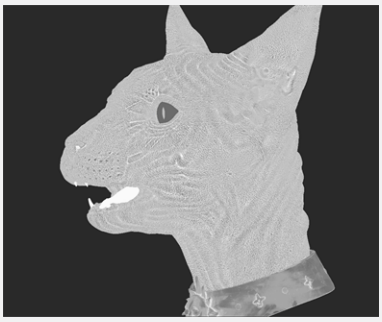
		The default of **Self** allows the user to change them. The other option **Channel** defines them from the channel the Image Write Node is connected to.
Colorspace	dropdown	The colorspace to which the output values are converted. Automatic is determined by the file name, size, and type of data in the image.
Raw Data	checkbox	If this is enabled, the image data is not converted. **Note:** As the raw colorspace nullifies the corresponding colorspace transform when either used as the input or output, there is a discrepancy between what is shown in the Image Manage palette thumbnail and the image viewer. This is the expected behavior.
Scalar Data	checkbox	Enable this option to indicate that pixel data is not affected by color management.

Invert Node

Access: **Nodes** > **Filter** > **Invert**



Bump map.



Inverted Bump map.

The Invert node is very useful when it comes to secondary channel creation or masks for Merge nodes. It flips the values of your textures. By default it flips the R, G, and B channels, but you can also enable or disable individual channels or include the alpha too.

Invert Node Node Inputs

Input: Inverts the input.

Invert Node Node Properties

Red, Green, Blue, Alpha *checkbox*	By enabling or disabling the checkboxes you define which color channels of the input you want to invert. By default all channels are enabled. If you want to include the **Alpha** or disable any of the others you can do so with these checkboxes.

Invert Node Workflow Example

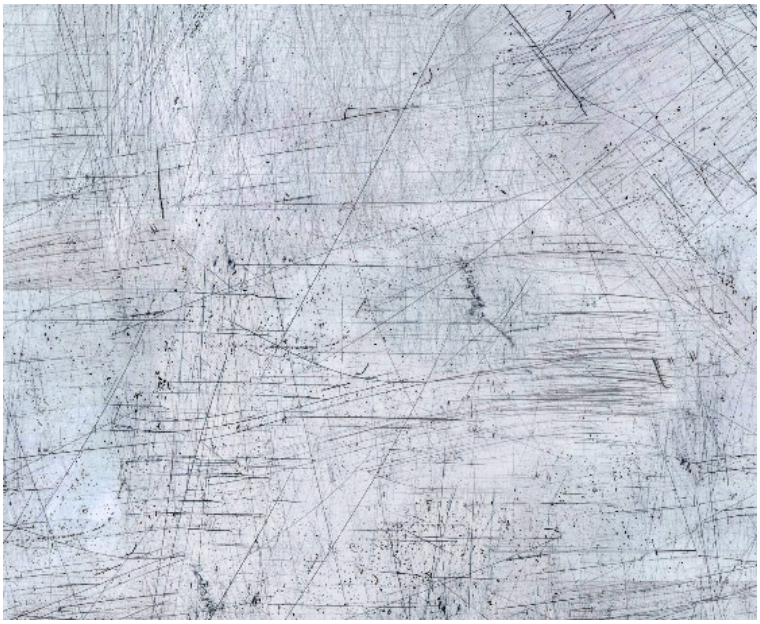
Inverting to Work with Blend Modes

The Invert node is useful when creating masks for merge nodes. Depending on the blending mode you use, you may need to use an Invert node to flip the black and white values so it works correctly.

In the example, we have the color map for a watch asset that we want to add a bit of detail to.

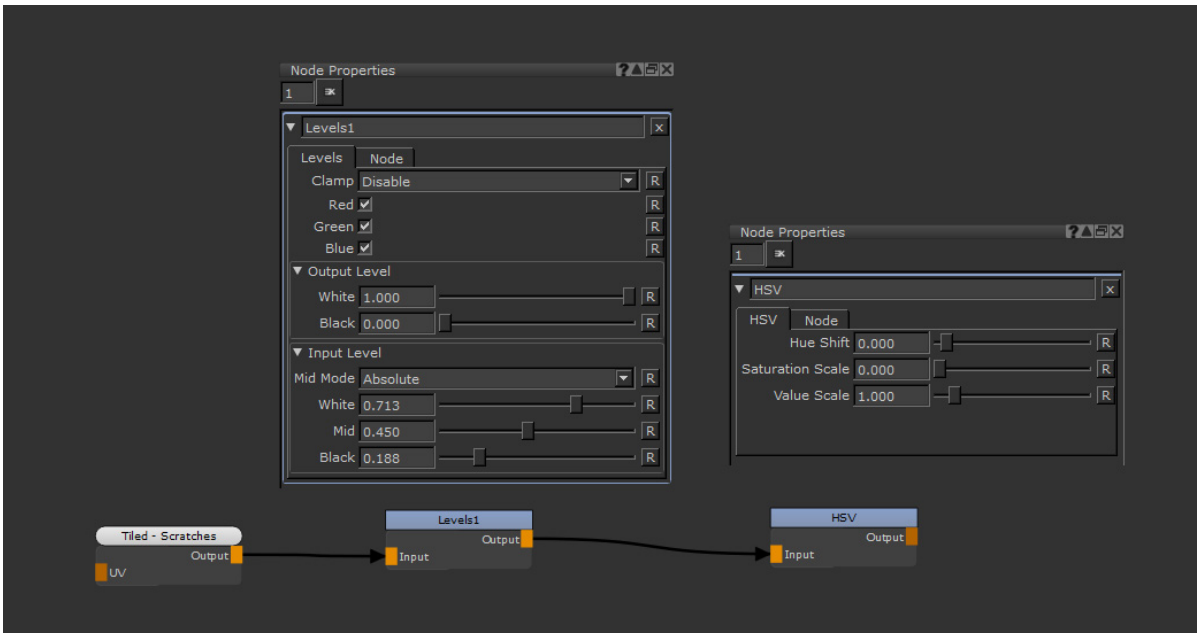
A color map for a watch asset.

Metals often get wear and tear, some lets add some lighter scratches to the base color. This scratched texture is a great image that can be added on top of our textures so far.



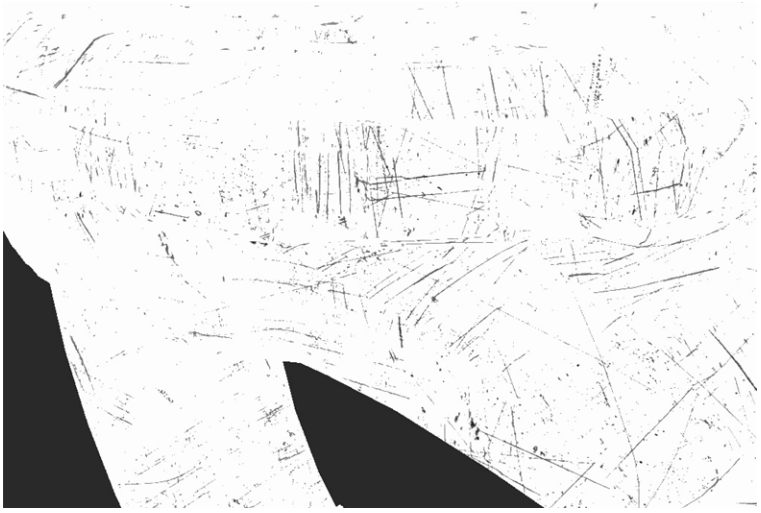
A scratched texture loaded into the image manager.

Lets get this image from our image manager onto the mesh by adding it as a Tiled node. Next, use a Levels node to make the image more contrasted, and then an HSV node to remove the unwanted saturation of the image.



Using a Levels node and HSV node to color correct our scratches to be used as a black and white mask.

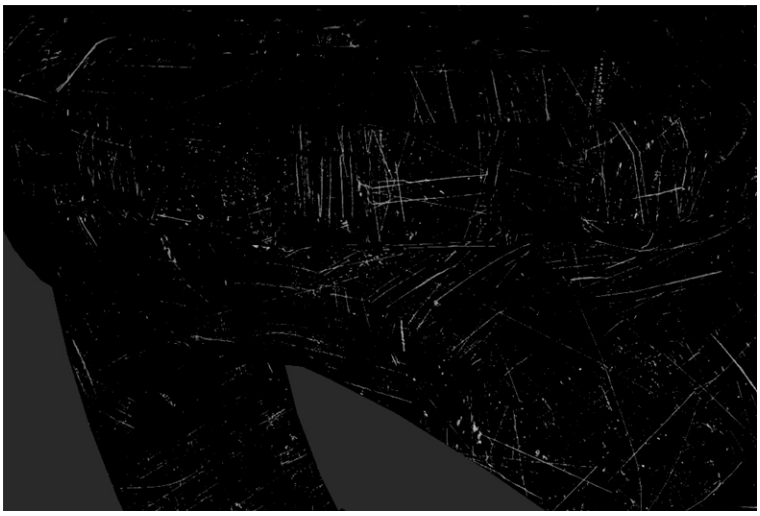
Now we have the scratches ready to be used as a mask or merged on top of our mesh.



The color corrected scratched textures on the watch mesh.

To remove the scratches from the background of this unedited texture and merge it on top of our base color, we would have to use a **Multiply** blend mode. This removes the lighter background and adds the darker scratches.

However, in this case we want the scratches to be lighter, not darker. We can use an Invert node to flip the dark scratches.

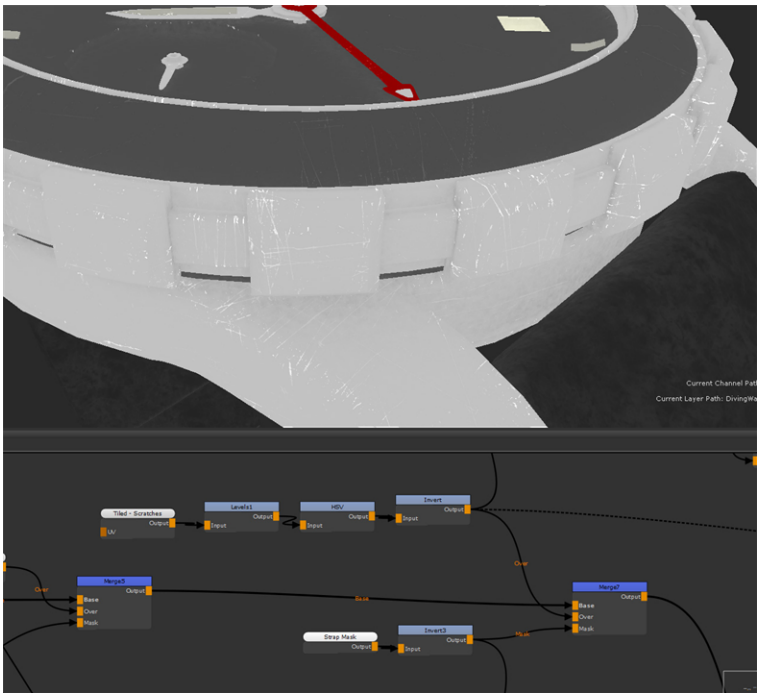


The scratches texture inverted.

From there we can use a blend mode which just adds the light scratches and removes the dark background. This is the **Add** blend mode.

Now that these scratches are ready to be used we can combine them in with the rest of our node tree. The Merge7 in the bottom right corner of the example is set to **Add** so these scratches are lightening the metal.

On top of that, the Paint node called **Strap Mask** is being inverted so the scratches don't affect the strap, and only affect the metal on the watch face. This is another great use of the Invert node - flipping masks that you have already painted for other uses.



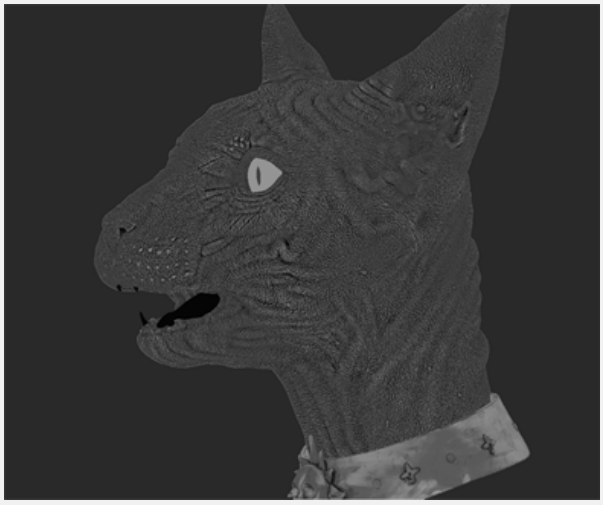
Using an Invert node to flip not only the scratches so they can be merged with the **Add** blend mode, but also to flip a mask

Inverting to Create a Specular Roughness Channel

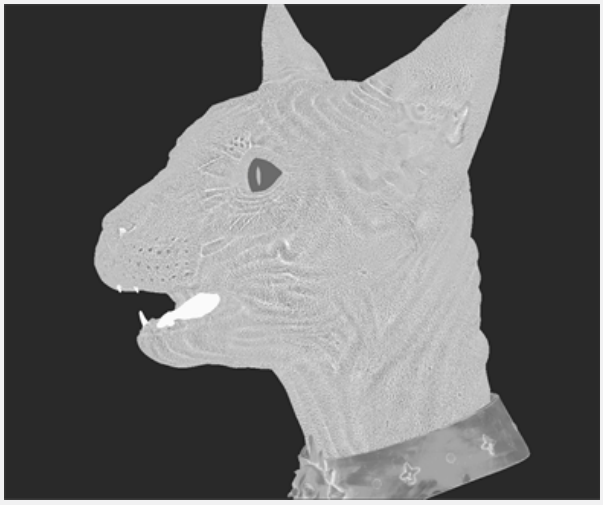
Invert is especially useful when creating a specular roughness map from other channels.

Specular roughness works in the opposite way to a lot of other channels. The dark values give a tighter specularity and the white values give rougher specularity. You can make a quick specular roughness base by inverting your **Specular** channel or by de-saturating your color map and then inverting it.

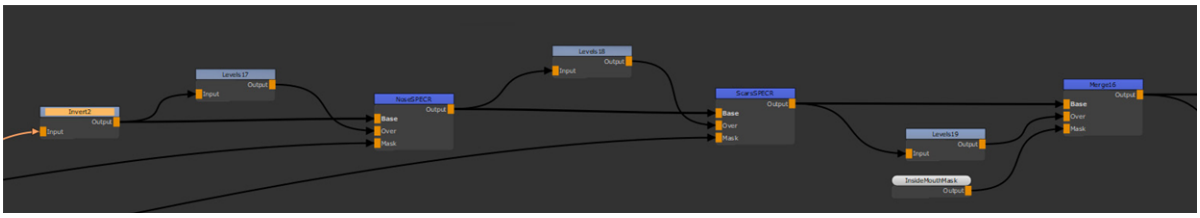
In the example is an asset which has a **Color** and **Bump** channel already textured. The bump for the skin will be made into a **Specular Roughness** channel so the details line up.



Bump map.



Inverted bump map with Invert node.

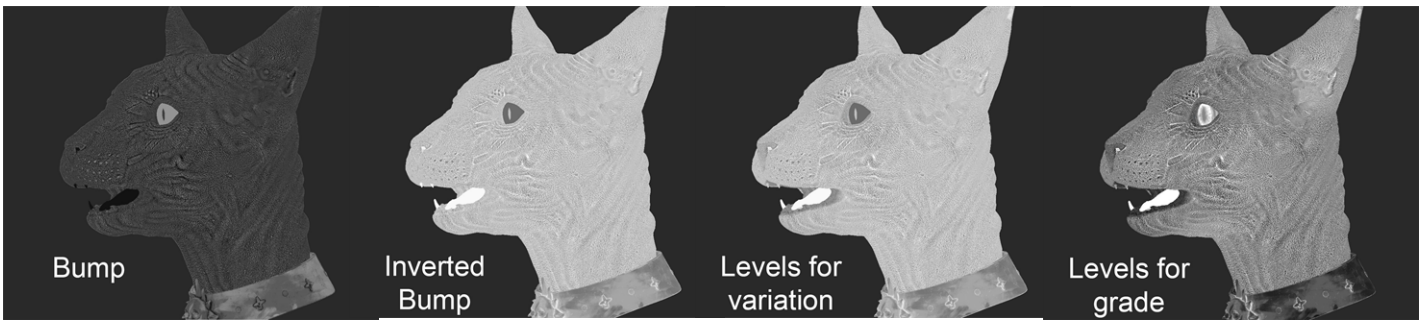


The Node Graph for creating this **Specular Roughness** channel.

By first inverting the bump, we get the areas that push in to be lighter, meaning they would be more rough.

From there you can use Levels nodes to apply different grades to different parts of the mesh. Since inside the mouth, the nose and around the eyes should appear wetter, creating a mask for them and using a Levels node means you can grade them darker, making the spec tighter.

Tip: For more on this workflow see Levels Node.

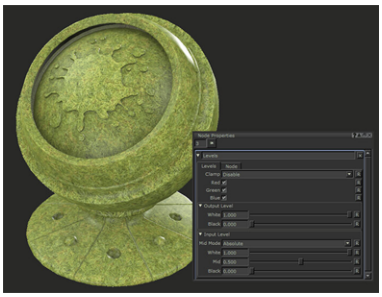


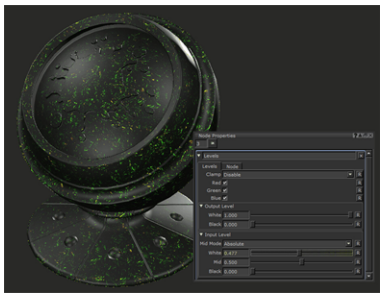
Then you can add another Levels node to help add more contrast. Texturing is never just about one node, it's often about using them in conjunction with each other. Without the Invert node this wouldn't have the opposite values required.

Levels Node

Access: **Nodes** > **Filter** > **Levels**

The Levels node takes your existing image color information and remaps in the input levels (color amount) to the specified output. By manipulating the white and black amounts of the input and output you can effectively manipulate the brightness of your textures.





A Levels node used on a tiled image with **Input Level White** values of 1.0, and 0.4 being used.

The Levels node is often used to add small color corrections, for example to lift or darken your textures, but is also very important when it comes to creating masks. By manipulating your tiled textures, projections or other elements in your project, the Levels node can really help you get the most out of your images. As well as subtle changes, you can use it to make big changes and get something completely different.

The Levels node also has the option to change single color channels if, for example, you want to darken just the Red channel in your image without changing the Blue or Green.

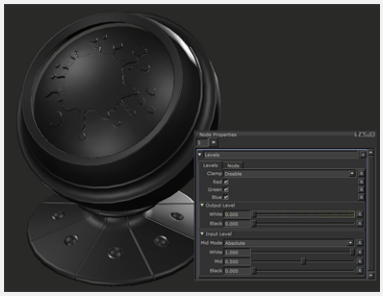
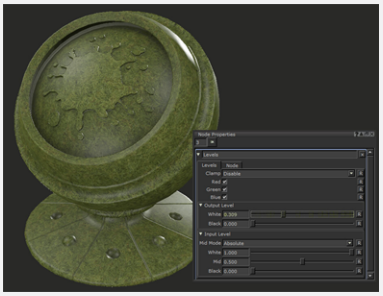
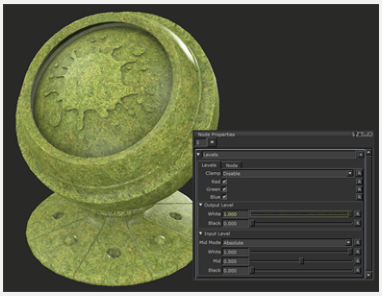
Levels Node Inputs

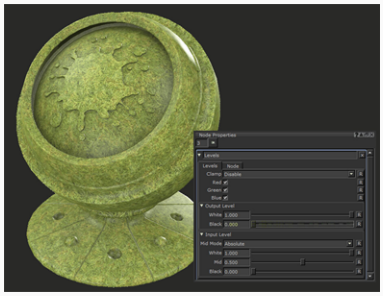
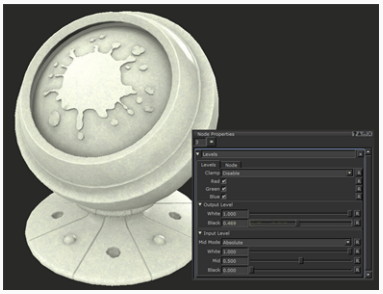
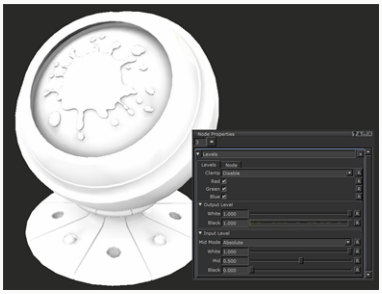
Input - Defines which nodes and textures the Levels node affects.

Levels Node Properties

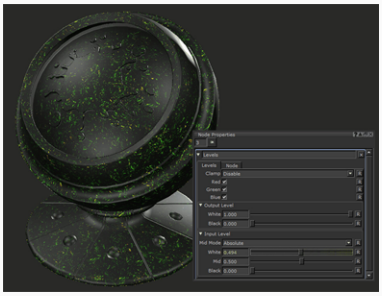
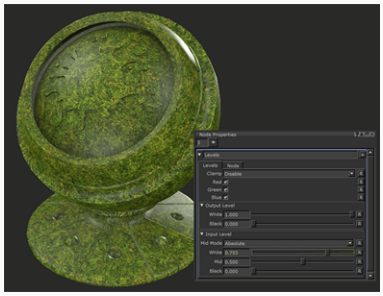
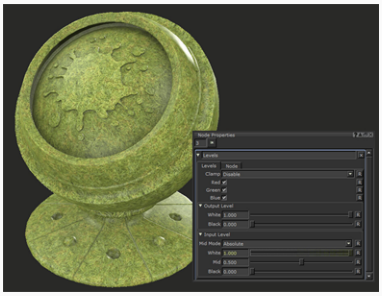
Clamp *dropdown list*	The **Clamp** dropdown lets you clamp values above 1 and below 0. Clamp is disabled by default, but you can change it to **Clamp Output Values**, or **Clamp LDR** (Clamp Low Dynamic Range). If you are having unpredictable results from a Merge node later in your node tree, or a you change the levels of a mask that doesn't seem to behave right, then clamping may fix that.
Red, Green, Blue *checkbox*	Tells the Levels node which color channels of your input you want to adjust. By default the red, green, and blue channels are selected.

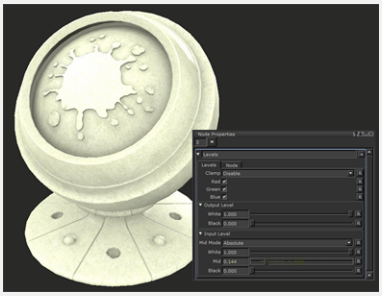
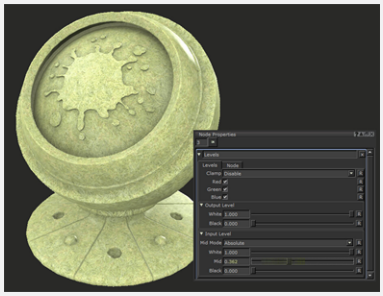
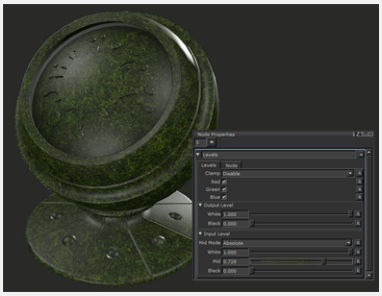
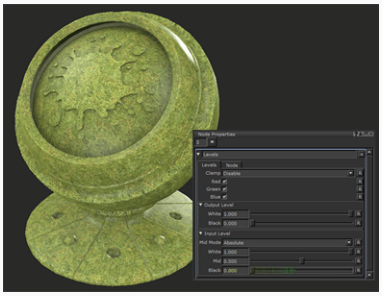
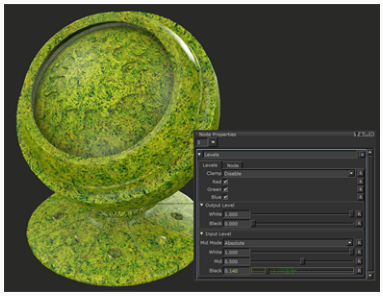
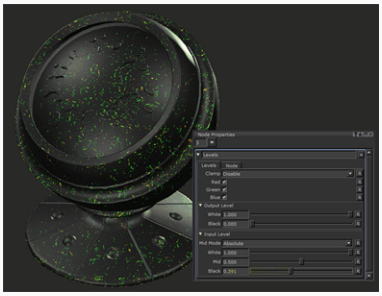
Output Level

White *text field, slider*	Applies a negative or positive gain on the white levels of the output.    **White** level at 0. **White** level at 0.3. **White** level at 1. Default is 1. While the slider maxes at 1, you can increase it by typing in the text field.
Black *text field, slider*	Applies a negative or positive gain on the black levels of the output.

	   **Black** level at 0. **Black** level at 0.4. **Black** level at 1. Default is 0. While the slider maxes at 1, you can increase it by typing in the text field.

Input Level

Mid Mode *dropdown list*	Changes the **Black** and **White** midpoint level as either **Absolute** color data of 0 and 1, or **Relative** to the black and white values you provide. For example, if **White**Input level is 1.0, and **Black** Input level is 0.2, with a **Midpoint** defined of 0.5: In **Absolute** mode, the midpoint is 0.5. In **Relative** mode, the midpoint is 0.6.
White *text field, slider*	Changing the **White** value redefines what input value is pure white. Decreasing the **White** value pushes the lighter values closer to white, potentially losing information by having large areas of pure white.    Input **White** value at 0.5. Input **White** value at 0.7. Input **White** value at 1. Default is 1.
Mid	Changing the **Mid** value redefines mid gray in your texture. Moving it lower will shift your

text field, slider	whole texture lighter, and higher will darken your textures.  Input **Mid** level at 0.1.  Input **Mid** level at 0.3.  Input **Mid** level at 0.7. Default is 0.5.
Black *text field, slider*	Changing the **Black** value redefines what input value is pure black. Moving it up moves the darker values closer to black, potentially losing information by having large areas of pure black.  Input **Black** level at 0.  Input **Black** level at 0.1.  Input **Black** level at 0.3. Default is 0.

Levels Node Workflow Example

Texture Grading with Levels

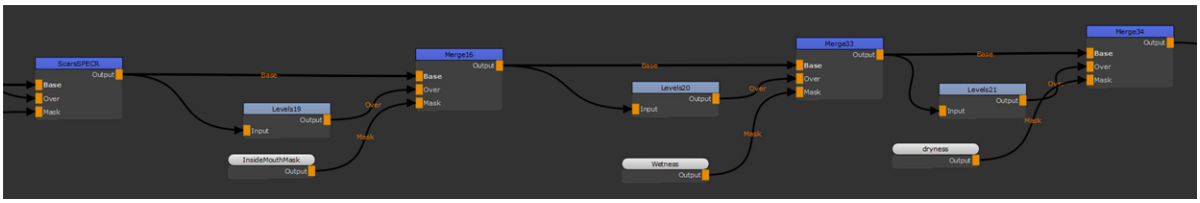
Adjusting the input levels of your **White** and **Black** up and down you can add contrast to your textures easily. Unlike a Contrast node, you have a lot more control over making fine tweaks to your dark or light values.

If you have imported image resources that need to match images from a different source (such as an on-set photography reference) a Levels node is a great way to get two images or projections to work together. The

Levels node is also a good node to use in conjunction with an HSV node to adjust the **Saturation Scale** and **Hue Shift** for final color tweaks.

Creating Secondary Maps with Levels

Levels are especially useful when creating channels from other channels.



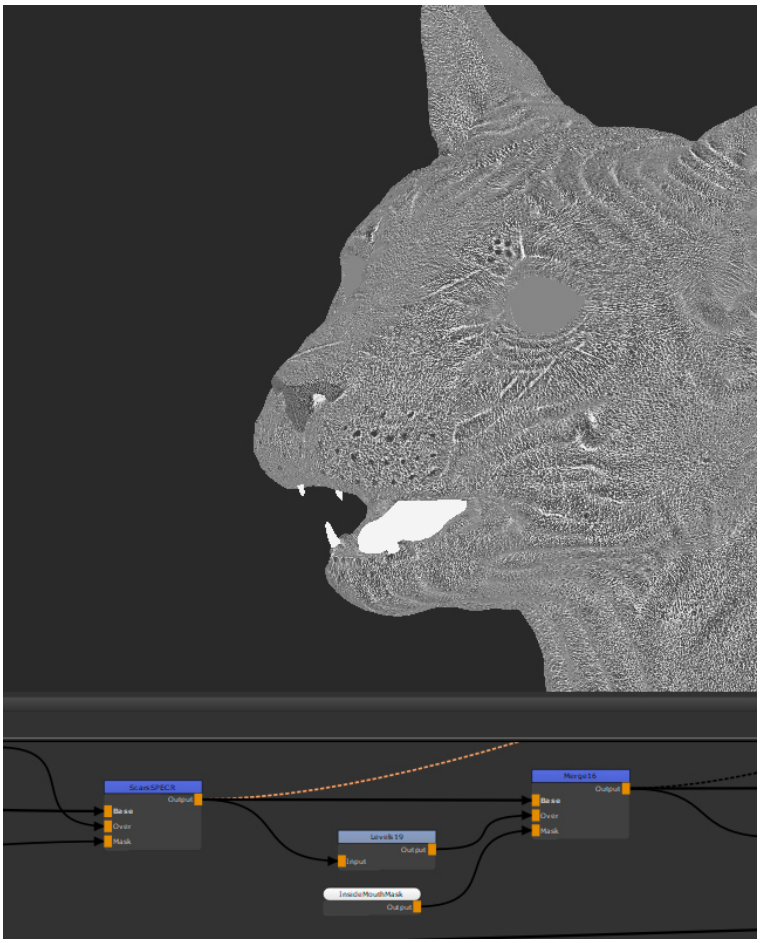
An asset which has had a **Color** and **Bump** channel already textured. Here we are using Levels nodes with different masks to tweak the values of different parts of the mesh.

The bump for the skin is converted into a **Specular Roughness** channel so the details line up.

Let's break down how to make a channel from another one:

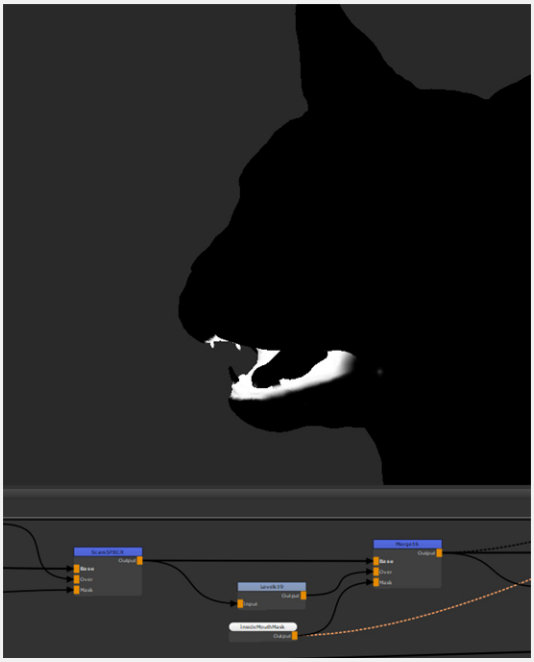
1. Inverting the **Bump** channel gives you areas that push in to be lighter, meaning they would be more rough.
2. From there you can use Levels nodes to apply different grades to different parts of the model.
3. Since you want the inside of the mouth, the nose, and around the eyes to appear wetter, creating a mask for them and using a Levels node means you can grade them darker, making the specularity tighter.

Let’s edit the mouth first. Here is the base texture:

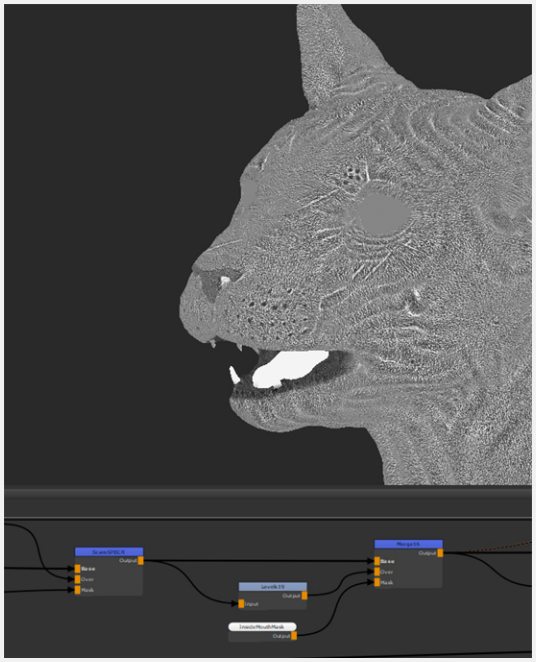


As you can see in the above image we can add a Levels node to darken the inside of the mouth, making the specularity tighter.

We also need to define which part of the mesh this Levels node affects. We can then use a mask of the inside of the mouth on the Merge node to only make that area darker:

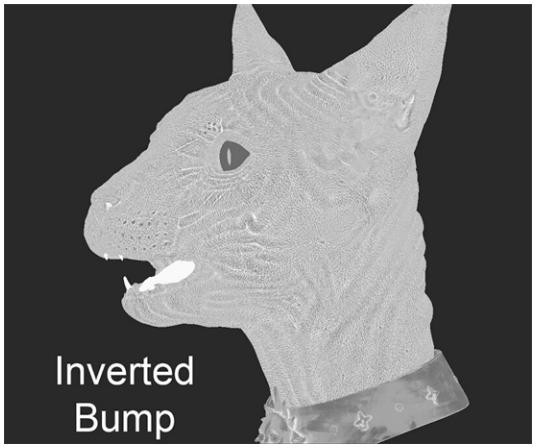


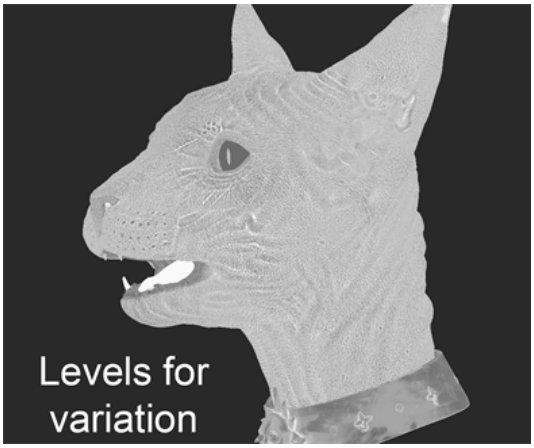
The mask.



The overall effect, isolated to the mouth.

This is a great way to use the Levels node to add edits to parts of your texture. You can use multiple Levels to make a series of changes to your texture that end up changing one map to be used as another. The image below breaks down the process:





This shows you can use a final Levels node to help add more contrast or overall balance the map.

Experimenting with the **Input Level** sliders, you are able to add contrast and bring a wider value range in the **Specular Roughness** channel. Like most nodes, it's about trying different values until you get the result you want.

Multi-Channel Merge Node

Access: **Nodes** > **Layer** > **Multi-Channel Merge**

The Multi-Channel Merge node lets you combine two materials together. It may seem rather overwhelming to begin with but if you know how to use a Merge node, you know how to use this one.

The Multi-Channel Merge node has the same inputs as a Merge node. These inputs are **Base**, **Over** and **Mask**, but the Multi-Channel Merge node has different types of outputs for each of the texture channels a material can output.

If you are working in the Node Graph with materials, a Multi-Channel Merge node is the easiest way to quickly combine materials together, with the ability to mask parts of a mesh out and change blend nodes on a per channel basis.

> **Note:** To learn how to combine two materials together, see Multi-Channel Merge Node Workflow Example.

When you place down a Multi-Channel Merge node, Mari asks what kind of **Shader Model** you are using. You need to select the one that matches up with the materials that you are using in your scene as it chooses the input and output channels the node has based on the shader type.

> **Note:** You can merge two multi-channel material nodes by pressing **M** on the keyboard to create a Multi-Channel Merge node. To learn more, see Merging Multi-Channel Material Nodes in the Node Graph.

Multi-Channel Merge Node Inputs

Base: Define which material and the corresponding channels to use as the base to be merged on top of.

Depending on the selected shading model you get different options. For the **Principled BRDF** shading model there are 17 texture channels that can be inputted:

- **Base** - Base Colour
- **Metl** - Metallic
- **Subs** - Subsurface

- **Spec** - Specular
- **Rgh** - Specular Roughness
- **SpcT** - Specular Tint
- **Anis** - Anisotropic
- **Shn** - Sheen
- **ShnT** - Sheen Tint
- **Clct** - Clearcoat
- **Cctg** - Clearcoat Gloss
- **AO** - Ambient Occlusion
- **Emis** - Emissive
- **Norm** - Normal
- **Vect** - Vector
- **Bump** - Bump
- **Disp** - Displacement

Note: The texture channel names appear as full names on the node's input and output streams.

Over: Define which material and the corresponding channels to merge on top of the base.

Depending on the selected shading model you get different options.

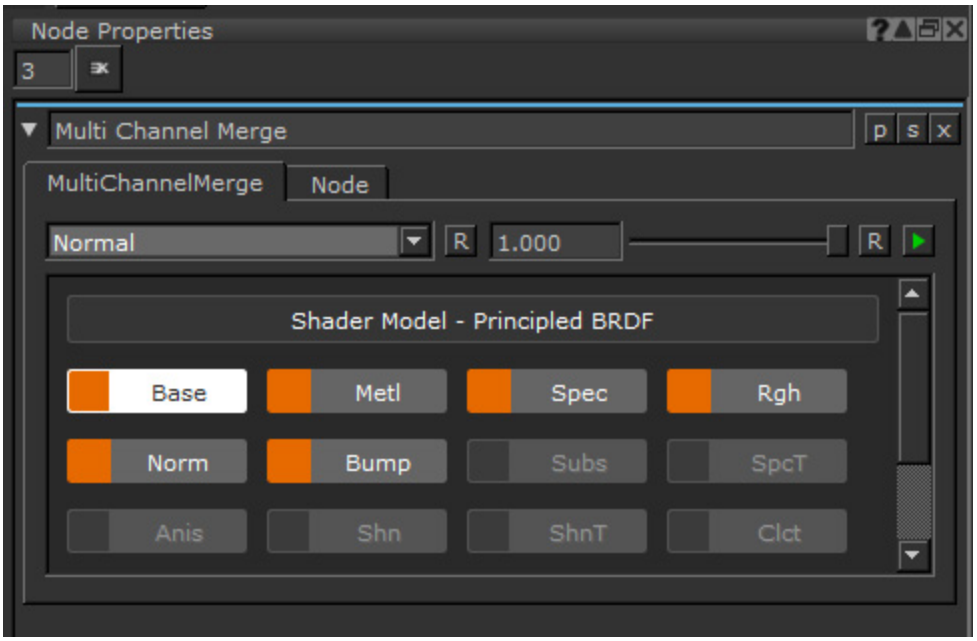
Mask: Uses a black and white mask to merge the **Over** inputs onto the **Base** inputs.

Multi-Channel Merge Node Properties

The Multi-Channel Merge node has a slightly different UI compared to the standard Merge node and other nodes. You can turn off the merging of individual channels by enabling and disabling the orange checkboxes under the Shader Model panel.

Note: Hold **Shift** and double-click the Multi-Channel Merge node to expand each of the merges in the **Node Properties**. To learn about the properties for each merge, see Merge Node Properties.

Note: You are only able to enable and disable channels and change blend modes for nodes that lead into a Channel node. This is in case you want to put a paint material over a brick material, but want the brick bump or displacement to show through the paint. Disabling the bump the paint provides in this scenario allows the brick bump to show instead.



You also do not change the blend mode for every channel, when you change the blend mode you only do it for the current channel selected.

Amount *text field, slider*	A modifier to the change the blend amount. 0 shows **Base** fully, and 1 blends the **Over** onto it fully. Default is 1.

Blend Modes

Since the blend modes are a per channel basis, the example images change the blend mode on the Base Color node unless otherwise specified.

Tip: The materials used in the examples are available in the **Materials** tab of the **Shelf** palette - **Bronze_Oxidized_Metal** as **Base**, **Bricks02** as **Over**. No mask is applied.

Normal	Displays the **Over** on top of the **Base**. The **Normal** mode is used for layering operations. You can adjust the **Amount** to set how much of the **Over** input to show. Decreasing the

	Amount allows you to show the **Base** input.
Copy	This copies the information from the **Over** input and doesn't involve any blending operations. **Note:** Adjusting the **Amount** has no effect.
Clear	Subtracts the opacity of the **Over** input from the **Base** input. It erases texture from the **Base** input using the alpha component of the **Over** input. For example, an **Over** input with 80% opacity leaves a **Base** input with 20% opacity.

Darken

Darken	Gives each pixel the darker of two possible values: the **Base** or the **Over** (whichever is darker).
Multiply	Darkens the color by multiplying the **Base** color information by the **Over** color information.
Color Burn	Burns the **Base** using the contents of the **Over** as input for the burn operation.

Lighten

Lighten	Gives each pixel the lighter of two possible values: the **Base** or **Over** (whichever is lighter).
Screen	Lightens the image by dividing the **Base** by the **Over**. **Note:** This is the opposite of **Multiply**.
Color Dodge	Dodges the **Base** using the contents of the **Over** to control the degree of dodging.
Add	Adds the values of the **Over** colors to the values on the **Base**.

Contrast

Contrast	The destination color moves away from the **Over** color by contrasting around the **Over** color. How much it contrasts depends on the alpha.
Decontrast	The destination color moves towards the **Over** color by decontrasting around the **Over** color. How much is decontrasted depends on the alpha.
Overlay	Uses the patterns and colors from the **Over**, but keeps the highlights and shadows from the **Base**.
Soft Light	In dark areas in the **Base** burns the **Over** onto the **Base**. In lighter areas, dodges it on.
Hard Light	Mimics a harsh spotlight. Makes light areas lighter using a screen effect, and dark areas darker using a multiply effect.
Vivid Light	In dark areas in the **Base**, adds the **Over** with more contrast. In light areas, adds it with less contrast.
Pin Light	For light areas in the **Over**, replaces pixels in the **Base** that are darker than the corresponding pixels in the **Over**. For dark areas in the **Over**, replaces pixels in the **Base** that are lighter than the corresponding pixels in the **Over**. Otherwise, leaves the **Base** unchanged.
Hard Mix	First performs a vivid blend, then thresholds the colors to their extreme values. The result is that the pixels can be one of eight colors - black, white, red, green, blue, cyan, magenta, or yellow.
Sponge Desaturate	Uses the **Over** as the input to desaturate the pixels on the **Base**.

Inversion

Difference	Inverts the colors by subtracting the values in the **Over** from those in the **Base**.
Exclusion	Inverts the colors, but with a lowered contrast.

Invert	Uses the **Over** as the input to invert the pixels on the **Base**.
Inverse Difference	Subtracts the colors in the **Over** from the **Base**, and inverts the result.

Burn

Burn Highlights	Burns the **Base** based on the highlights in the **Over**.
Burn Midrange	Burns the **Base** based on the middle range (that is, not the lightest or darkest pixels) of the **Over**.
Burn Shadows	Burns the **Base** based on the contents of the **Over**, with darker pixels being burned harder.

Dodge

Dodge Highlights	Dodges the **Base** based on the highlights in the **Over**.
Dodge Midrange	Dodges the **Base** based on the middle range (that is, not the lightest or darkest pixels) of the **Over**.
Dodge Shadows	Dodges the **Base** based on the contents of the **Over**, with darker pixels being dodged harder.

Component

Hue	Takes the luminance and saturation from the **Base** pixels, with the hue from the **Over**.
Saturation	Takes the luminance and hue of the **Base** color, with the saturation from the **Over**.
Color	Takes the luminance of the **Base**, and the color and saturation of the **Over**.
Luminance	Takes the hue and saturation of the **Base** color, and the luminance of the **Over**.

Note: This is the opposite of the **Color** blend mode.

Vectors

Paint Flow Vectors	Changes the **Over** to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. Unlike **Nudge Flow Vectors**, the **Paint Flow Vectors** mode blends between the vectors instead of adding them. This mode is best for use with flow maps.
Nudge Flow Vectors	Changes the **Over** to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. The **Nudge Flow Vectors** mode adds the vectors together to get the blend result, and is best for use with flow maps.
Paint Normal Vectors	Changes the **Base** to tangent vector space. Allows the Z value to face outward in 3D space, so that vectors can face out from the object's surface anywhere on the object. Unlike **Nudge Normal Vectors**, the **Paint Normal Vectors** mode blends between the vectors instead of adding them. This mode is best for use with normal maps.
Nudge Normal Vectors	Changes the **Over** to tangent vector space. Allows the Z value to face outward in 3D space, so that vectors face out from the object's surface anywhere on the model. The **Nudge Normal Vectors** mode adds the vectors together to get the blend result, and is best for use with normal maps.
Scale Vectors	This is most useful with the basic **Paint** tool, while preferably using grayscale paint. **Scale Vectors** uses the intensity of color to control the length of the normal in 3D space.

Normal Maps

Add Normal Maps	Blends two RGB-encoded vector layers with the **Add** input by converting the raw values from the 0.0 to 1.0 colorspace-encoded range, to the -1.0 to +1.0 vector range. Then applying the blend factor to the **Over** input, adding the values and normalizing the values, before converting it back to the 0.0 to +1.0 colorspace-encoded range.
Mix Normal Maps	Blends two RGB-encoded vector layers with the **Add** input by converting the raw values from the 0.0 to 1.0 colorspace-encoded range to the -1.0 to +1.0 vector range. Then applying the blend factor to the **Over** input, adding the values and normalizing the values,

	before converting it back to the 0.0 to +1.0 colorspace-encoded range.
Mix Reoriented Maps	Blends two RGB-encoded vector layers in a way that preserves the strength and details of both the **Base** and the **Over** inputs.

Multi-Channel Merge Node Workflow Example

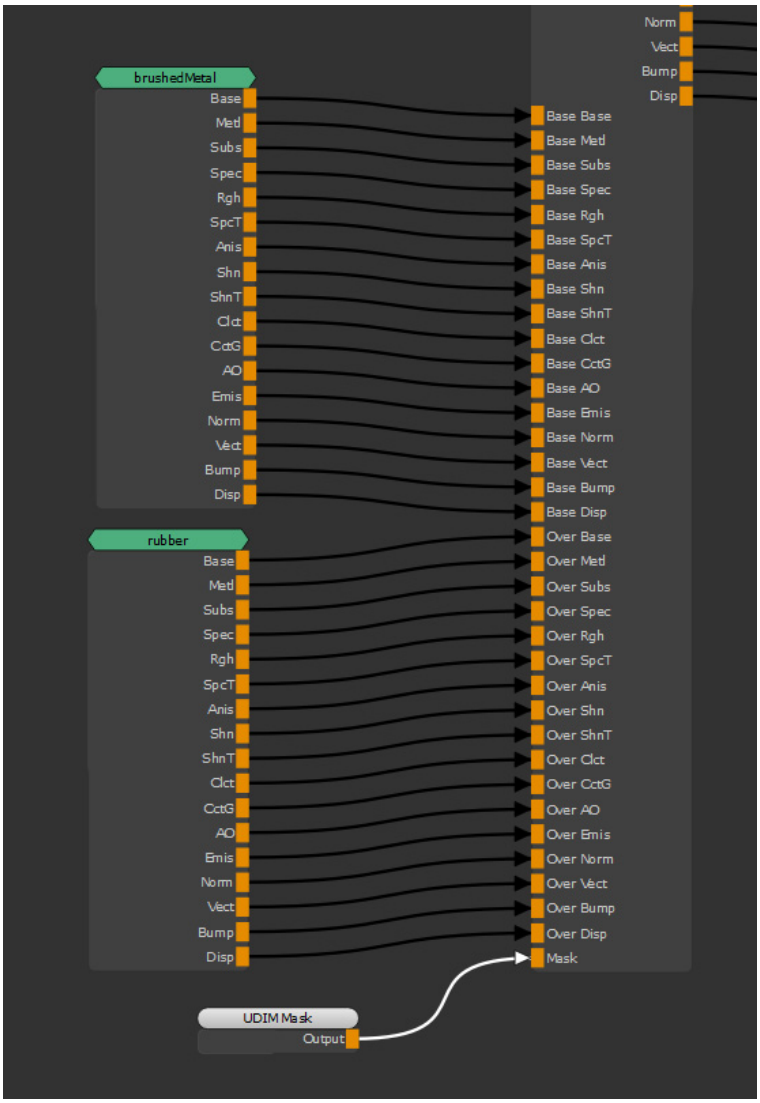
Setting up Materials with a Multi-Channel Merge Node

Working with the Multi-Channel Merge node and the Materials system allows you to quickly build up base textures for all your channels. In this example we will use three Material nodes to quickly get a watch asset filled in.

After creating your own materials you can save them to the **Shelf** to use later.

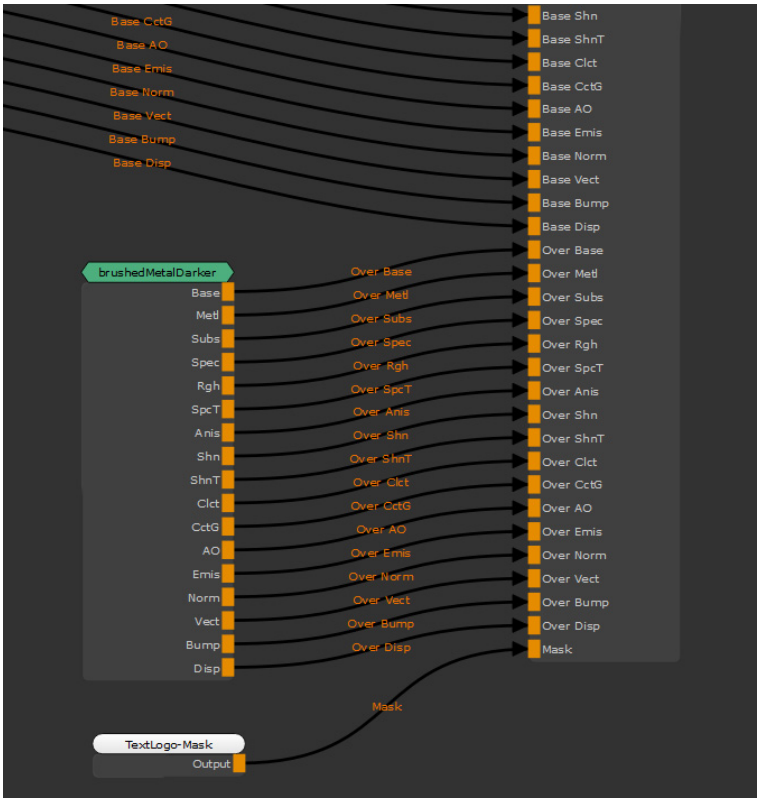
In this example, we are using custom built materials, but there are presets in the **Mari Materials** tab of the **Shelf** palette.

After placing down two Material nodes, you can add a Multi-Channel Merge node to combine them by selecting both Material nodes and pressing **M** on the keyboard.

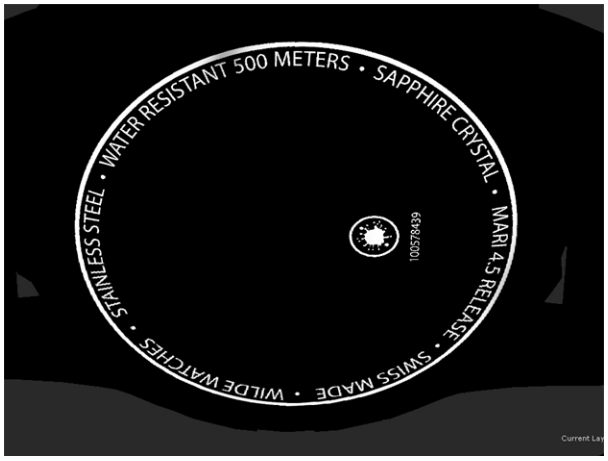


In this example, since the rubber strap is on its own UDIM, we can use a UDIM Mask node as the **Mask** input for the node. This merges the rubber over the brushed metal material on just the strap:

Lets add a third Material node. This is a copy of the base brush metal material, but darkened so that we can use for the text on the back of the watch.



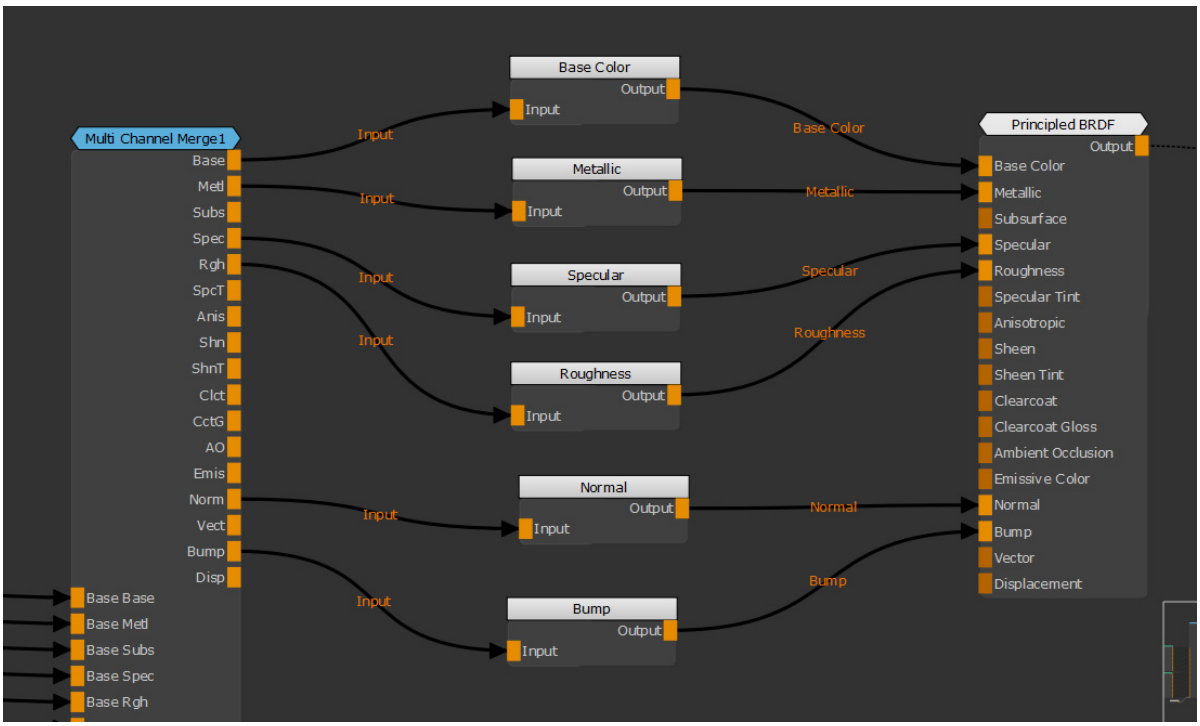
This time we will use a Paint node for the mask. By setting up a mask in an external graphic design program with text we can tell Mari where to make the metal darker. The mask in the example below has been imported into this Paint node. The **Bump** channel of this new material is also a darker value so it will push in, giving the impression of an engraving.



Mask

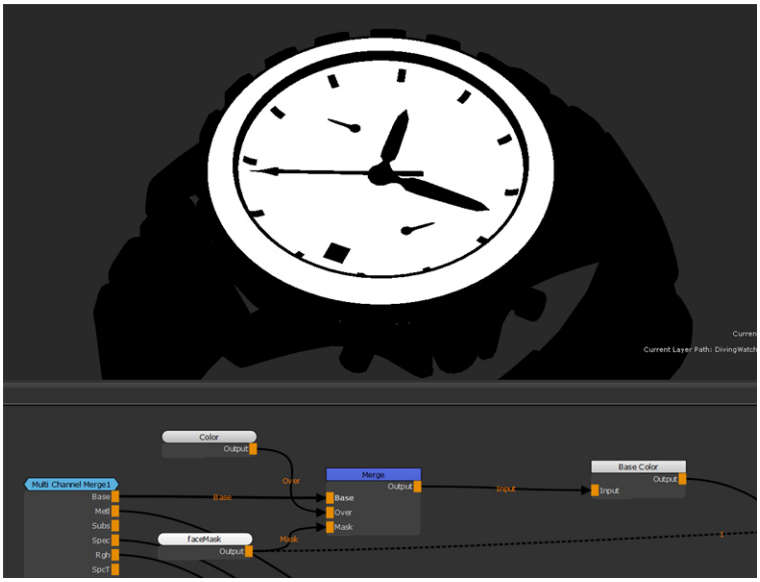
Result

Now we have base materials set up, we can use the outputs of the final Multi-Channel Merge node to input into the channels of our project.



Note: When connecting merge nodes to end point nodes that use 8-bit color data, a dialog appears asking if you would like to automatically enable **Use Blending Colorspace** on the merge node. See Use Blending Colorspace to learn more.

We can now add extra details between our Multi-Channel Merge node and the Channel nodes. We can add a darker color onto the face and dial of our mesh, like watches have in real life. By merging a color into the **Base Color** Node Graph with a mask we can add some extra variety. Below you can see the mask that the Color node in the following example is using.

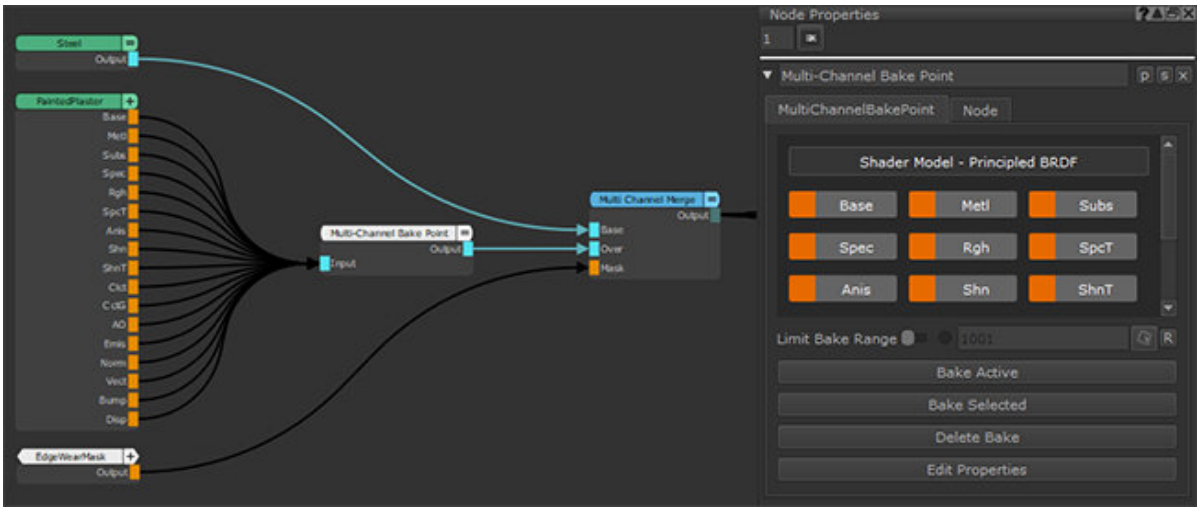


And here is the final result, you can see how a simple change like that makes it look better.

With the Multi-Channel Merge node we have been able to set up textures for every channel of our scene that we can then go and add extra details onto. This is the power a Material workflow.

Multi-Channel Bake Point Node

Access: *Right-click* | **Nodes** > **Layer** > **Multi-Channel Bake Point**



Using a Multi-Channel Bake Point node to bake multiple channels
and optimize materials in a scene

Multi-Channel Bake Point nodes work in the same way as Bake Point nodes except they allow you to bake multiple channels using a single node. Multi-Channel Bake Point nodes can be used to optimize your scene by baking materials once you have finalized them, preventing your project from slowing down as you continue to work.

Multi-Channel Bake Point nodes create individual Bake Points for all channels belonging to your chosen shader model. These individual Bake Points are nested within the Multi-Channel Bake Point node so that only one node is required to bake an entire material. Each nested Bake Point can be added individually as an Export Item in the Export Manager. By default, the nested Bake Points are named in the following format:

Bake Point (<SHORTNAME>**)**

Where <SHORTNAME> is the stream channel, for example:

- **Bake Point (Base)**
- **Bake Point (Spec)**

If you rename the Multi-Channel Bake Point node, the nested Bake Points are renamed to match. For example, if you rename the Multi-Channel Bake Point node to **PaintedPlaster_BP**, the **Bake Point (Base)** Bake Point is renamed to **PaintedPlaster_BP (Base)**.

Note: For more information, refer to the documentation for the Bake Point Node.

Bake Point Node Inputs

Input : Bakes the input.

The individual input channels depend on the selected shader model from the **Create Multi-Channel Bake Point Node** dialog.

Bake Point Node Outputs

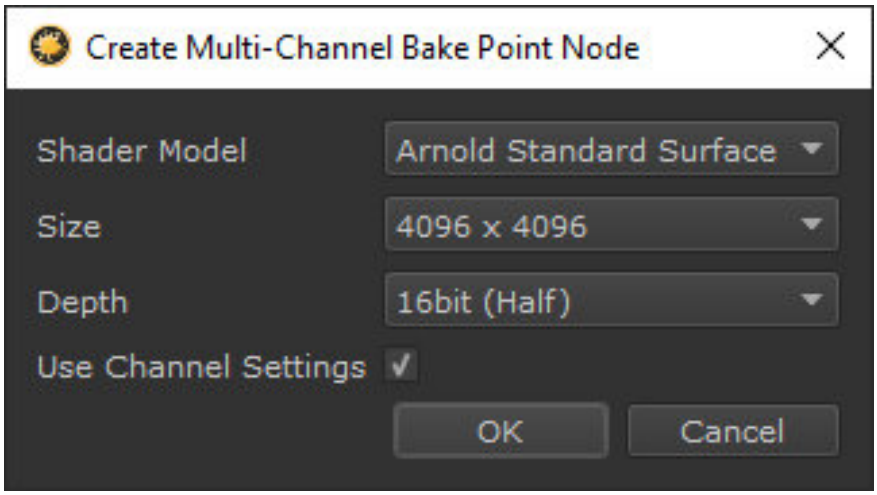
Output : The baked input.

The individual output channels depend on the selected shader model from the **Create Multi-Channel Bake Point Node** dialog.

Create Multi-Channel Bake Point Node Dialog

When you create a Multi-Channel Bake Point node, the **Create Multi-Channel Bake Point Node** dialog is displayed and you can choose the settings to apply to the node and the elements that are baked from it. After choosing the appropriate options, click **Okay** and the Multi-Channel Bake Point node is created.

The **Create Multi-Channel Bake Point Node** dialog is similar to the dialogs shown when creating Paint or Bake Point nodes.

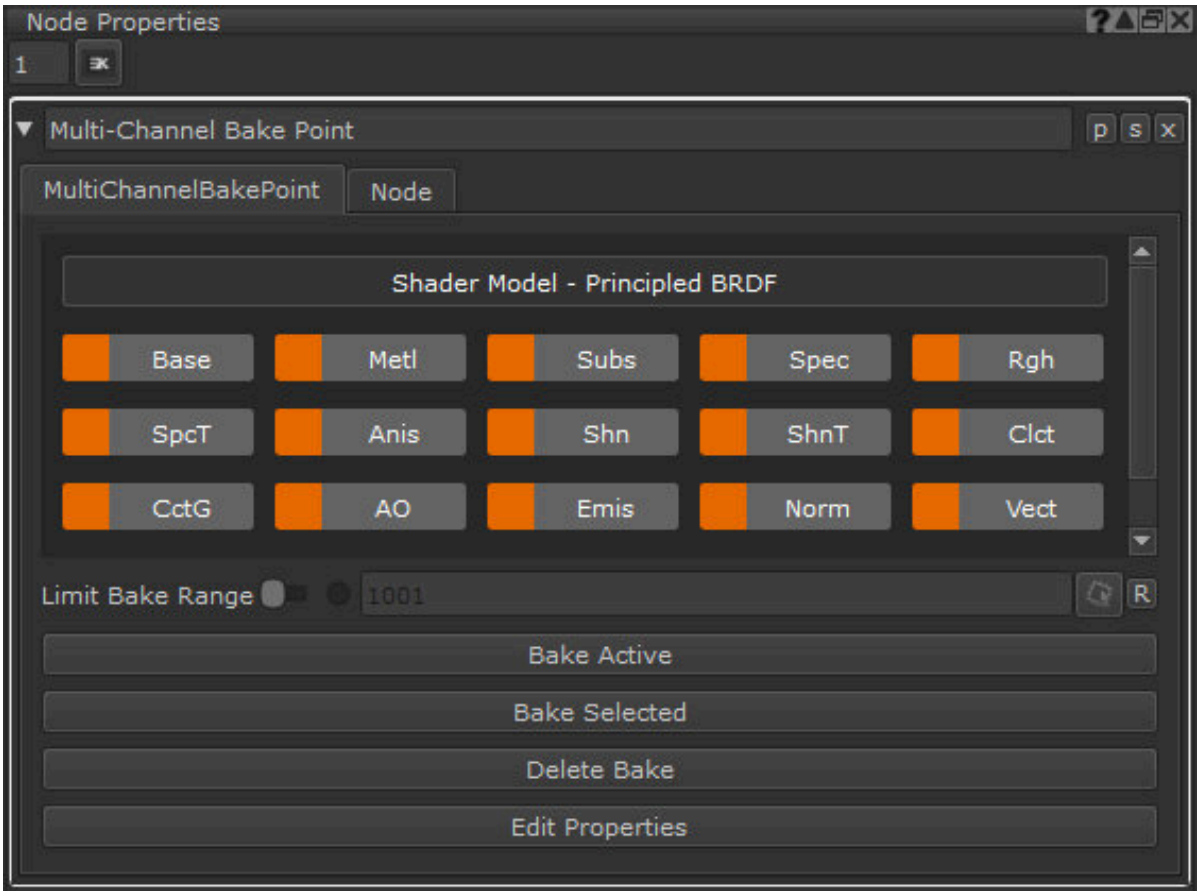


The **Create Multi-Channel Bake Point Node** dialog

Shader Model *dropdown*	Choose the shader model you are using in your scene so that the correct input and output channels can be assigned to the Multi-Channel Bake Point node. • **3Delight Principled** • **Arnold Standard Surface** • **BRDF** • **Principled BRDF** • **Unreal** • **VRayMtl** If you have selected a Multi-Channel node such as a Material before creating the Multi-Channel Bake Point node, then the **Shader Model** is pre-selected to match the Multi-Channel node. This is because the Multi-Channel Bake Point node needs to connect to the selected Multi-Channel node and therefore must be of the same **Shader Model**. In this case the **Shader Model** option is disabled.
Size	The patch size for the bake.

dropdown	• **256 x 256** • **512 x 512** • **1024 x 1024** • **2048 x 2048** • **4096 x 4096** • **8192 x 8192** • **16384 x 16384** • **32768 x 32768**
Depth *dropdown*	The color depth for the bake. • **8bit (Byte)** • **16bit (Half)** • **32bit (Float)**
Use Channel Settings *checkbox*	Enable this option if you want the **Size** and **Depth** to be inherited from an upstream Channel node instead of the specified settings in this dialog. If there is no upstream Channel node and this option is enabled, the specified settings in this dialog are used.

Multi-Channel Bake Point Node Properties

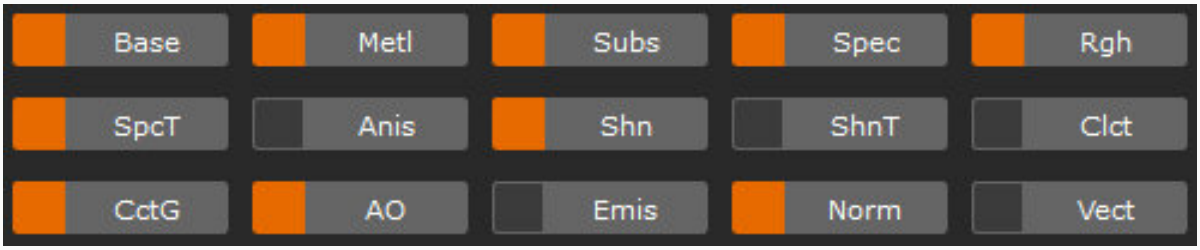
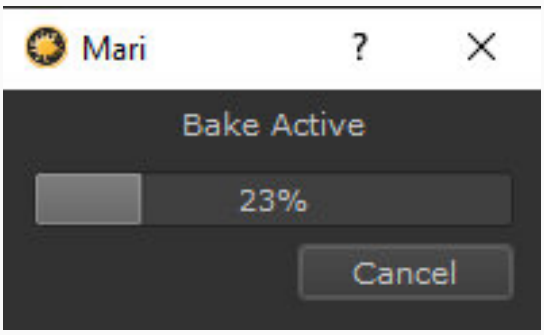
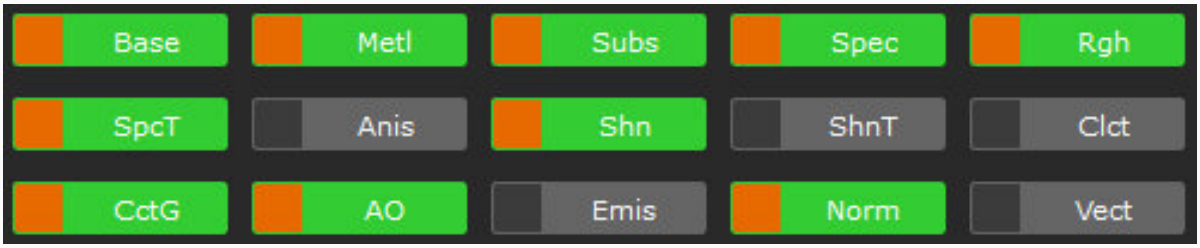


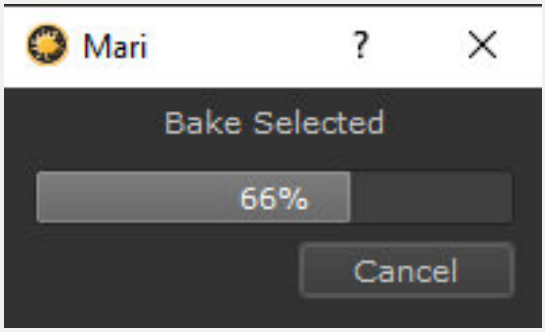
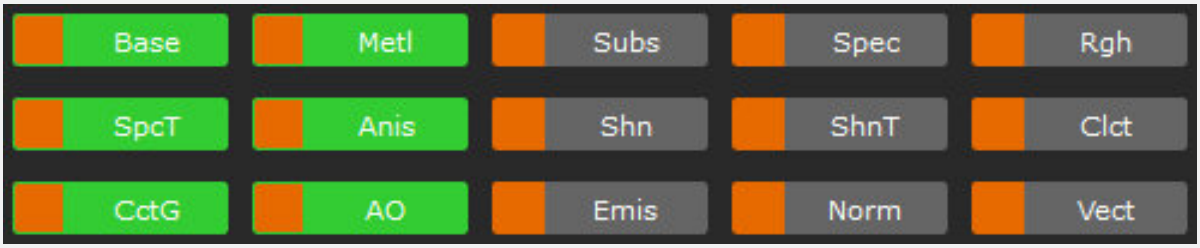
MultiChannelBakePoint Tab

Multi-Channel Properties	This section of the Multi-Channel Bake Point Node Properties allows you enable, disable, and select individual channels contained inside the Multi-Channel Bake Point node. From here you can choose which channels you want to create Bake Points for, by enabling those channels. If you only want to create Bake Points for a certain number of channels, you can select them in the **Multi-Channel Properties**, and use the **Bake Selected** button. All channels are enabled by default when a Multi-Channel Bake Point node is created. To disable a channel, click the orange button on the channel. Only enabled channels can be baked.

	Base channel enabled Base channel disabled To select a channel, click on its name. Hold **Ctrl** + Click to select multiple channels. When selected, channels are highlighted white. Base channel selected Once a channel is baked, it is highlighted green. If a baked channel is out of date because there have been upstream edits, that channel is highlighted red. Base channel baked Base channel bake out of date
Limit Bake Range *toggle*	Enable to only bake data within the specified **Patch Range** with this Multi-Channel Bake Point node, allowing you to reduce bake times by only baking required patches. Once turned on, a green icon is displayed and the **Patch Range** option is enabled. Turn off to bake all patches on your object.

Patch Range
 	
text field
	If **Limit Bake Range** is enabled, you can specify patches on your object to be baked by this Multi-Channel Bake Point node by typing either a single patch number, or a range of patches.
	
	For example, if you want to only bake patches 1001 to 1005, enter:
	`1001-1005`
	
	If you want to bake patch 1008, enter:
	`1008`
	
	If you want to bake patches 1003 to 1006, and patch 1009, enter:
	`1003-1006 1009`
	
	Click the **Patch Range Actions** button to display a menu of options for quickly setting the **Patch Range**:
	
	• **Set to Current Patch Selection** - Click to set the **Patch Range** to the patches that

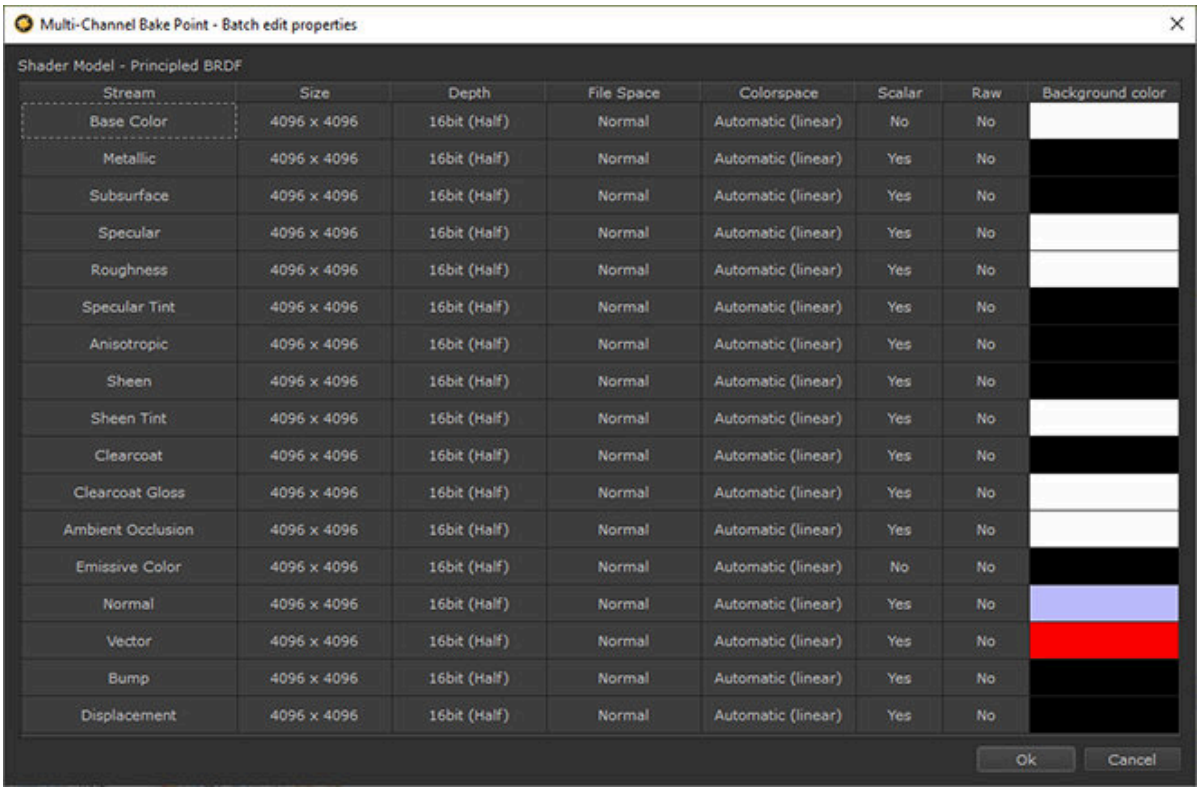
	are currently selected in the Patches palette. • **Set to Patch Selection Group** - Click to set the **Patch Range** to the patches that are in the selected Patch Selection Group. • **Select Patches in Range** - Click to select the patches defined in the **Patch Range** field in the Patches palette.
Bake Active *button*	Click to bake only the active, enabled channels connected to downstream shaders. A channel is active if there is a Channel node downstream. If a stream is disconnected from the Multi-Channel Bake Point node, or if there is no Channel node, that stream is not baked when using the **Bake Active** button.  The **Bake Active** dialog is displayed. Click **Cancel** to stop the bake.  Once active channels have been baked, they are displayed in green in the **Multi-Channel Properties**. 
Bake	Bake only channels that are selected and enabled in the **Multi-Channel Properties**.

Selected *button*	Selected channels are highlighted white. If a channel is selected but disabled, that channel is not baked. The **Bake Selected** dialog is displayed. Click **Cancel** to stop the bake.  Once the selected channels have been baked, they are displayed in green in the **Multi-Channel Properties**. 
Delete Bake *button*	Click to clear any information held in the Multi-Channel Bake Point node. Once the Bake is deleted, the channels turn gray in the **Multi-Channel Properties** and the Multi-Channel Bake Point node returns to its default color.
Edit Properties *button*	Click to open the **Multi-Channel Bake Point - Batch edit properties** dialog. From here you can edit the individual Bake Points nested within the Multi-Channel Bake Point node. For more information, see the Multi-Channel Bake Point - Batch Edit Properties Dialog section in this topic.

Node tab

Description *text field*	Write a custom description for this node to explain its function.
Export as Gizmo *button*	Click to export this node as a gizmo in the **.mng** file format, so that other artists can use it to perform similar tasks. For more information, see Exporting Groups as Gizmos.

Multi-Channel Bake Point - Batch Edit Properties Dialog

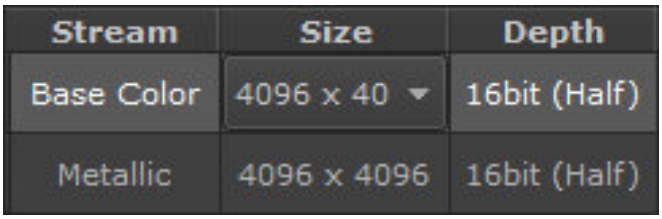


Stream	Size	Depth	File Space	Colorspace	Scalar	Raw	Background color
Base Color	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	No	No	
Metallic	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Subsurface	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Specular	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Roughness	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Specular Tint	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Anisotropic	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Sheen	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Sheen Tint	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Clearcoat	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Clearcoat Gloss	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Ambient Occlusion	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Emissive Color	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	No	No	
Normal	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Vector	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Bump	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	
Displacement	4096 x 4096	16bit (Half)	Normal	Automatic (linear)	Yes	No	

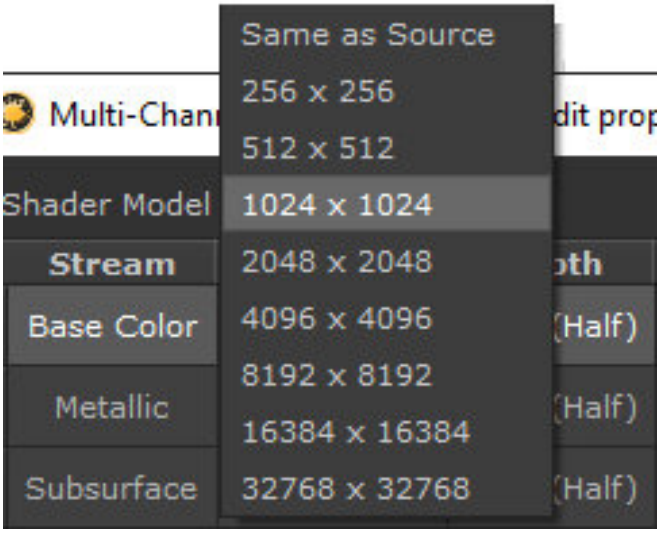
You can use this dialog to make batch edits to the individual Bake Points contained within a Multi-Channel Bake Point node. The Bake Points for each stream are listed on the left, and the properties you can edit are along the top.

To make an edit:

1. Double click on the option you want to change and a dropdown menu is displayed.



2. Click the dropdown menu and choose an option.

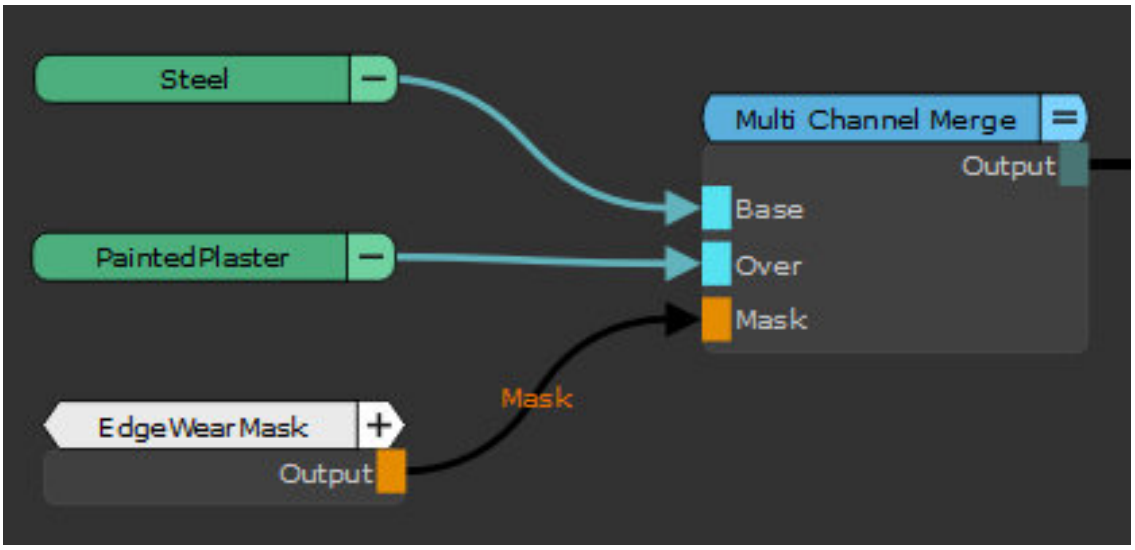


Size *dropdown*	The patch size for the bake. See Create Multi-Channel Bake Point Node Dialog.
Depth *dropdown*	The color depth for the bake. See Create Multi-Channel Bake Point Node Dialog.
File Space *dropdown*	This defines how values are imported or exported. • **Normal** - nothing is changed. • **Vector** - the data is treated as vector data and is remapped. • **Vector (flipped Y)** - the data is treated as vector data and is remapped. The Y component is inverted. • **Normal (flipped Y)** - nothing is changed except the Y component is inverted.
Colorspace *dropdown*	If the **Raw** option is set to **No** then the colorspace of the data stored in the bake is converted to the specified **Colorspace**.

Scalar *dropdown*	Enable this option to indicate that pixel data is not affected by color management. Enabling this changes the view transform to **None**. This option can be useful when working with channels that don't use color data, for example, bump or metallic maps.
Raw *dropdown*	Specify if the data is raw or is in a different colorspace. Disabling this option allows you to specify the colorspace using the **Colorspace** option, resulting in a conversion to the bake.
Background Color *dropdown*	Specify a background color onto which the baked image is composited. **Note:** Patches outside the specified **Patch Range** have a uniform, transparent background and are not affected by the **Background Color**.

Multi-Channel Bake Point Node Workflow Example

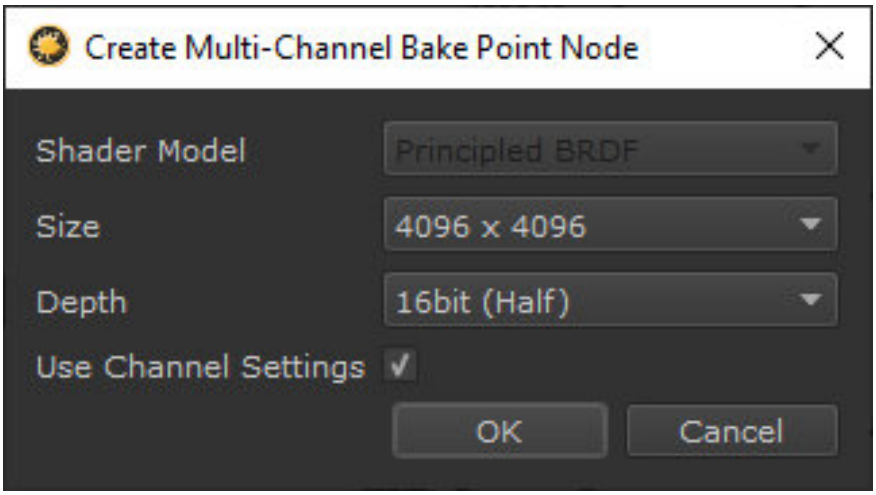
This example demonstrates how to place a Multi-Channel Bake Point node to bake certain channels from a material named **PaintedPlaster**, and optimize the scene so that further edits can be made to other materials more quickly.



Tip: This workflow example uses the Example Material Design Project that ships with Mari.

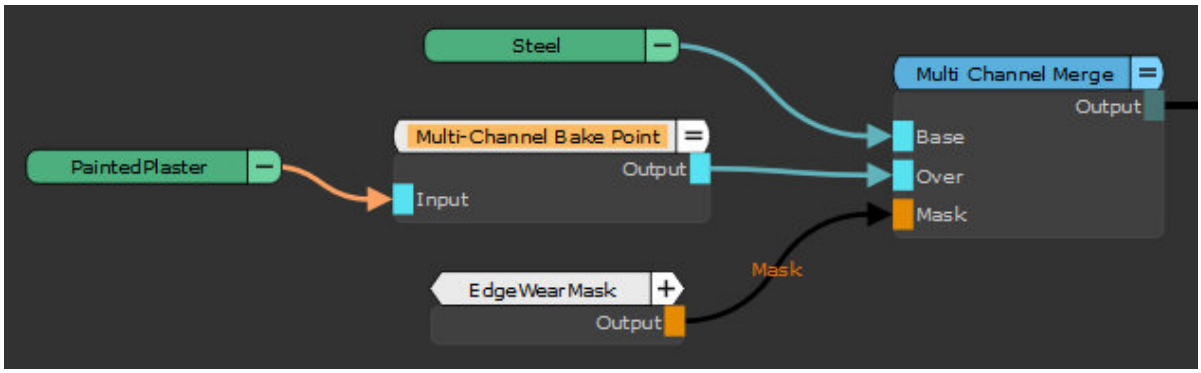
1. In the Node Graph, select the **PaintedPlaster** material.
2. Press **Tab** and type **Multi-Channel Bake Point**, then press **Enter**.

   The **Create Multi-Channel Bake Point Node** dialog opens. The **Shader Model** is pre-selected as **Principled BRDF** to match the material.



3. Choose the patch **Size** and color **Depth** to use for the bake.
4. Click **OK**.

   A Multi-Channel Bake Point node is created and is connected to the **PaintedPlaster** Material node.



> **Tip:** Click the button to the right of the node name to expand the Multi-Channel Bake Point node and you'll see the inputs and outputs for each channel.
> For more information about expanding and collapsing nodes, see Collapsing Nodes.

5. Double-Click on the node to open the **Node Properties**.
6. In the **Multi-Channel Properties**, hold **Ctrl/Cmd** and click the names of the channels you want to bake.

7. Click **Bake Selected**.

   The selected channels are baked.

Note: For more workflow examples, you can also refer to the Bake Point Node Workflow Examples

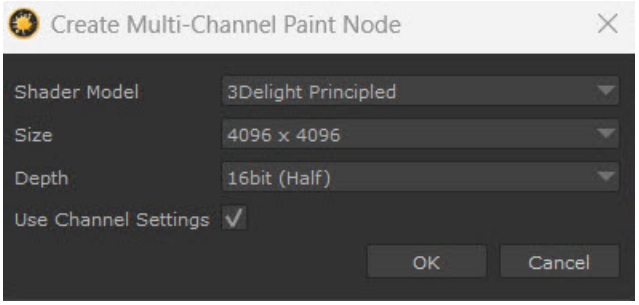
Multi-Channel Paint Node

Access: **Nodes** > **Misc** > **Multi-Channel Paint**

The Multi-Channel Paint node enables you to paint into up to eight different streams simultaneously from a single node. This means you can define images or colors for your individual texture streams, and paint into those channels simultaneously.

This means when you're using the Multi-Channel Paint node, you don't have to manually align your diffuse with your displacement and bump. This can help you save time when painting using texture packs.

Multi-Channel Paint Node Properties



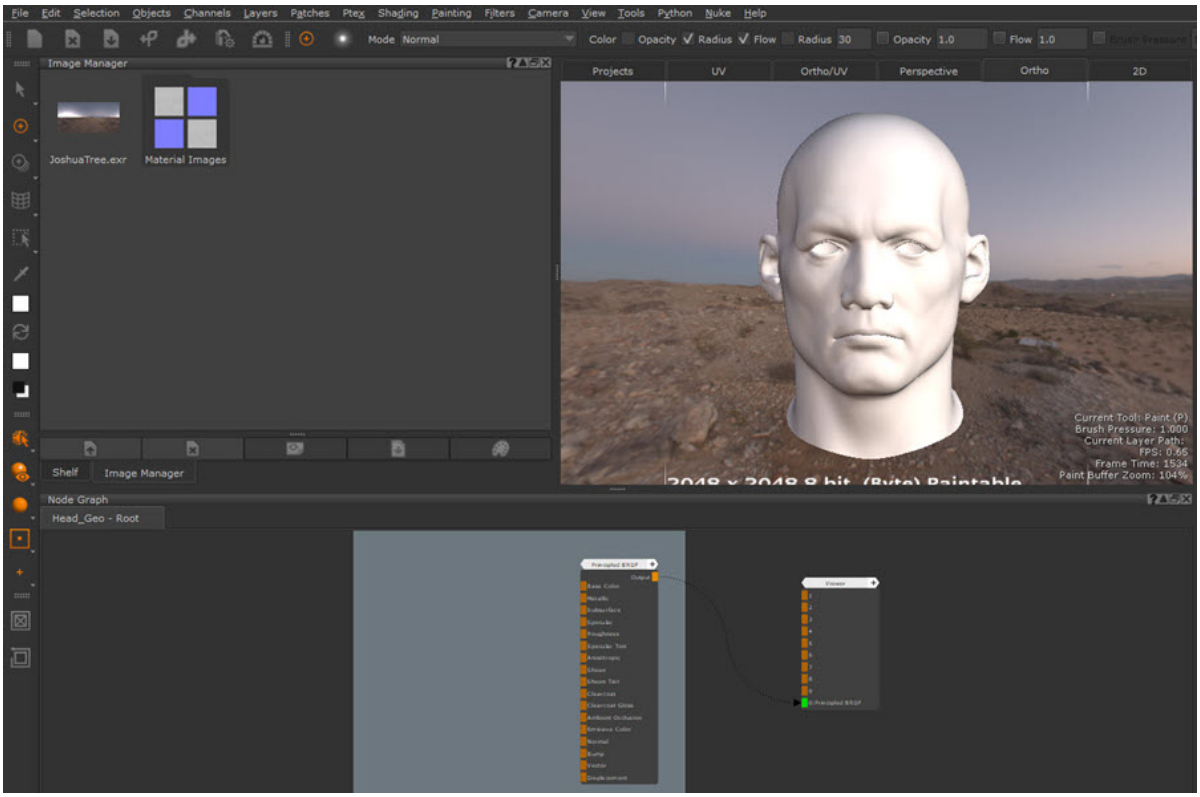
When you create a Multi-Channel Paint node, a UI dialog will pop up. This defines the settings for the Multi-Channel Paint node you're creating.

Shader Model *Dropdown list*	This option defines the Shader Model that the node will work in. The available streams to paint into will reflect those of the shader model you choose. The available shader options are: • 3Delight Principled • Arnold Standard Surface • Autodesk Standard Surface (1.0.2.x) • BRDF • Principled BRDF • USD Preview Surface • Unreal • VRayMtl • VRayMtl (V-Ray 6)
Size *Dropdown list*	Chooses the resolution of the images when exported.
Depth *Dropdown list*	Chooses the bit depth of the images that get exported.
Use Channel Settings *Tickbox*	For streams with a Channel node downstream, use its size and depth instead of the specified settings above.

Multi-Channel Paint Node Workflow Example

Importing External Textures into a Multi-Channel Paint Node

You can use the Multi-Channel Paint node to paint external images simultaneously into a model. For example, if you have a TexturingXYZ texture set that has both color and displacement images in it, you can import these into Mari to use in the Multi-Channel Paint node.



First, import the texture pack into Mari. To do this, drag the texture pack into your **Image Manager**.

The texture pack will automatically form an **Image Group** from the folder you dragged into Mari. When you click on the image groups, the Ingest Template dropdown list will appear. Learn more about the Ingest Template at Ingest Template Dropdown List.

You can auto assign the images into the Ingest template, or use the dropdown to specify which Image you want for each stream. This will mean when you import the images into the Multi-Channel Paint node, the images will automatically be added to the specific streams you have assigned to them.

> **Note:** You don't have to pre-define the streams in your Image Group. You can directly assign them to the Multi-Channel Paint node, described below.

To learn more about Image Groups, read Image Groups.

Setting Up Your Multi-Channel Paint Node

To prepare your Multi-Channel Paint node ready for painting:

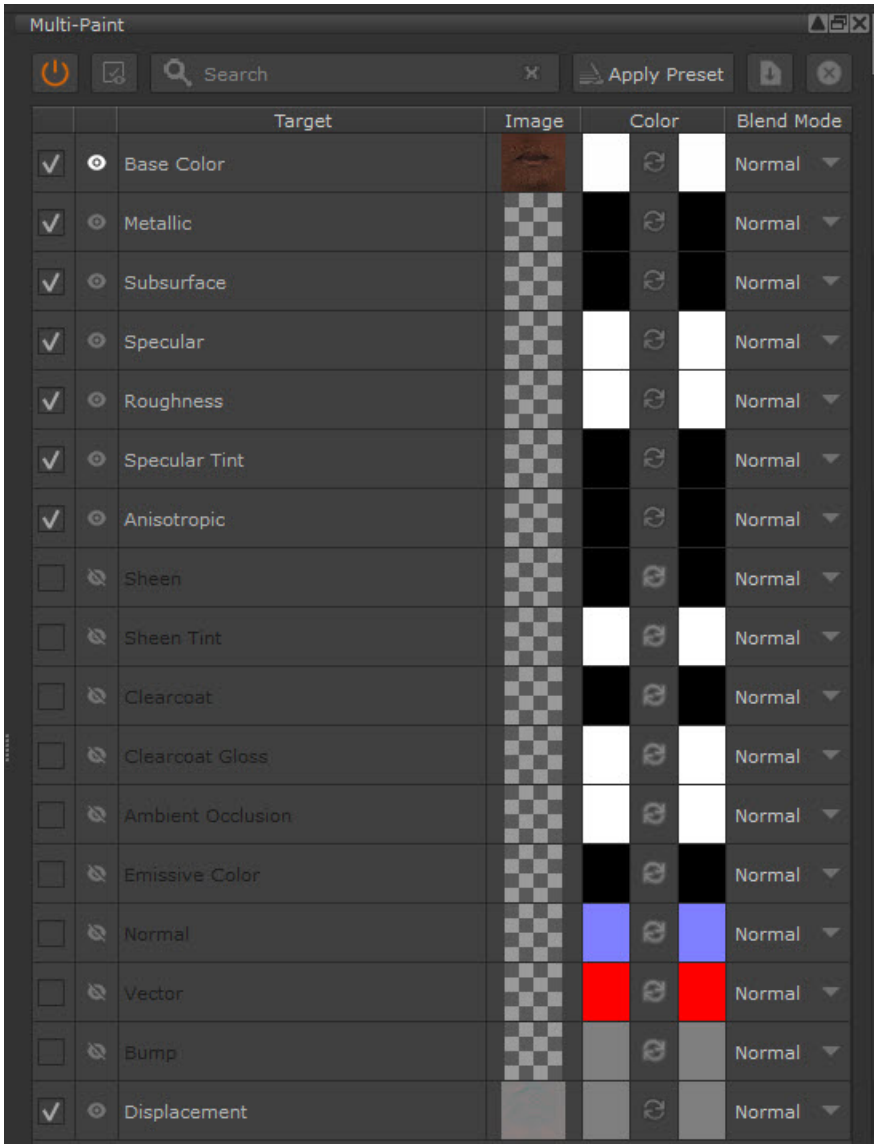
1. Create a Multi-Channel Paint node in your Node Graph.

2. In the pop up, select the shader model you want to work in. You can also define your Resolution and bit Depth.

> **Note:** To ensure the node works with your current shader, make sure the shader model matches the others used in your project.

3. With the node selected, go to the Multi-Paint Palette.
4. Edit the streams you want to paint into. If you have pre-assigned images to specific channels in the same shader, you can drag the Image Group into the Multi-Paint palette.

   Alternatively, drag specific images from the Image Manager into the stream you want to paint into, or click the Image column for the channel and choose the image.



Tip: You can customize the Multi-Channel Paint node colors in **Edit** > **Preferences** > **Node Graph** > **Node Category Colors**.

Multi-Paint using the Paint-Through Tool

1. Connect the node to your node graph and select it as the active paint target.
2. Select the **Paint Through** tool. The primary stream image will appear in the Viewer. Match up the image with where you want to paint on the model (for more information on Paint-through, go to "Painting Through" an Image).

Note: When using the Paint-through tool, you can only view one channel at a time - the Primary channel. To change the Primary Channel, click the eye icon ( ) next to the specific channel in the Multi-Paint palette.

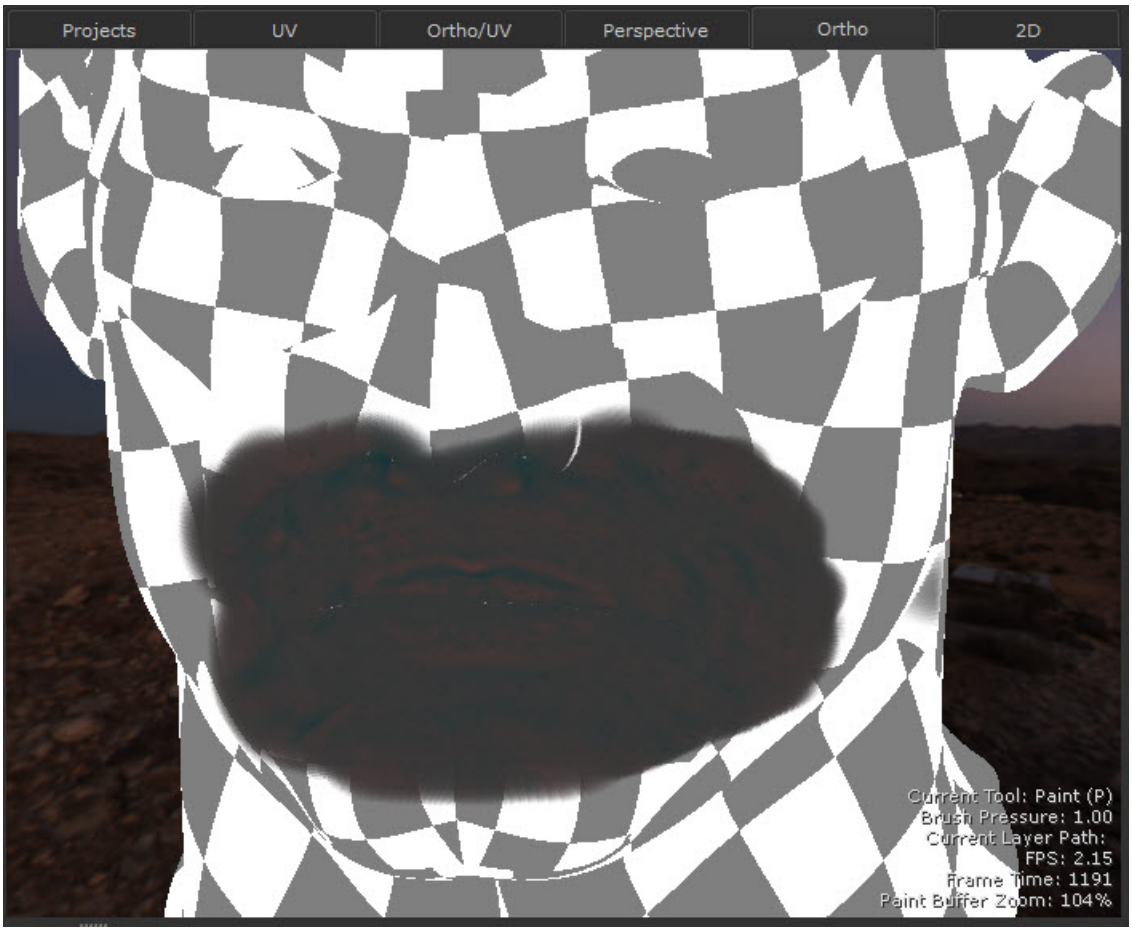
3. Paint onto your model.

Note: When you select a Multi-Channel Paint node as your active paint target, it automatically activates multi-channel painting. If you switch to a Paint node, it will switch back to single-channel painting.

Whilst we are painting on the Primary stream, you are also painting on the other streams you've set up in the Multi-Paint palette. For example, here is the Base Color and Displacement Red information from the same painting session:

Base color:

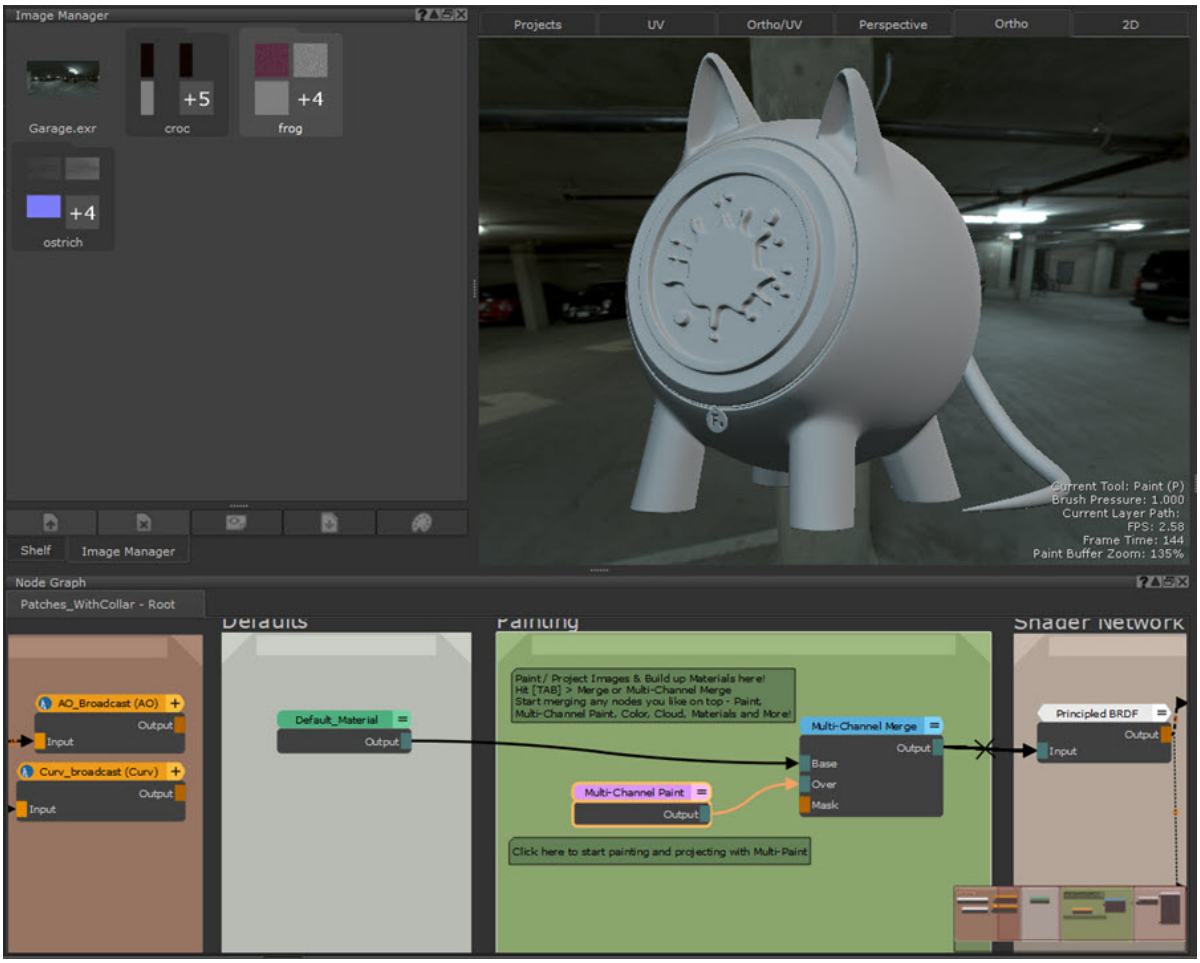
Displacement (Red):



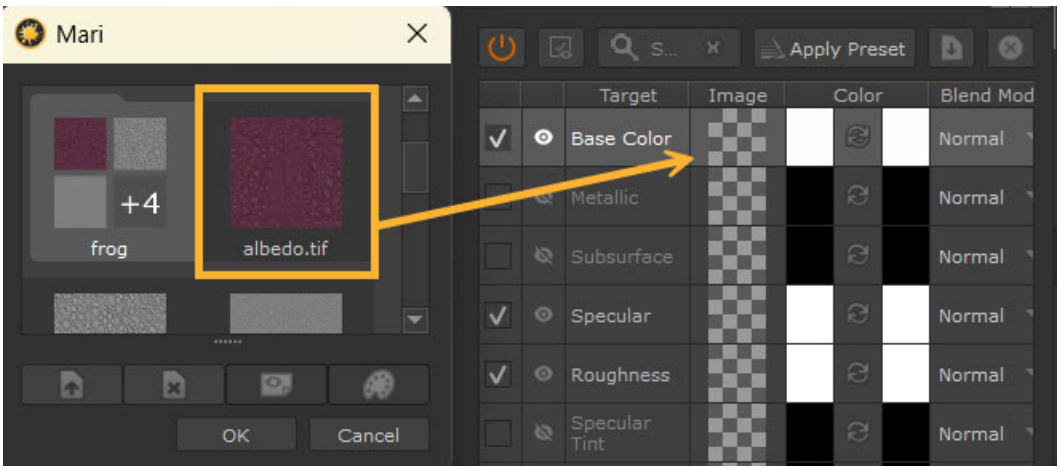
Using the Roller Tool to Multi-Paint the Patches Example Project

Mari 7.5 introduces **Patches**, a new example project for you to experiment with Multi-Paint (and any other tool in Mari you want to try). To access this project, go to **Help >Example Projects > Create Patches ShaderCat Project**.

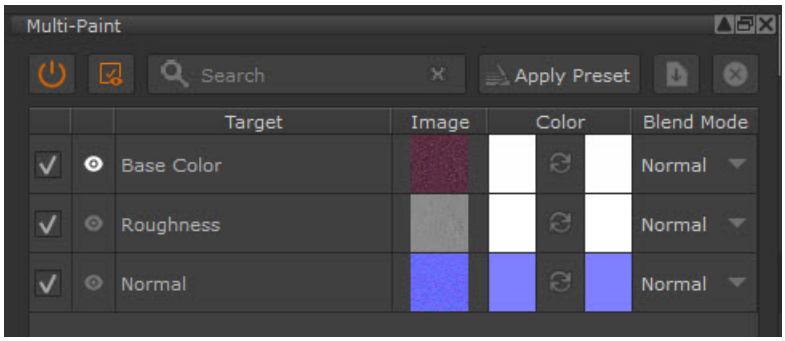
This workflow example uses Patches to demonstrate Multi-Painting using the Roller Tool.



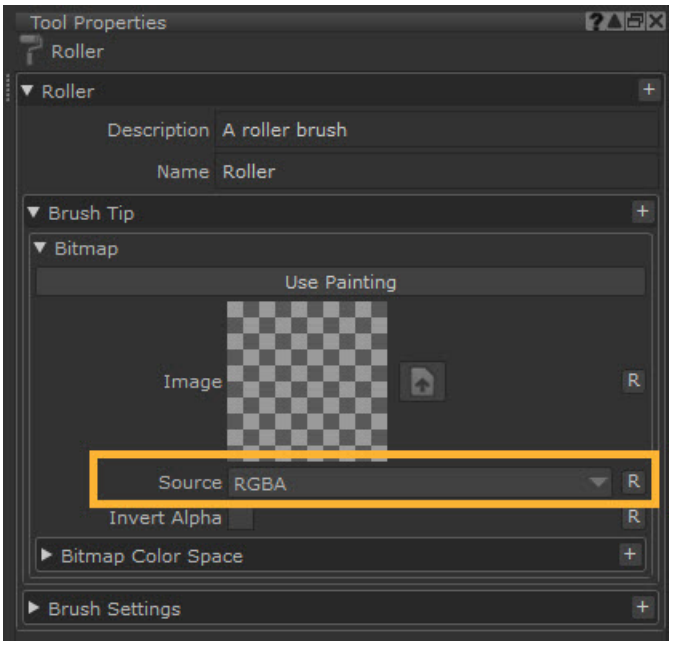
1. In the Node Graph, create a **Multi-Channel Paint** node.
2. Open the **Multi-Paint** Palette.
3. Select the images you want to paint into for each stream. Double-click on the **Image** icon ( ) of your chosen stream in the Multi-Paint Palette, and choose an image from the window that appears. For example, the albedo image will be assigned to the **Base Color** stream.



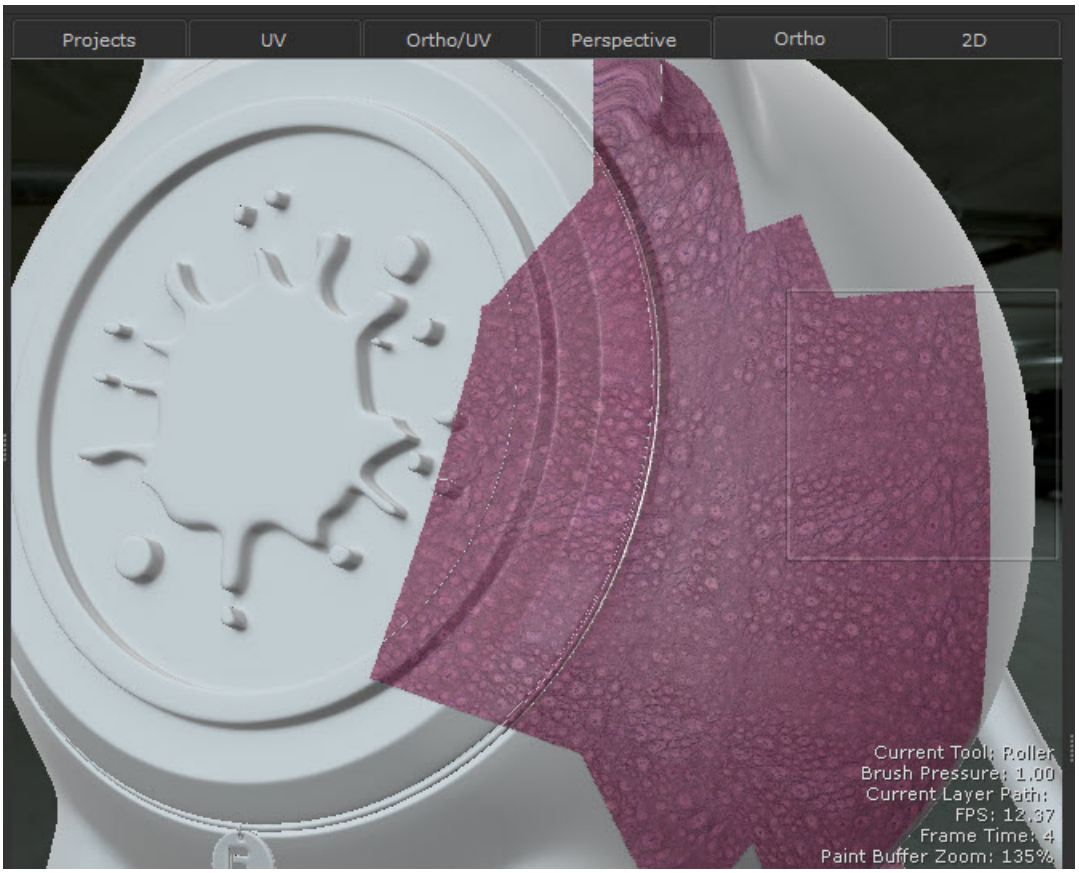
The **Roughness** and **Normal** channels will also be painted into, so those images are also assigned to streams.



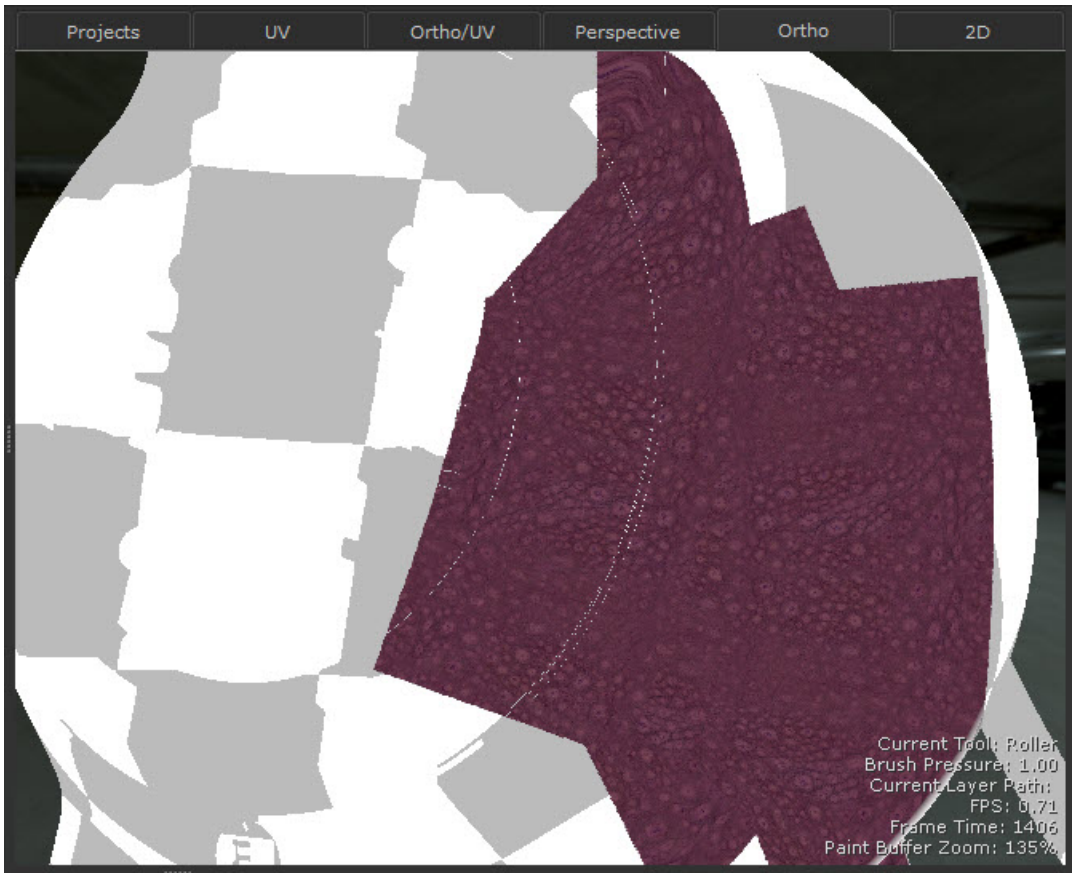
4. Connect the node to your node graph and select it as the active paint target.
5. Select the **Roller Brush** tool.
6. With the Roller Brush selected, go to **Tool Properties** > **Brush tip** > **Source** and change the value from **Alpha** to **RGBA**.

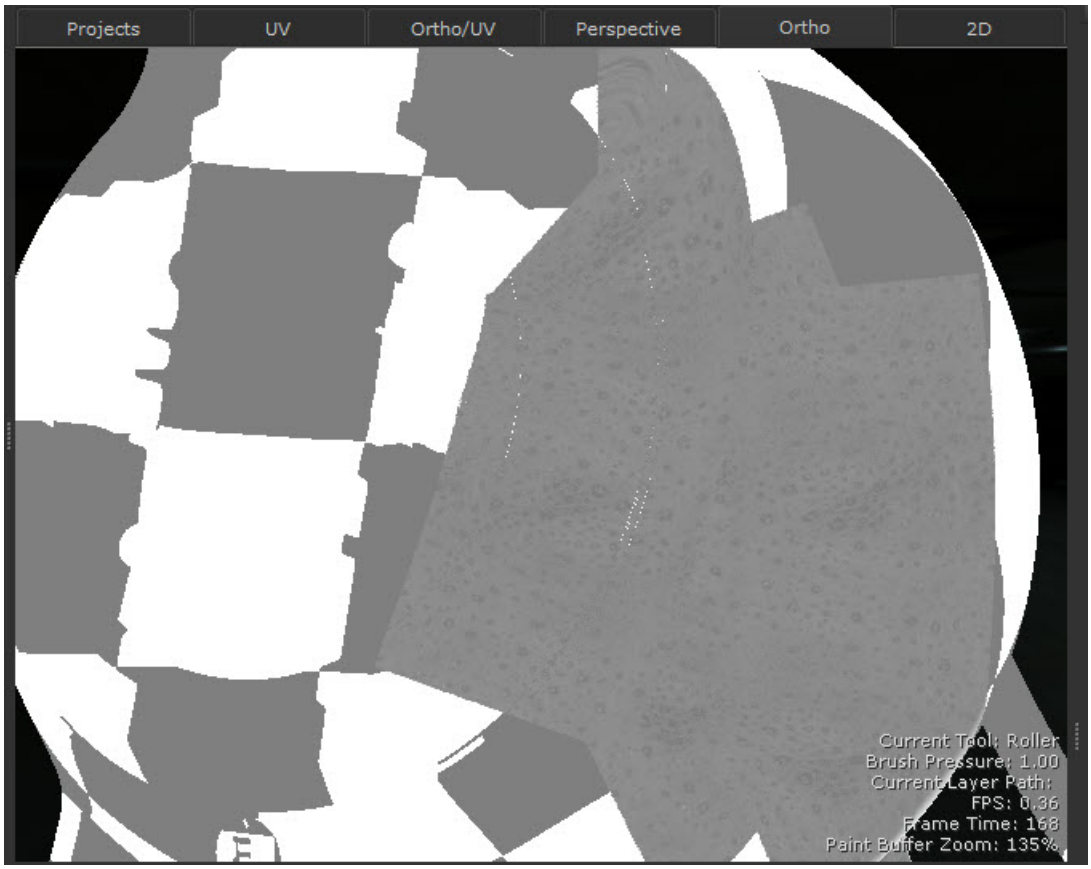


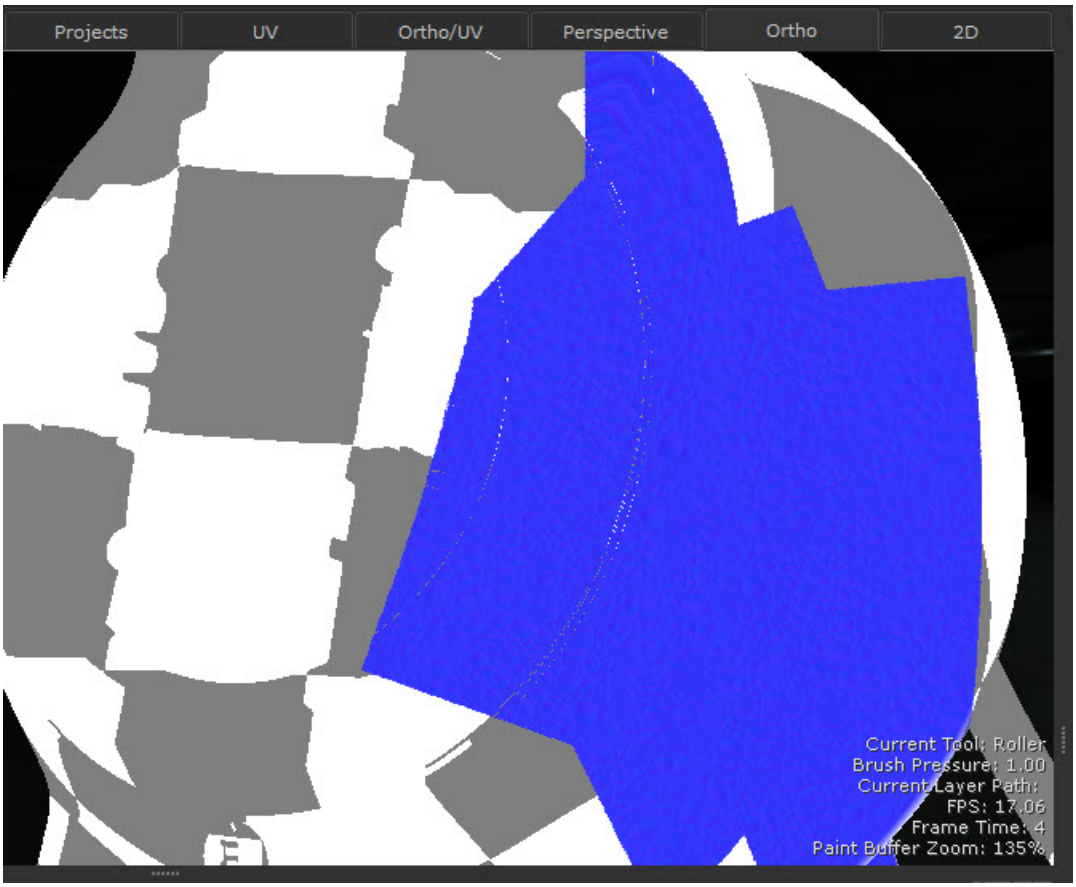
7. Start painting onto your model. The Primary Stream will be the only one visible, but all streams will be painted simultaneously.



The different streams are painted simultaneously:


Projects
UV
Ortho/UV
Perspective
Ortho
2D
Current Tool: Roller
Brush Pressure: 1.00
Current Layer Path:
FPS: 0.71
Frame Time: 1406
Paint Buffer Zoom: 135%


Projects
UV
Ortho/UV
Perspective
Ortho
2D
Current Tool: Roller
Brush Pressure: 1.00
Current Layer Path:
FPS: 0.36
Frame Time: 168
Paint Buffer Zoom: 135%

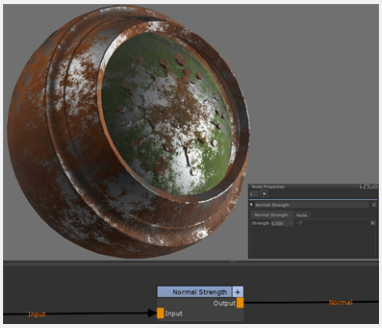


Normal Strength Node

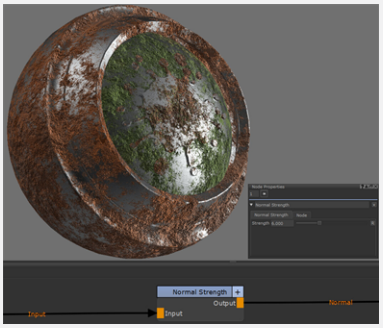
Access: **Nodes** > **Filter** > **Normal Strength**

The Normal Strength node adjusts the strength of the input tangent space normal map by blending it towards or away from a flat normal. The higher the strength, the further the input normal points from a flat surface and the more extreme the resulting normal map becomes.

This node can be used to make adjustments to a normal map without having to make any changes to the map itself. It is a good way of iterating and comparing various strengths.



Normal Strength with a value of 0.5.



Normal Strength with a value of 6.

Normal Strength Node Inputs

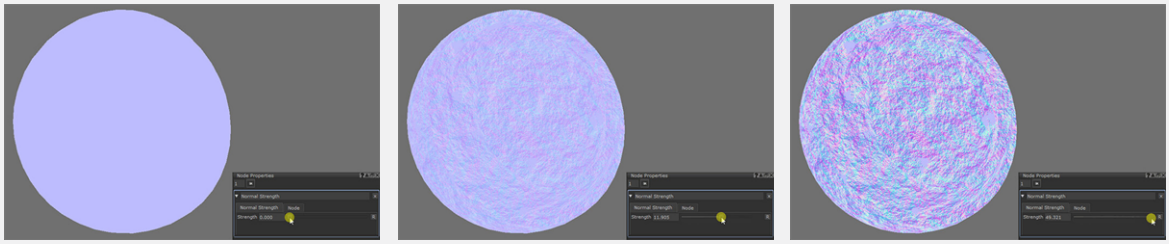
Input	Description
Input	The normal map to input.

Normal Strength Node Outputs

Output	Description
Output	The normal map that is output after the strength has been adjusted.

Normal Strength Node Properties

Strength *text field, slider*	The strength of the normal map. A value of **0** outputs a flat normal, value of **1** outputs the same normal as was input.

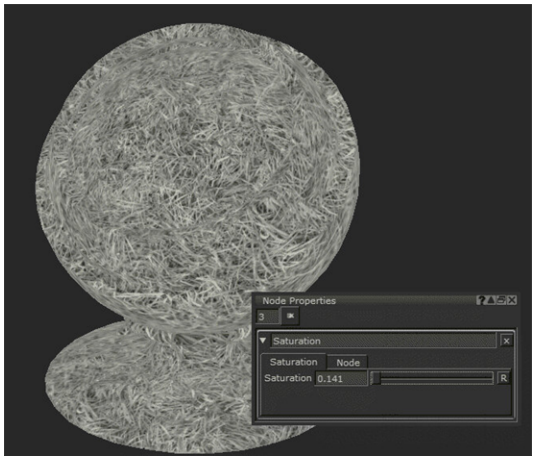


Strength: 0 | **Strength**: 11.905 | **Strength**: 49.321

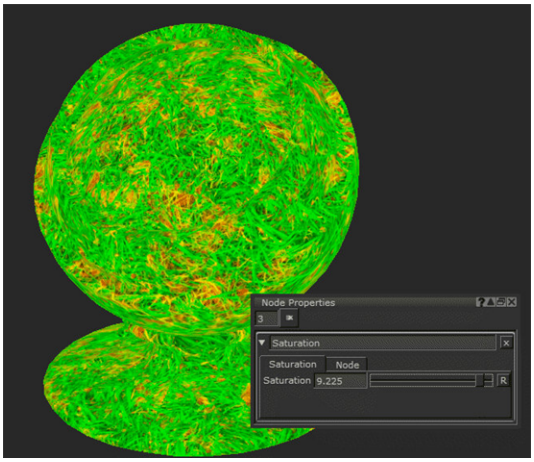
Saturation Node

Access: **Nodes** > **Filter** > **Saturation**

The Saturation node allows you to edit your inputs saturation.



Saturation node on a texture with a **Saturation** input of 0.1.



Saturation node on a texture with a **Saturation** input of 9.2.

The Saturation node is a useful node for situations where you have imported images that don't quite match your textures because they have too much or too little saturation. You can also use it to create a black and white image for masking from an image with color data.

Saturation Node Inputs

Input: Saturation node adjusts this input.

Saturation Node Properties

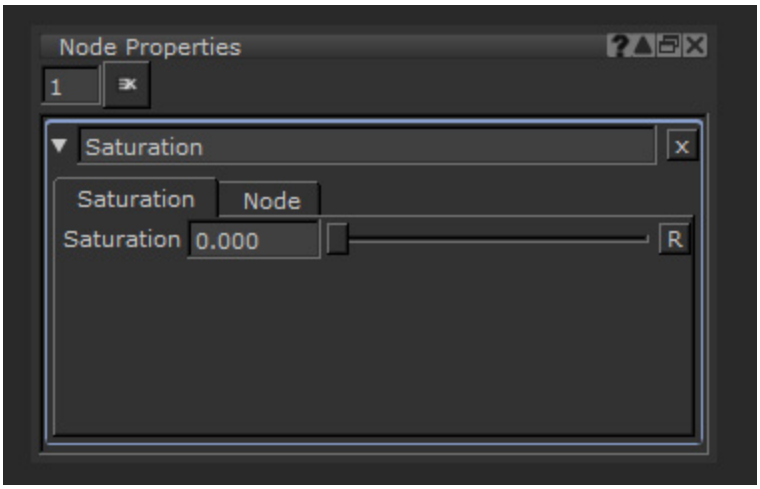
Saturation *text field, slider*	A modifier to the amount of saturation the output has. Moving the slider to the left down to 0 will turn your input into a greyscale texture, where as sliding it right will make the texture more saturated. Default is 1.

Saturation Node Workflow Example

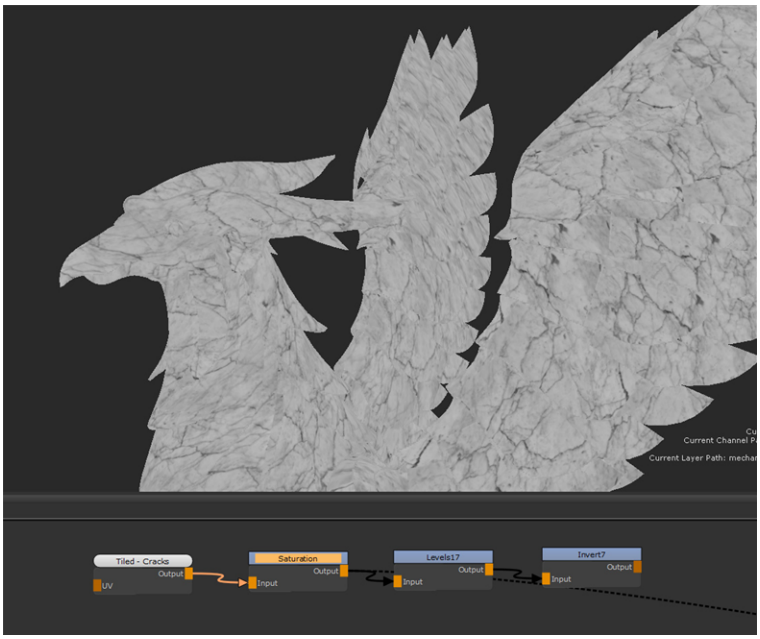
Desaturating to make a Mask

To create a mask from an image in your **Image Manager**, you can use a Saturation node to desaturate it. In the example is an image of marble which has been desaturated. It’s a great texture to make a crack mask out of, let's break down how that has been achieved .

Using the Saturation node lets you change **Saturation** to 0, resulting in a grayscale version of the image.



The value of the Saturation node.

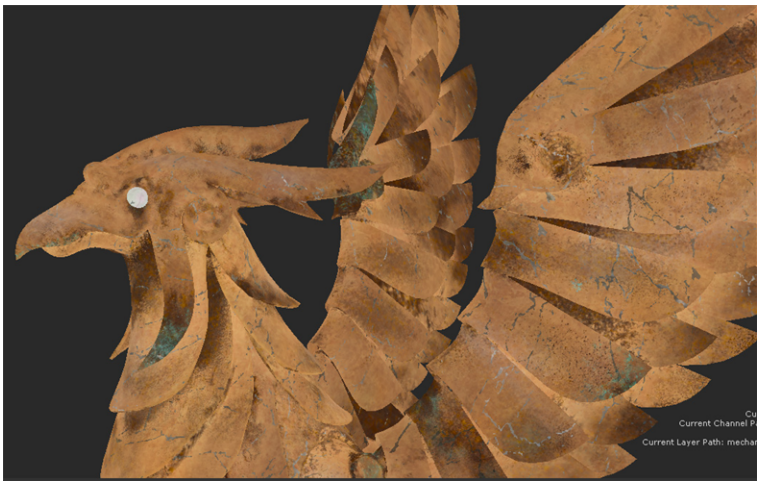


The desaturated Tiled image.

You can then use a Levels node to add contrast to the image to get just the cracks to show.

Then, an Invert node can be used if you want to use the cracks as a mask. This image is now ready to be used for anything. You could use it as a mask for another set of nodes, or merge it straight on top of your textures.

In this example, it has been used with a Merge node to add another tiled texture over the current base color.



Using the newly created mask to add a grunge texture in a cracked formation.

Using imported images in multiple ways, like making a mask from image with a Saturation node is a great way to keep your scene cleaner and helps you be more versatile as a texture artist!

Scalar to Color Node

Access: **Nodes** > **Filter** > **Scalar to Color**

The Scalar to Color node converts scalar RGB values to color RGB values, which can be useful for blending scalar data, such as a roughness map, into a color channel.

Scalar to Color Inputs

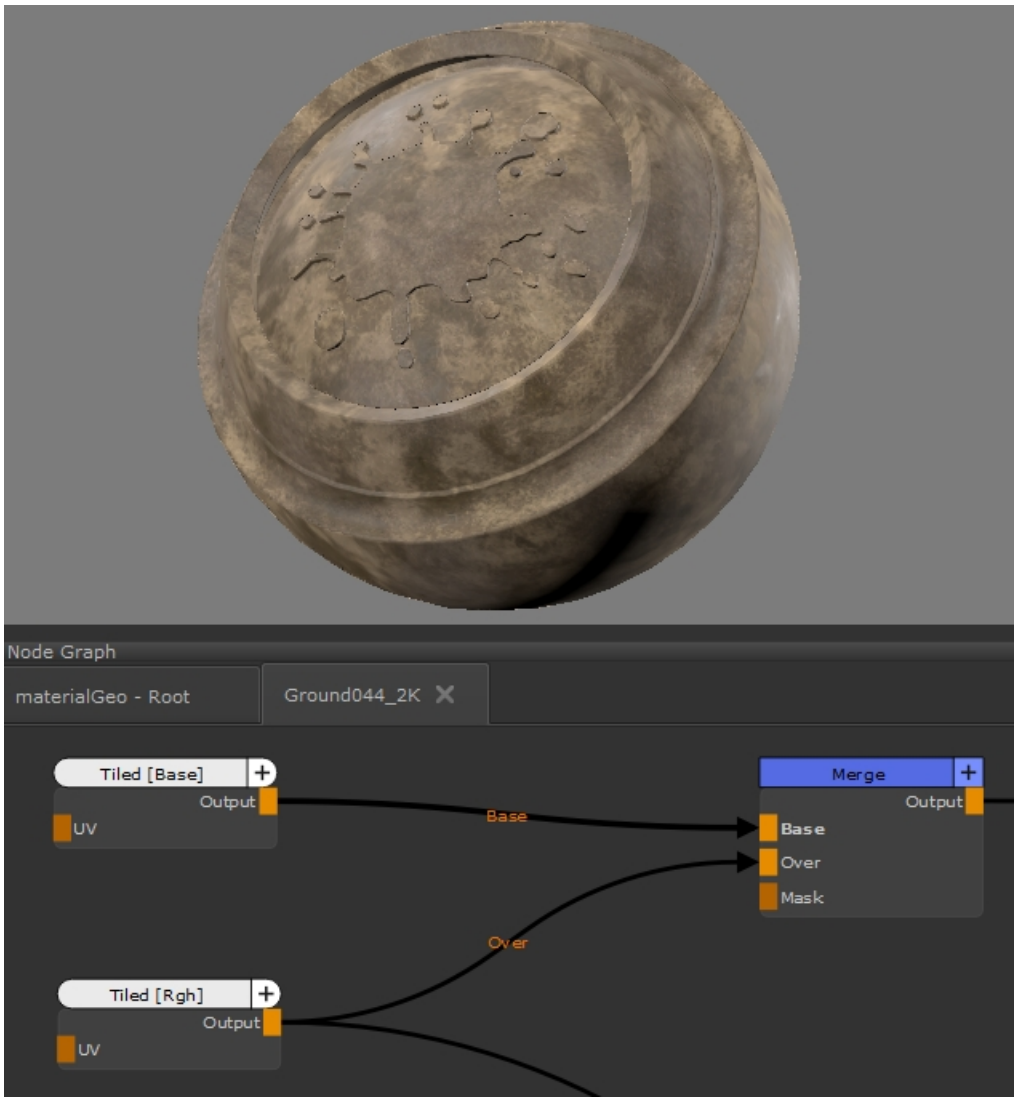
Input: The stream you want to convert to color data.

Scalar to Color Node Properties

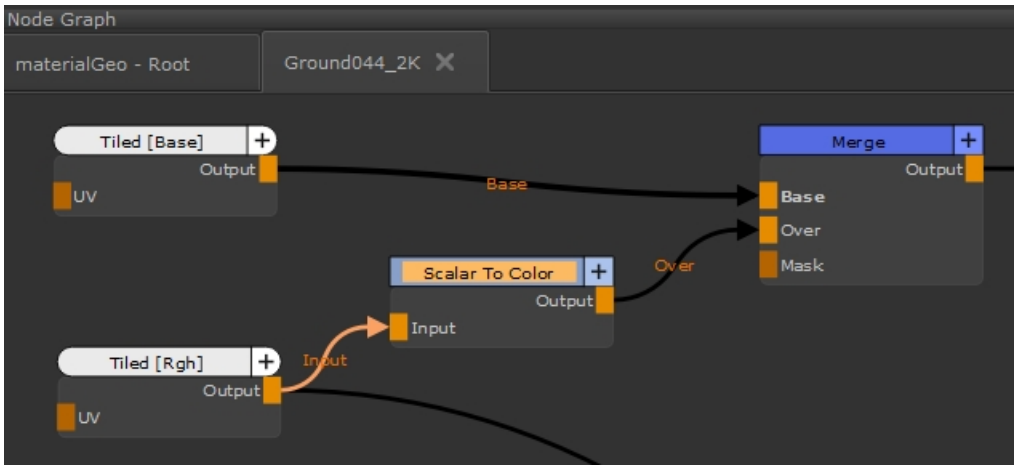
The Scalar to Color node does not include any node properties.

Scalar to Color Node Workflow Example

This example shows the internal node tree in a Material node. In this case, we're blending a roughness map, **Tiled [Rgh]** with the base color, **Tiled [Base]**, using an overlay blending mode in the Merge node.



Roughness is scalar data, so it must be converted to color using a Scalar to Color node before blending to avoid unwanted brightness. In the example, the Scalar to Color node is disabled and enabled to show the difference in output.

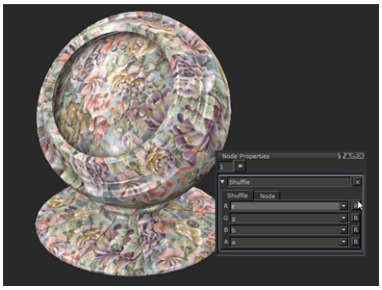


Scalar to Color disabled.

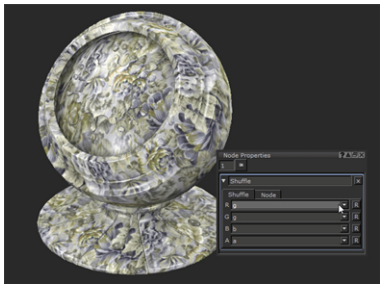
Scalar to Color enabled.

Shuffle Node

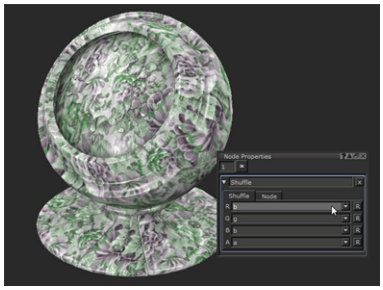
Access: **Nodes** > **Filter** > **Shuffle**



Shuffling the red channel of a tiled fabric texture to **R**.



Shuffling the red channel of a tiled fabric texture to **G**.



Shuffling the red channel of a tiled fabric texture to **B**.

The Shuffle node is a node that lets you rearrange different color channels of the texture into other color channels.

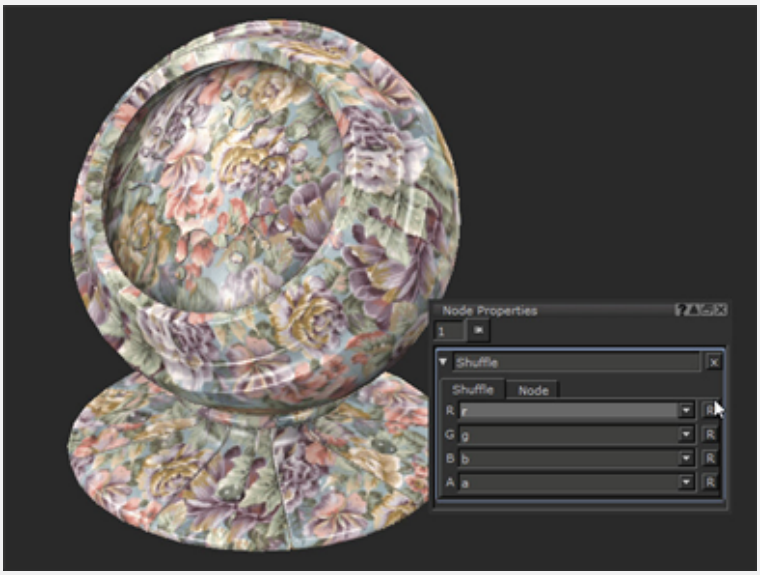
Shuffling the red, green, blue and alpha of a texture is especially useful for shuffling out channel packed images. A channel packed image is one that has multiple grayscale images placed in the different color channels of an image. Since a mask is just grayscale data, you can put those values into the R, G, B, and alpha channels, optimizing your textures since you can get three masks into a single exported image. In your renderer you just need to unpack it to use the correct channel as the mask rather than all three combined.

You can also get a black and white image of a single channel to use for masking, which would give you a different result to standard desaturation.

Shuffle Node Inputs

Input: The texture to be adjusted by the Shuffle node.

Shuffle Node Properties

R *Checkbox*	Rearranges the red channel input, allowing you to shuffle in another channel's values to the red values instead.  If you set R, G, and B to the same output you get a grayscale output of that channel. This is especially useful for channel packed images to get out the individual masks. By default the Shuffle node outputs each channel as itself causing no change.
G *Checkbox*	Rearranges the green channel input, allowing you to shuffle in another channel's values to the green values instead. If you set R, G, and B to the same output you get a grayscale output of that channel. This is especially useful for channel packed images to get out the individual masks. By default the Shuffle node outputs each channel as itself causing no change.
B *Checkbox*	Rearranges the blue channel input, allowing you to shuffle in another channel's values to the blue values instead.

	If you set R, G, and B to the same output you get a grayscale output of that channel. This is especially useful for channel packed images to get out the individual masks. By default the Shuffle node outputs each channel as itself causing no change.
A *Checkbox*	Rearranges the alpha channel input, allowing you to shuffle in another channel's values to the alpha values instead. If you set R, G, and B to the same output you get a grayscale output of that channel. This is especially useful for channel packed images to get out the individual masks. By default the Shuffle node outputs each channel as itself causing no change.

Shuffle Node Workflow Example

Shuffling out Channel Packed Masks

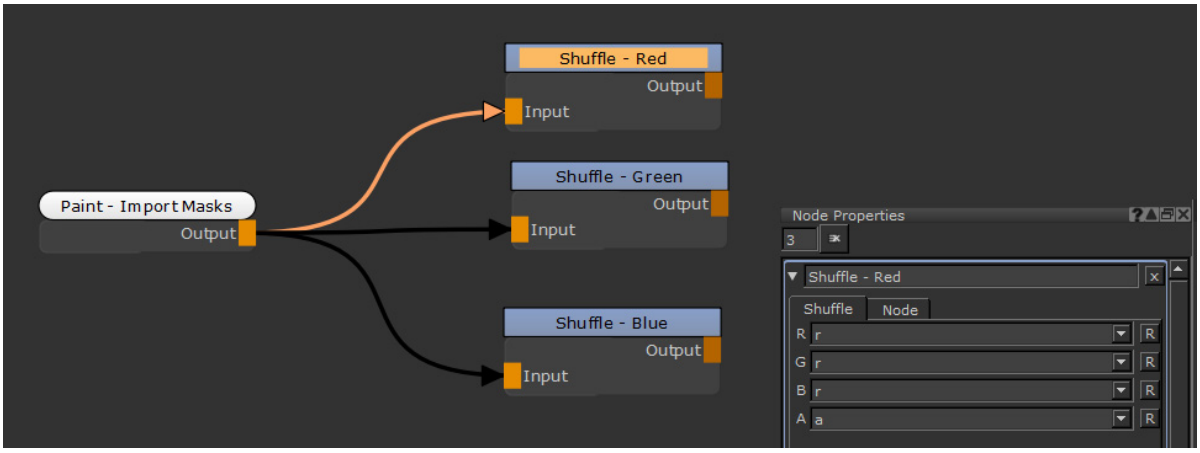
In the example we have a channel packed texture that has been imported into a Paint node. To learn more about this, see the Color Switch Node.

If you are working with other users on a project it could well be the case that you get handed their masks, and if they are working in an optimized way, you could well get masks packed together.

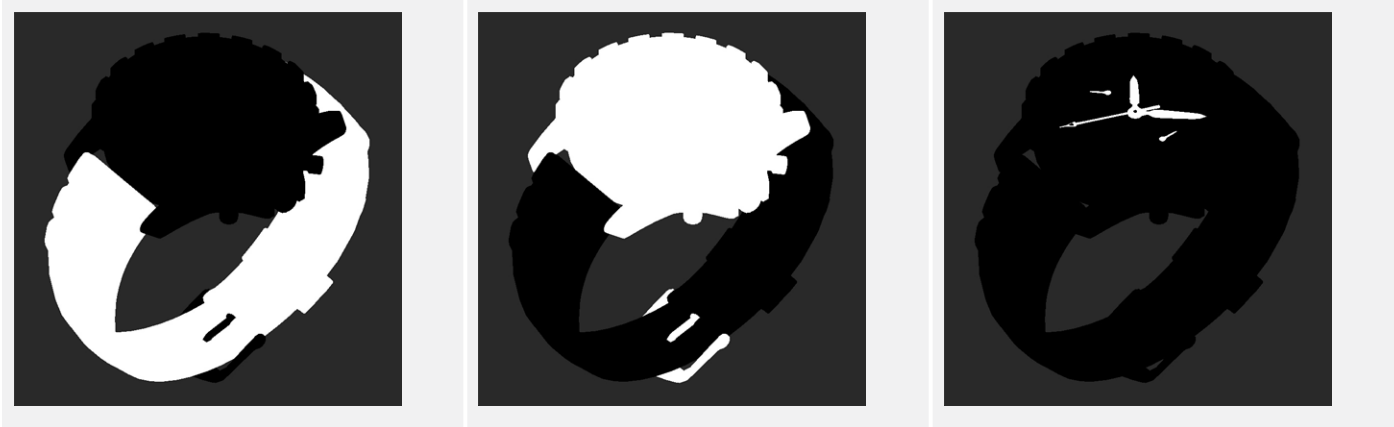
Three masks in a single texture using the three different color channels, also known as a channel packed image.

This watch asset has three masks, one for the leather strap, another for all the metal and a third for the watch hands. They have been imported into a Paint node but we need to split them out into black and white masks to be able to use them individually in Mari.

To unpack these three masks you can place down a Shuffle node into your **Node Graph**. First, to get out the mask in the red channel, a Shuffle node with the R, G, and B all outputting as **r** gives you the correct texture.



Tip: You can set down a different Shuffle node for the green and blue masks, but doing the same but with **g** and **b** respectively.



The three masks shuffled out with the Shuffle node.

Channel packing doesn't just have to be masks. In games design, where optimization is key, you may have your Specular Roughness, Metallic, and Ambient Occlusion all in a single image as they are all grayscale data maps.

To see how to create the channel packed masks, check out the Color Switch Node.

Switch Node

Access: **Nodes** > **Misc** > **Switch**

The **Switch** node allows you to create multiple variations at any point in your node graph, and quickly change which input and upward stream is used with a simple dropdown menu. See Variations with the Switch node to learn more about this.

Switch Node Inputs

There is one input by default: Input. You can add as many inputs as you require and rename them to custom names. These inputs can then be switched between in the **Active Input** field in the **Node Properties**.

Switch Node Outputs

There is one output by default: Output 1.

Switch Node Properties

Active Input	*dropdown*	Adjusts which input is actively in use in the Switch node.
Inputs	*text*	List of available inputs on the Switch node.
	button	Adds a new input.
	button	Removes an input.
	button	Moves the order of the inputs in the list.

Ambient Occlusion Node

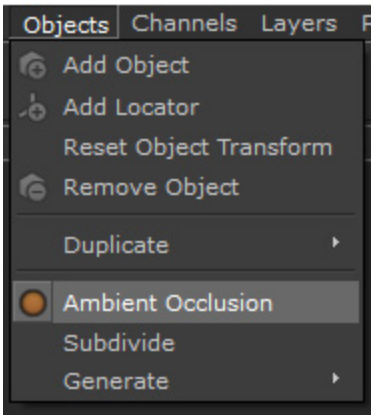
Access: **Nodes** > **Geometry** > **Ambient Occlusion**

The Ambient Occlusion (AO) node uses the object in your scene to create a black and white texture that represents the occlusion on your object. Crevices are filled with a dark shadow using black or gray, and open, smooth surfaces are white. Ambient occlusion is great for making your textures appear more realistic.

Ambient Occlusion applied to a model.

Baking the Ambient Occlusion

1. Open the **Objects** menu and click **Ambient Occlusion**.



The Ambient occlusion option inside the Objects menu.

2. Wait for the Ambient Occlusion to calculate.

   You can monitor the progress in the bottom right corner of the application window.

3. Add an Ambient Occlusion node to your node graph by right-clicking and selecting **Nodes** > **Geometry** > **Ambient Occlusion**.

4. Connect the Ambient Occlusion node **output** to either the Ambient Occlusion channel on your BRDF shader, or directly into the Viewer.

   The calculated Ambient Occlusion appears on your object in the viewer.

Note: If any changes are applied to your object, you need to re-calculate the ambient occlusion.

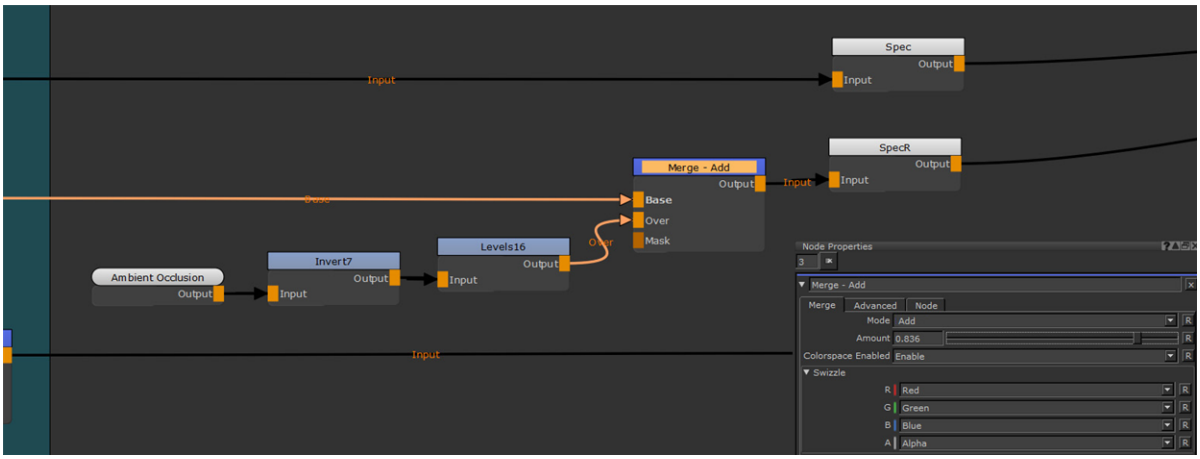
Ambient Occlusion Node Workflow Example

Example - Combining Ambient Occlusion with Maps

Using an Ambient Occlusion node allows you to produce a quick mask for crevices or areas where dirt, grime, or dust would naturally build up. Then, a Levels node helps you to add contrast and darken the black areas of the ambient occlusion. You can use an Ambient Occlusion node as the base for Specular Roughness as those areas are often rougher due to grime build up, or at the end just to add some variation to the map.

The darker areas are where dirt and grime would build up.

After baking the ambient occlusion and placing down an Ambient Occlusion node, you need an Invert node to invert the black and white values of your map so you can make use of a Levels node. Specular roughness uses white as rougher and black as tighter. You can add a Levels node so that you have a bit more control over how much or how little the ambient occlusion affects your textures.



The image shows the node graph setup with Ambient Occlusion, Invert, and Levels plugged into a specular roughness ('SpecR' in the image above) Channel node.

The Invert node flips the ambient occlusion so it works with your specular roughness values. The white values become black, and the black values become white. You can then use a Levels node to make the ambient occlusion darker or lighter, pushing or lessening the effect it has when merged. The blend mode when merging is **Add**.

A Merge node is needed to combine the ambient occlusion with the specular roughness. As you want to use the ambient occlusion to add roughness, then the blend mode of **Add** adds the inverted white from your Ambient Occlusion node.

Experiment with the Merge node's opacity to choose how much you want the ambient occlusion to affect your final specular roughness. Using ambient occlusion can help make your textures look more realistic. The node can be used elsewhere, such as in the base color to dirty your base color up, or in a metallic channel to add metallic wear.

In the image, the crevices of the mesh are lighter in the specular roughness, which creates a dirty and dusty look.

Curvature Node

Access: **Nodes** > **Geometry** > **Curvature**

The Curvature node approximates curvature, concavity and convexity on a surface of a piece of geometry in screen space. It is not intended to be an accurate representation of surface curvature, instead it aims to provide a quick approximation without the artist having to bake a curvature map. As this is the case, it can save time and aid workflow efficiency, as baking a curvature map can be heavy and time consuming.

Curvature generated from a Curvature node.

Curvature generated from a Geo-Channel node.

Curvature Node Inputs

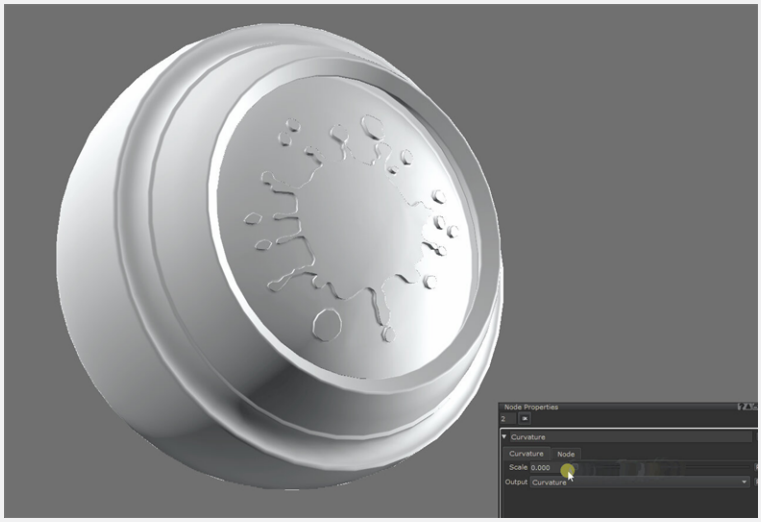
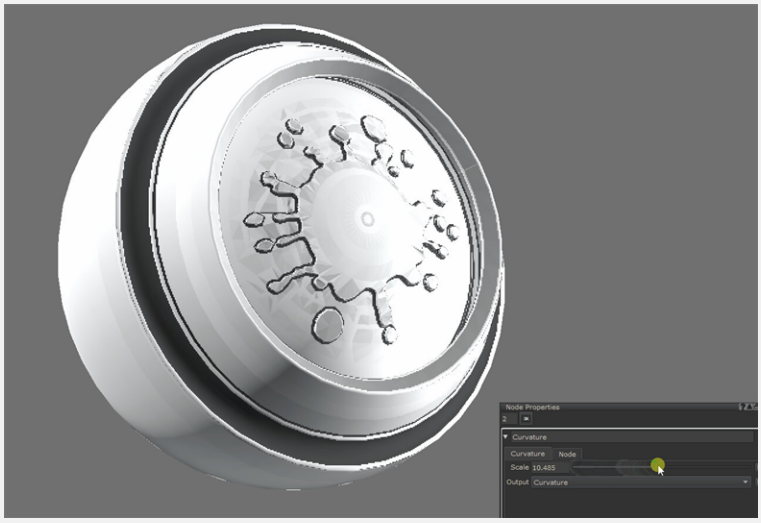
Input	Description
Normal	Allows the user to feed in a normal map to get extra curvature information from it.

Curvature Node Outputs

Output	Description
Output	This outputs whatever is specified by the **Output** attribute in the **Node Properties**.

Curvature	This outputs the surface curvature.
Concavity	This outputs the surface concavity.
Convexity	This outputs the surface convexity.
Both	This outputs both the surface concavity and convexity.

Curvature Node Properties

Scale *text field, slider*	This defines how pronounced the curvature effect is.  Curvature node with a **Scale** of 0. 

	Curvature node with a **Scale** of 10.
Output *dropdown list*	Select what to output. **Note:** These will only change what is being output if you are using the **Output** port. The output ports override this **Output** in the **Node Properties**, so if you are using the **Curvature**, **Concavity**, **Convexity** or **Both** outputs, you will not be able to switch what is output using this dropdown. **Curvature** - Shows the surface curvature with concave areas tending towards black and convex areas tending towards white. Flat areas tend towards gray. **Concavity** - Concave areas tend towards white. Flat and convex areas tend towards black. **Convexity** - Convex areas tend towards white. Flat and concave areas tend towards black. **Both** - Convex and concave areas tend towards white. Flat areas tend towards black.

Geo-Channel Node

Access: **Nodes** > **Geometry** > **Geo-Channel**

The Geo-Channel system allows you to use the same material on a variety of models to achieve the look of edge wear and tear and other photo-realistic effects on your meshes. You might not know what geometries are going to be used when authoring a material. Geo-Channels solve this by driving geometry-agnostic data, such as curvature and ambient occlusion maps into your material setup. The Geo-Channel node points the material at specifically set up Geo-Channels on your object.

Material without Geo-Channel node.

Material with Geo-Channel node driving a curvature map to show edge wear in the material.

You can use the Geo-Channel node anywhere in the Node Graph by making sure Geo-Channels have been setup for your object under the **Geo-Channel Properties** in the **Objects** palette. See Adding Geo-Channels to Objects for more information.

Tip: You can also export baked data from Bake Point nodes directly into Geo-Channels, to act as an intermediate caching point for complex Node Graph setups that can then be added to your Node Graph through a Geo-Channel node. See Syncing Bake Points to Geo-Channel Nodes for more information.

Geo-Channel Node Properties

Channels *Dropdown*	Select the Geo-Channel alias from the dropdown box to be used by the Geo-Channel node.
Background Color *swatch*	Fills any UDIMs on the geometry not included in the Geo-Channel with the selected color. For example, if your geometry has UDIMs 1001-1003 but your Geo-Channel only has 1001-1002, 1003 is filled with the selected background color.

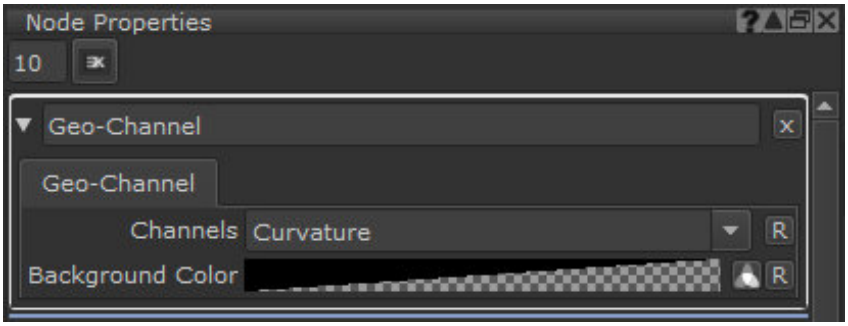
Assigning a Geo-Channel to a Geo-Channel Node

Note: Make sure you have Geo-Channels setup on your object. See Adding Geo-Channels to Objects for more information.

1. In the Node Graph, press **Tab**.

   A search field appears.
2. Type the following into the search field:

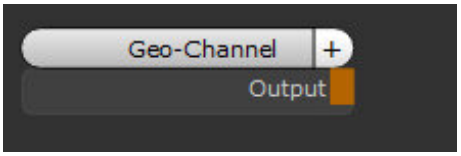
   `Geo-Channel`
3. Click the resulting **Geo-Channel** node.
4. Double-click the Geo-Channel node in the Node Graph to bring up the **Node Properties**.



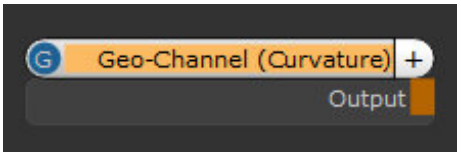
The Geo-Channel **Node Properties**.

5. Click the **Channels** dropdown and select the Geo-Channel you want to use with this Geo-Channel node.

   The Geo-Channel node updates to display a blue icon and the channel name, representing the Geo-Channel you selected.



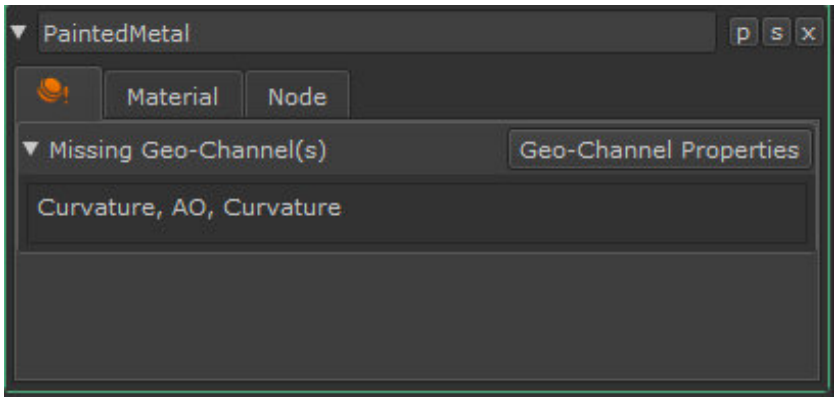
Without a Geo-Channel selected.



With a Geo-Channel (Curvature) selected.

Importing Materials with Geo-Channels into New Projects

When you import a material containing a Geo-Channel into a new project, if the Geo-Channel is missing from the object(s) in your project, double-clicking the material in the node graph displays a warning in the node properties. The missing Geo-Channels are displayed in the **Missing Geo-Channel(s)** panel.



Tip: Click the Geo-Channel layer in the **Layers** Palette to open the Geo-Channel properties when working with layers instead of the Node Graph.

Tip: See Working with Materials in the Node Graph for more information on exporting and importing materials.

To add the missing Geo-Channel:

1. Click **Geo-Channel Properties**.

   This opens the **Geo-Channel Properties** tab.
2. See Adding Geo-Channels to Objects for information on how to add a Geo-Channel to the object.

Note: Make sure the geo-channel name matches the geo-channel selected in the Geo-Channel node.

Geo-Channel Node Workflow Example

Creating Edge Wear and Tear Using Geo-Channels

Geo-channels can be used in a material pipeline to create realistic wear and tear such as rust, chipped paint, or worn leather on your objects. Since geo-channels are geometry-agnostic, you can use the same material across a variety of objects while ensuring the material looks realistic for the specific object. The Geo-Channel node is most useful when used as a mask on the object.

Video: See An Introduction to Geo-Channels in Mari.

Syncing Bake Points to Geo-Channel Nodes

You can export baked data from Bake Point nodes directly into Geo-Channels. The Geo-Channel node can then act as an intermediate caching point for complex Node Graph setups. Using this method, you can share common masks in your project across various other channels, groups, gizmos, and materials, with multiple Geo-Channel nodes receiving the same Bake Point node data.

Note: The Bake Point data you want to store to a Geo-Channel can contain other Geo-Channel nodes, but it does not need to.

To export Bake Point node data to a Geo-Channel:

1. Create a new Geo-Channel on your object.

   See Adding Geo-Channels to Objects.
2. Add a Bake Point node to your Node Graph if you don't already have one.

   See Bake Point Node.
3. Double-click the Bake Point node in the Node Graph to open the **Node Properties**.
4. In the **Bake Point** tab of the **Node Properties**, click **Bake**.

   The Bake Point node caches all the data in the Node Graph that is upstream to the Bake Point node and turns green to indicate it is up to date.
5. Click the **Export** tab on the Bake Point node **Node Properties**.
6. In the **Geo-Channel** dropdown, select the Geo-Channel you want to export the Bake Point data to.
7. Enable **Sync on Bake** to automatically sync the selected **Geo-Channel** with the latest bake data.

For more information on the Bake Point **Node Properties**, see Bake Point Node

8. Click **Export Baked to Geo-Channel**.

   The Bake Point node name updates to display the name of the selected Geo-Channel in brackets.

   You can add as many Geo-Channel nodes as you need to your Node Graph using the selected Geo-Channel from step 6. The exported Bake Point data from step 8 is shared using those Geo-Channel nodes.

Position Node

Access: **Nodes** > **Geometry** > **UV**

Mari mainly deals with two coordinate spaces: 3D and UV. The Position node outputs position information in 3D space. Using the 3D node on its own is not useful as it only shows position information represented by color but you can use it in conjunction with other nodes. For instance, you can use the Position node with nodes that require a **Position** input. You can also use it with procedural nodes such as noise nodes or texture type nodes, for example, Tiled nodes. By default, a lot of Mari's procedural nodes generate patterns and noises based on 3D space position. See Position: Node Graph Workflow Example.

Position Node Inputs

No input.

Note: See Position: Node Graph Workflow Example for more information on the output.

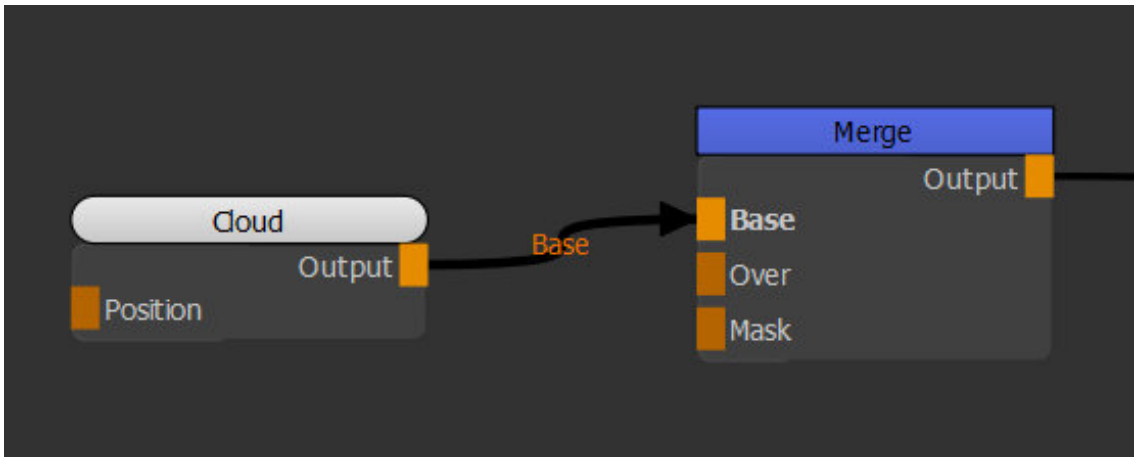
Position Node Properties

No properties.

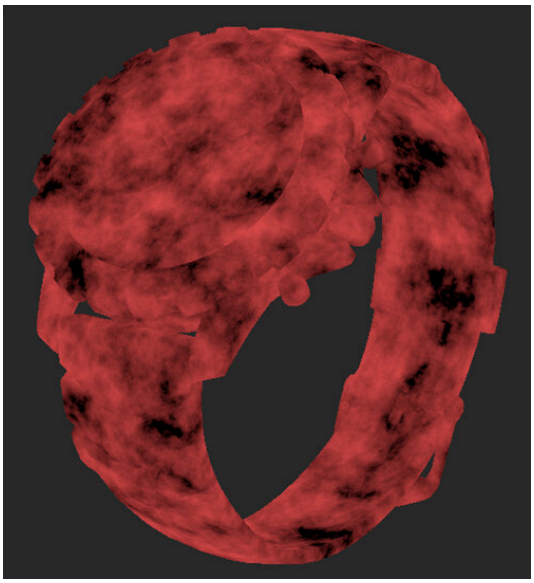
Position: Node Graph Workflow Example

In this example, let's use a Position node to shift the position of the Cloud node noise applied to the object.

1. Create a Cloud node and connect it to the **Base** input of the Merge node.

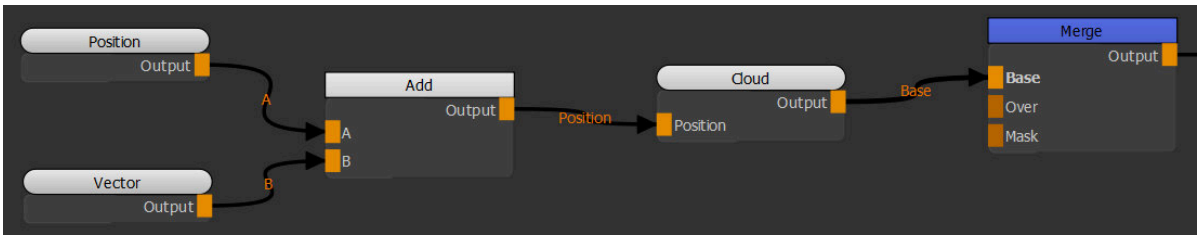


Adding the Cloud node to the scene to texture the object.



Cloud noise applied to the object.

2. Create an Add node and connect it to the **Position** input of the Cloud node. Create a Position node and a Vector node and connect them respectively to the **A** and **B** inputs of the Add node.

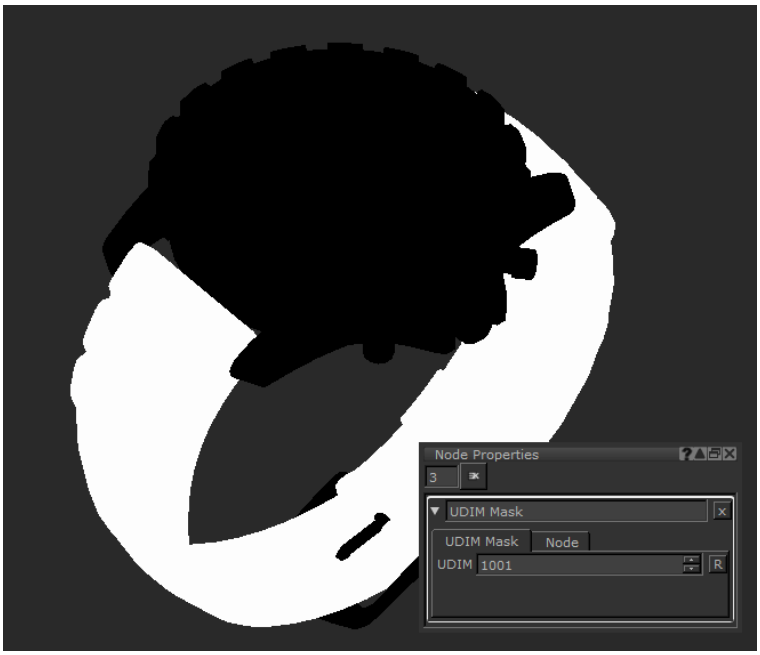


Adding an Add, a Position, and a Vector node to adjust the position of the noise.

Notice how moving the **X** slider of the Vector node moves the noise on the **X** axis as we are changing the position information through the Position node.

UDIM Mask Node

Access: **Nodes** > **Geometry** > **UDIM Mask**



Using the UDIM Mask to mask off areas of this watches mesh.

The UDIM Mask node outputs a black and white mask of a single specific UDIM patch. It is the quickest way to create a mask for an entire patch and is more efficient than using a filled paint node.

If you have well organized UVs, then a UDIM mask can be an efficient way to mask areas. For example, put all your metallic objects onto a single UDIM, or all objects that need to be textured as wood, you can use the UDIM Mask to quickly select them all and start assigning textures to everything at once.

Note: You can only select one UDIM and cannot do a range or multiple UDIM numbers at once. To do this you would need to use a Paint node with filled colors.

UDIM Mask Node Properties

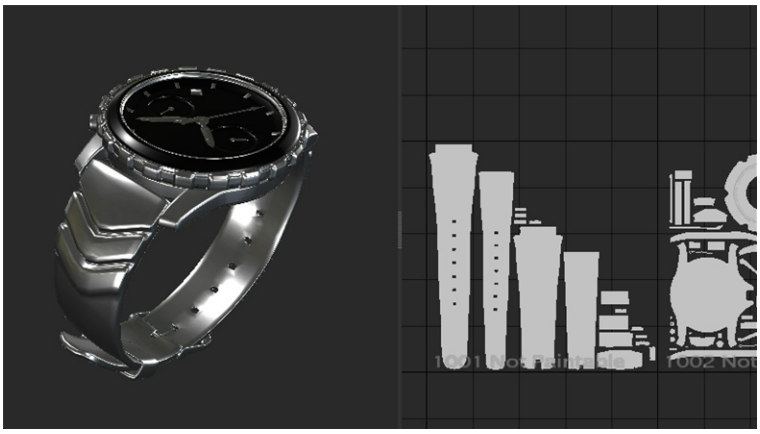
UDIM	Using **UDIM** you can select which UDIM to output a mask of. The **UDIM** number specified

increments	here is filled with white while the other UDIMs are black. By clicking the incremental arrows up and down you can scroll through the different UDIMs starting at 1001. You can also type yourself a UDIM number. **Note:** You can only select one **UDIM** and cannot do a range or multiple. Default is 1001.

UDIM Mask Node Workflow Example

Quick Masking an Entire Patch

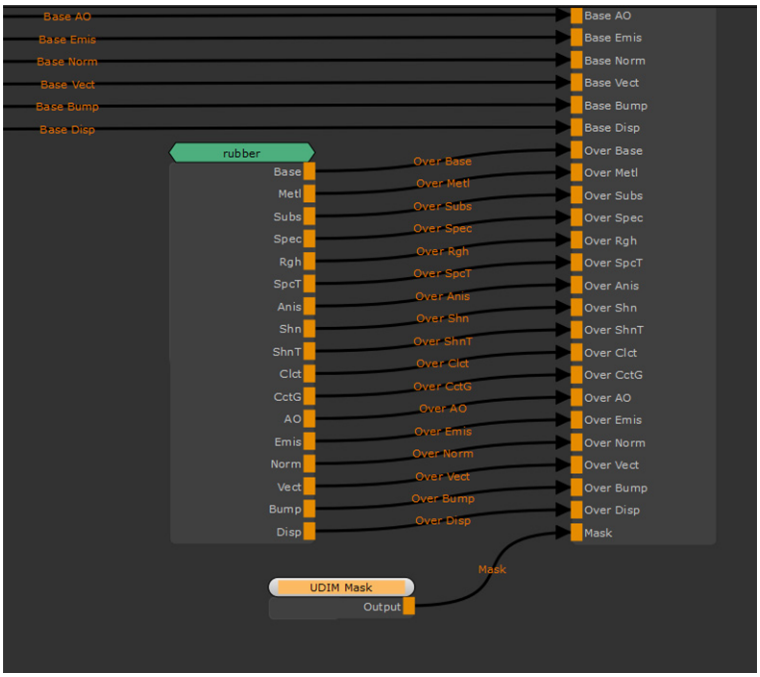
In this watch object the strap is one the same UDIM with nothing else on the patch. At the moment there is just this metal texture but that needs to change.



A watch asset with the whole rubber strap in UDIM 1001.

While you could set up a Paint node to create a mask for it, it would take longer and also be less efficient as you are storing paint data. This is the perfect time to use a UDIM Mask.

The strap is on the first patch, so by placing a UDIM Mask node down with the default UDIM setting of **1001**, we have a mask for the strap in a few clicks. Now we can use that mask to add a different texture to it.



The UDIM Mask is being used to mask this **rubber** Material node in this Multi-Channel Merge node.

By placing the mask into the mask slot of the Multi-Channel Merge, the rubber is now only on the strap rather than merging over everywhere.

Without UDIM Mask.

With UDIM Mask.

A quick and super easy way to create a mask for this asset!

UV Node

Access: **Nodes** > **Geometry** > **UV**

Mari mainly deals with two coordinate spaces: 3D and UV. The UV node outputs position information in UV space. Using the UV node on its own is not useful as it only shows position information represented by color but you can use it in conjunction with other nodes. For instance, you can use the UV node with nodes that require a **Position** input. You can also use it with procedural nodes such as noise nodes or texture type nodes, for example, Tiled nodes. By default, a lot of Mari's procedural nodes generate patterns and noises based on 3D space position, but depending on the UV layout you may want to use the UV node with some objects to make them look more realistic. See UV: Node Graph Workflow Example.

UV Node Inputs

No input.

Note: See UV: Node Graph Workflow Example for more information on the Output.

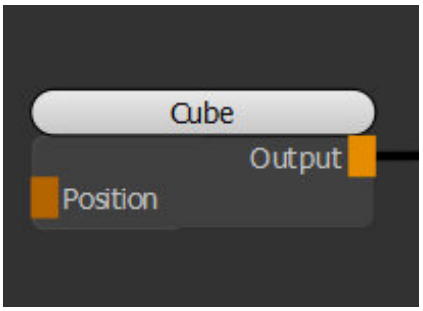
UV Node Properties

No properties.

UV: Node Graph Workflow Example

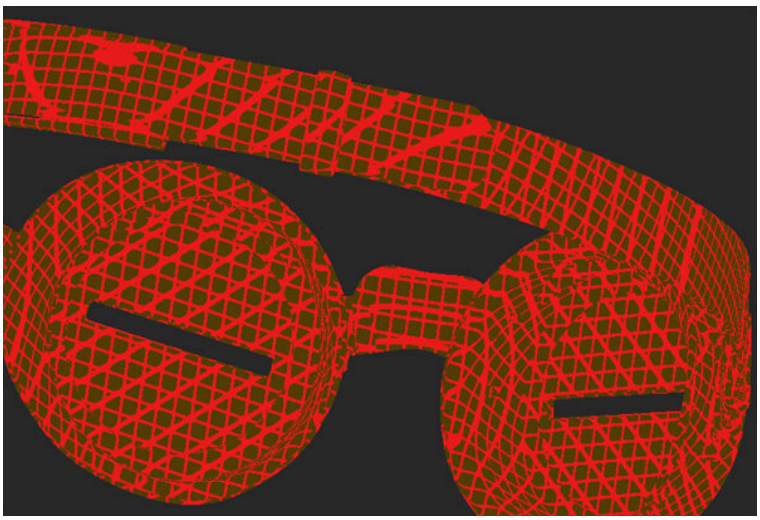
In the example below, let’s use a UV node to better apply a cube pattern to the straps of the pair of goggles.

1. Create a Cube node.
   The Cube node gives a grid pattern look to the pair of goggles.



Adding a Cube node to the scene to texture the pair of goggles.

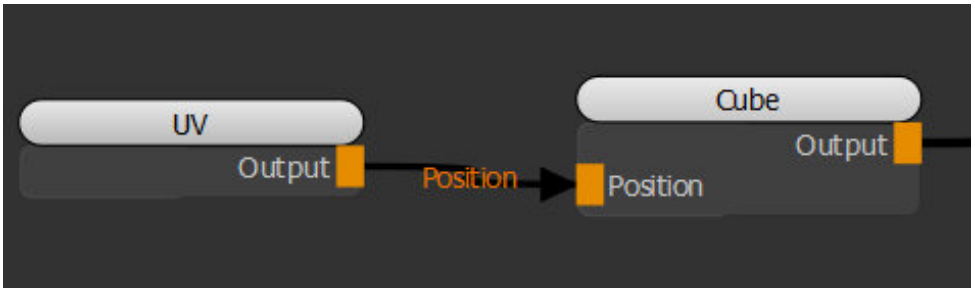
Notice how the grid pattern is applied to the straps.



Cube node applied to the pair of goggles to render a grid pattern look.

2. Create a UV node and connect it to the **Position** input of the Cube node.

   This displays the pattern using UV space instead of 3D space.



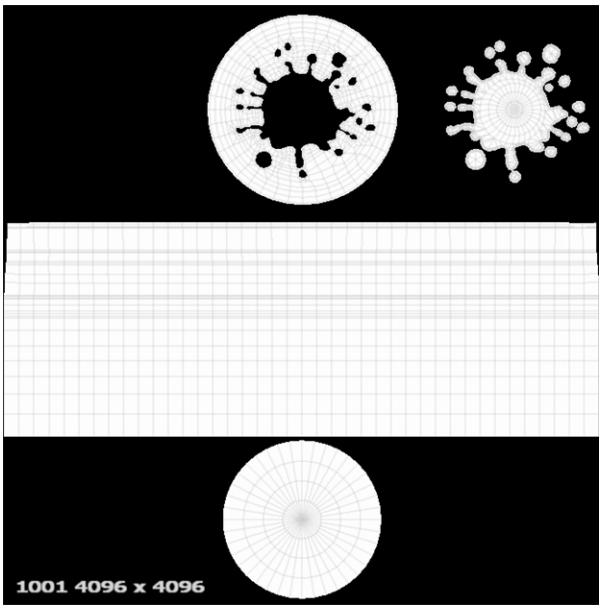
Adding a UV node to the scene to switch to UV space.

Notice how the grid pattern is applied differently and is better aligned to the strap pieces.

The pattern is applied using UV coordinate space.

UV Mask Node

Access: **Nodes** > **Geometry** > **UV Mask**



A black and white mask created based on the UVs of a shader ball asset.

The UV Mask creates a black and white mask as an output based on your UVs. The inside of your UVs fills with white and anything outside fills with black.

At first glance this node may not seem that useful, but the node does not include bleed at the edges so it's very useful when combined with other pieces of software like Nuke to create edge masks and other useful images for texturing.

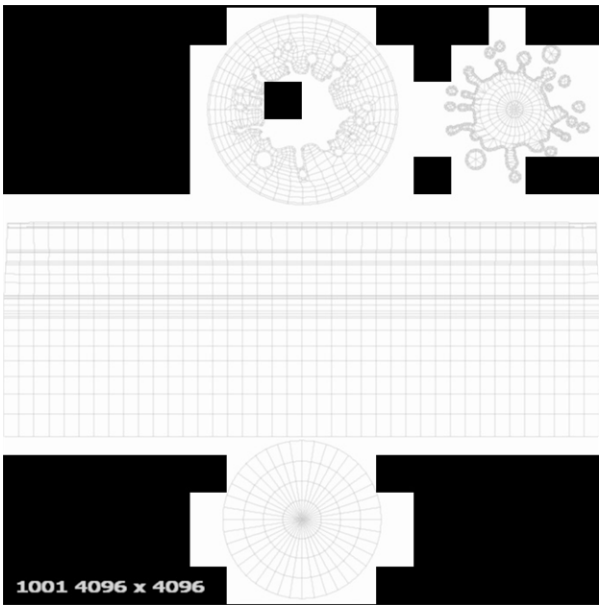
Unlike most nodes, the UV Mask has no input or properties. Its output is based on the UVs of your object and nothing else.

UV Mask Node Workflow Example

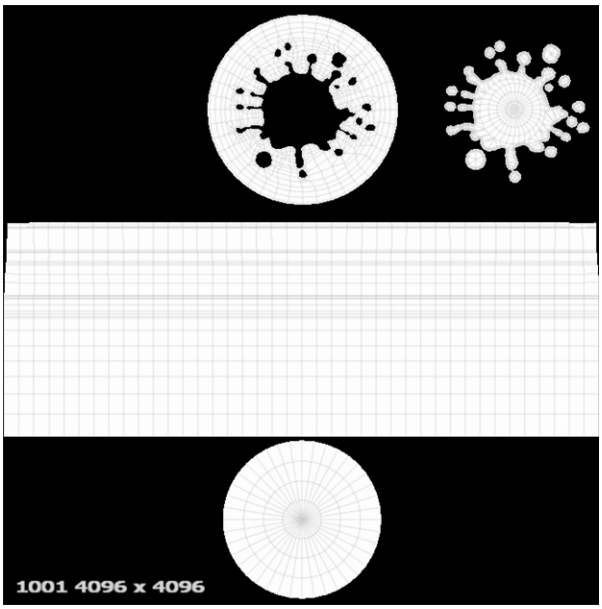
UV Mask for Edge Detection Texturing with Nuke

Let's take a look at a great use for the UV Mask, edge detection of UV shells. This is a fairly advanced technique but a useful one.

In the example, you can see the UV view of the same object, textured in two ways. The first was with the paint tool in a Paint node. As you can see when the paint buffer is baked down, the white is bled over the edges.



Paint node - hand painted mask of the whole object.



UV Mask - output.

The second image is the UV Mask. In the 3D view it looks exactly the same, but in the UVs when exported there is no bleed.

Here is an object that has had all of its maps textured. The wings are looking a bit plain though. Since the object is going to be an aged golden statue, it would be nice to get a bit of detail around the edges of the wing feathers to make it look a bit more grand.

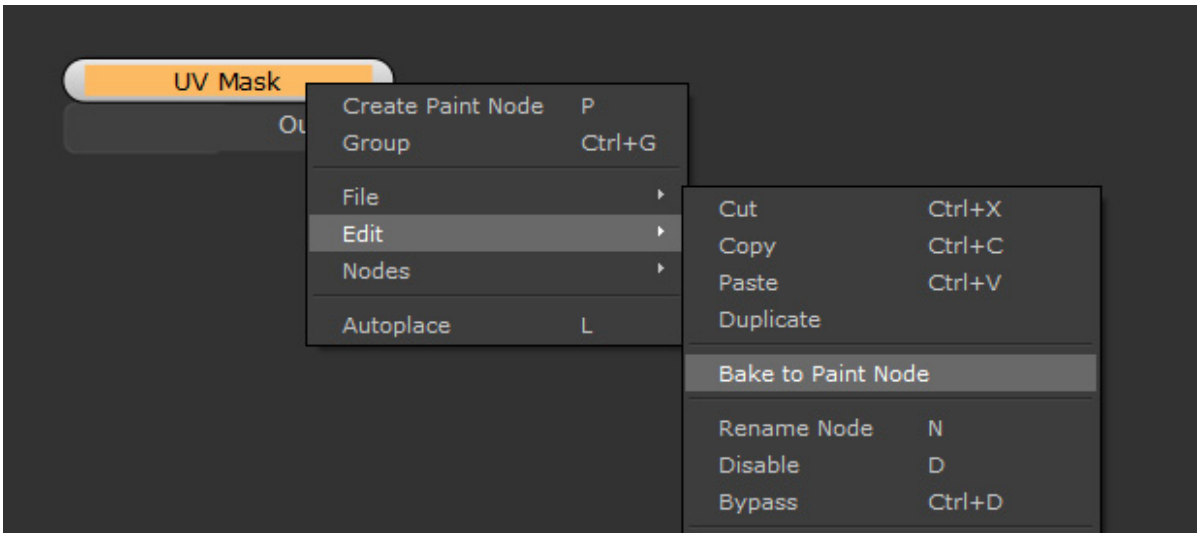
The **Base Color** of this mechanical bird asset.

Hand painting the edge of each piece of the wing would be a painful task and even with steady stroke enabled it would be really difficult to get a clean edge. If we could create a mask of the geometry and then erode into that mask, we could then flip it and have just the edge of all the feathers.

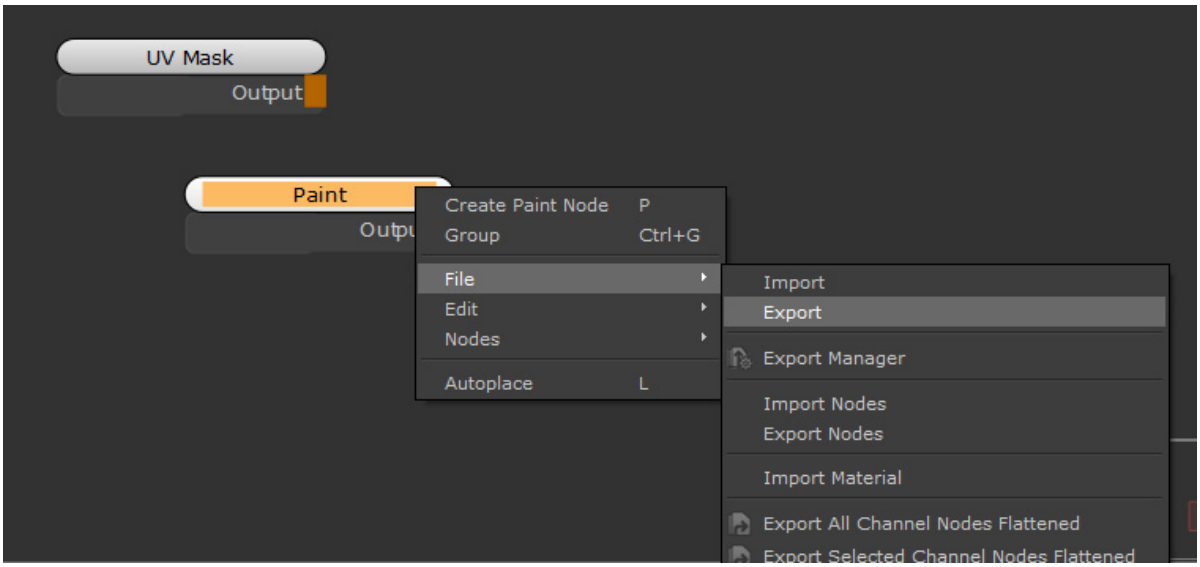
But if we have a mask with bleed then it's going to be useless, we need a mask that fills all the UVs but with no bleeding so we can shrink the mask a bit and get an edge. That's where the UV mask node comes in.

> **Tip:** The UV Mask node is really powerful when exported and used with other pieces of image editing software.

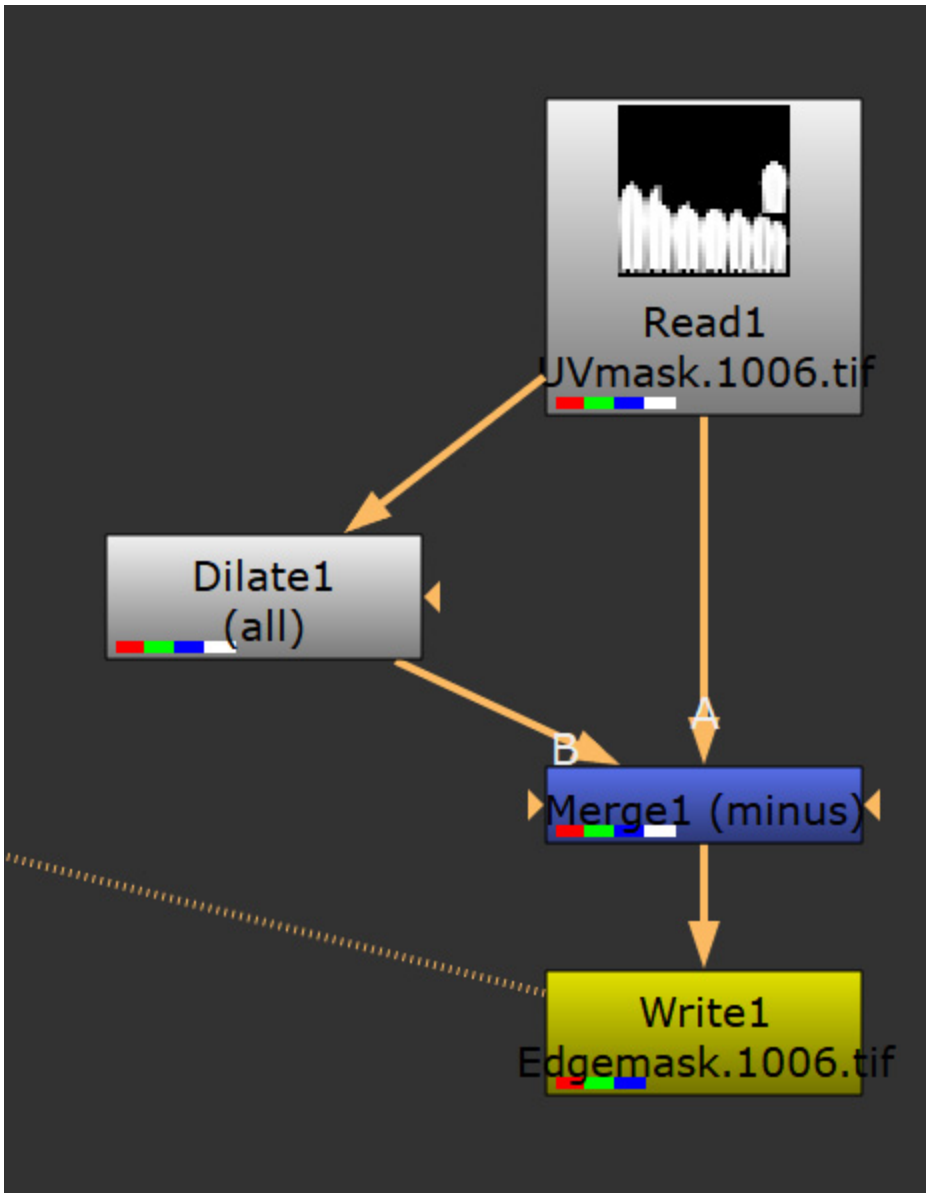
First we need to export the **UV Mask** node's data. That requires us to bake it a Paint node first as you cannot export non baked nodes. To do that you just need to right-click the node and select **Edit** > **Bake to Paint Node**.



Once that is baked, it is ready to be exported by right-clicking and selecting **File** > **Export**.



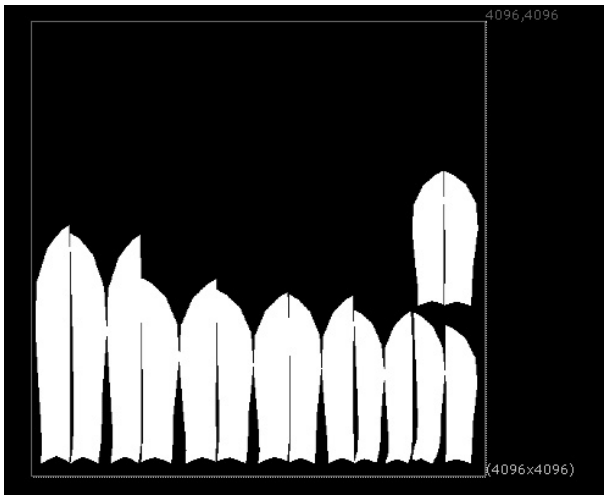
We can now take those images into Nuke or another image software suite where you can manipulate the edges of selections.



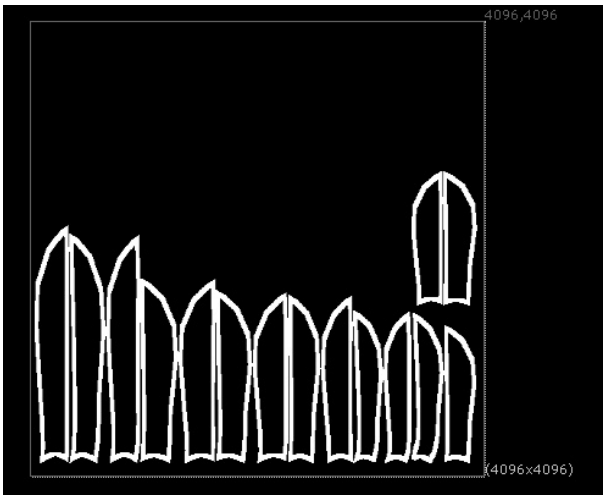
The Nuke Setup.

Here is a very simple Nuke script to get an edge mask. If you haven't used Nuke before then it's nodal work flow is similar to Mari's, but this is all achievable in other image editing software.

First the UV Mask is read in, then it is eroded inwards using the Dilate node in Nuke, then we minus the original from this new dilated version which cuts out the interior of the UV islands as you can see below. Then we **Write** it out from Nuke.

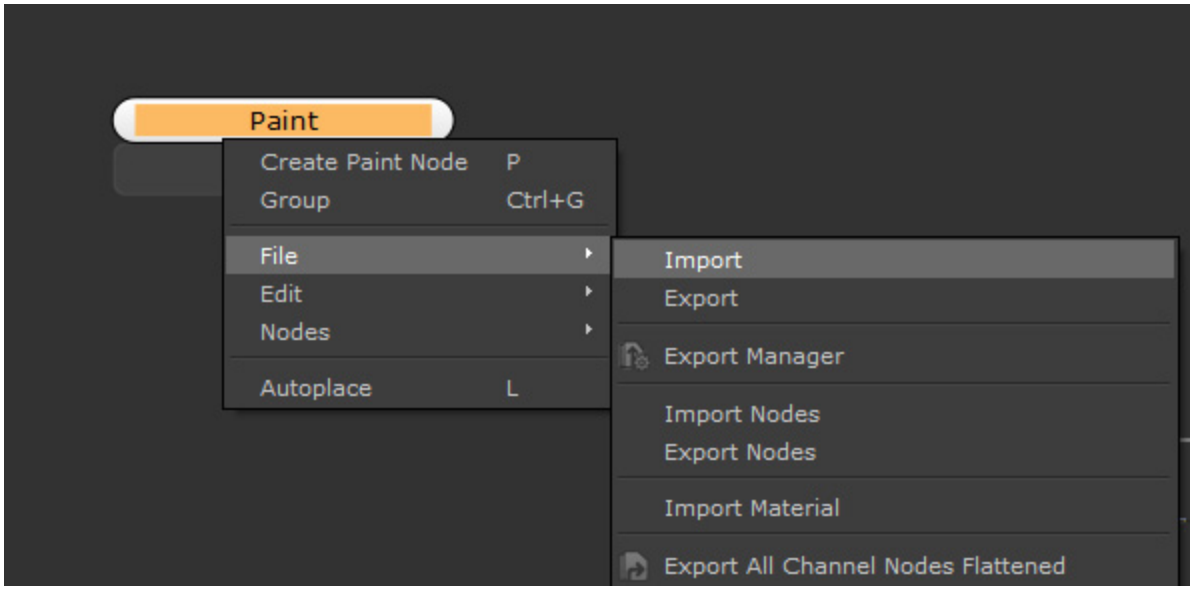


Before



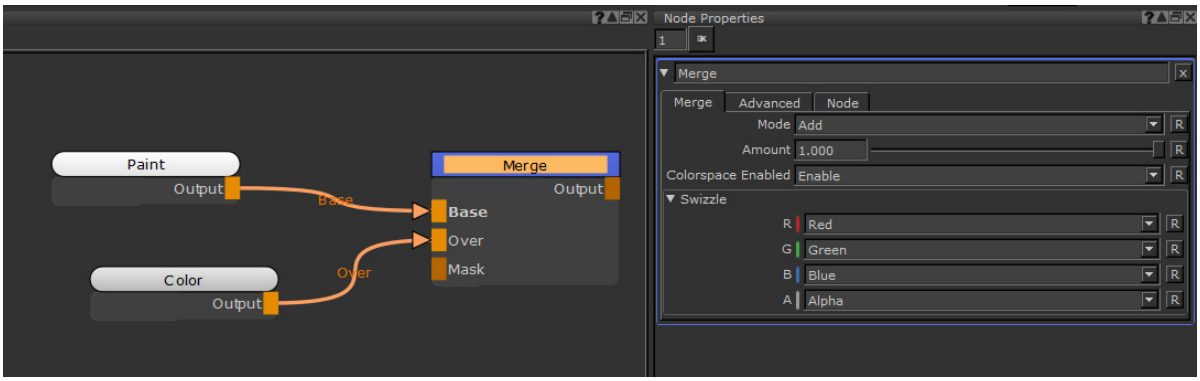
After

Now we have this mask we can import as a new Paint node in Mari.

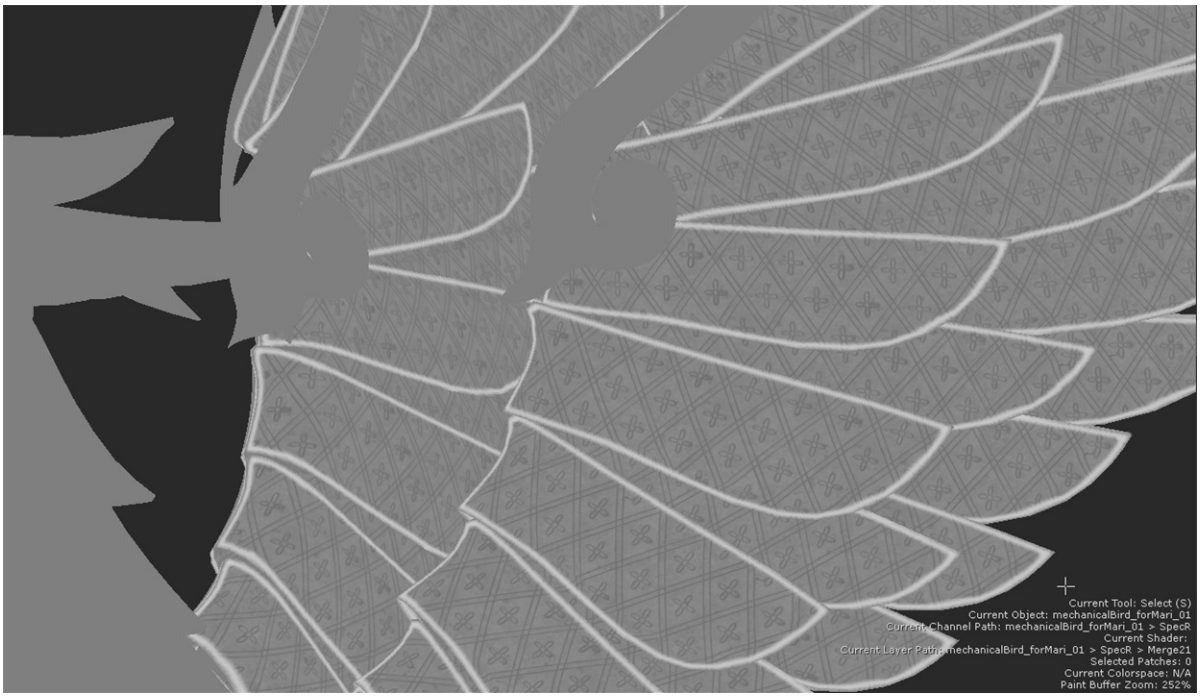


How to import images back into a Paint node.

Using a Merge node with a blend **Mode** of **Add**, we can merge it on top of a Color node of 0.5 gray to make our displacement push out at the edges.



For an extra flourish a tiled texture has also been merged in to add some more detailing to make the design seem more grand.



Here is the final render with the displacement added to the shader to help add detail.

While there would have been ways to achieve a similar result without the UV mask, the speed and lack of bleed on the node is what sets it apart and can speed up workflows.

View as Color Node

Access: **Nodes** > **Misc** > **View as Color**

The View as Color node forces the View Transform LUT into the color display context. This is not the same as disabling the Viewer Transform because you may have a LUT specified for the Viewer Transform's color context that does not match the look of the canvas when the Viewer Transform is enabled.

View as Color Inputs

Input: The stream you want to view as color data.

View as Color Node Properties

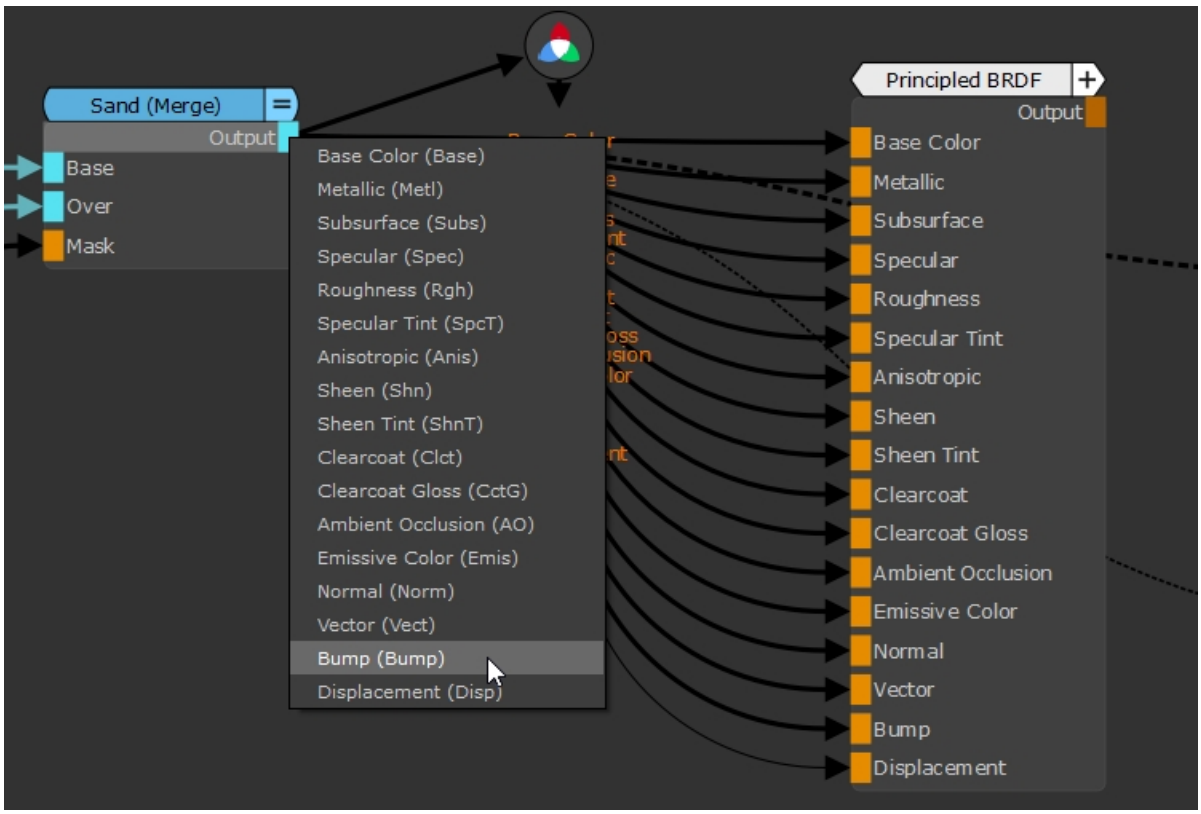
The View as Color node does not include any node properties.

View as Color Node Workflow Example

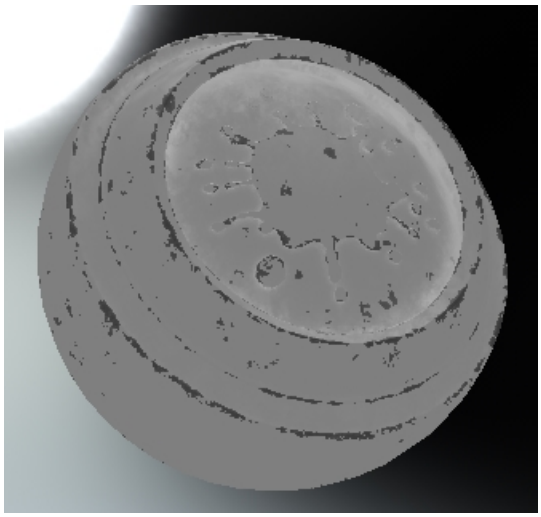
View as Color allows you to view individual streams in your node graph, rather than viewing the entire shader through the Viewer Transform.

1. Add a View as Color node to the node graph.
2. Drag the input from the View as Color node to the output of the node you want to view.

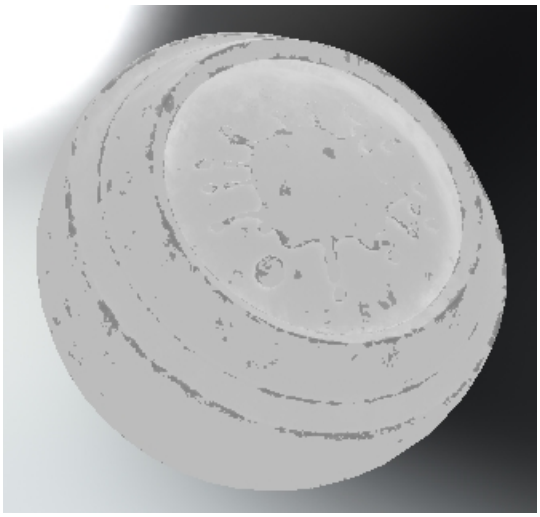
   The available streams are displayed.



3. Select the required stream from the list and connect the Viewer to the View as Color node.



Bump viewed as scalar data.



Bump viewed as color data.

View as Scalar Node

Access: **Nodes** > **Misc** > **View as Scalar**

The View as Scalar node forces the View Transform LUT into the scalar display context. This is not the same as disabling the Viewer Transform because you may have a LUT specified for the Viewer Transform's scalar context that does not match the look of the canvas when the Viewer Transform is disabled.

View as Scalar Inputs

Input: The stream you want to view as scalar data.

View as Scalar Node Properties

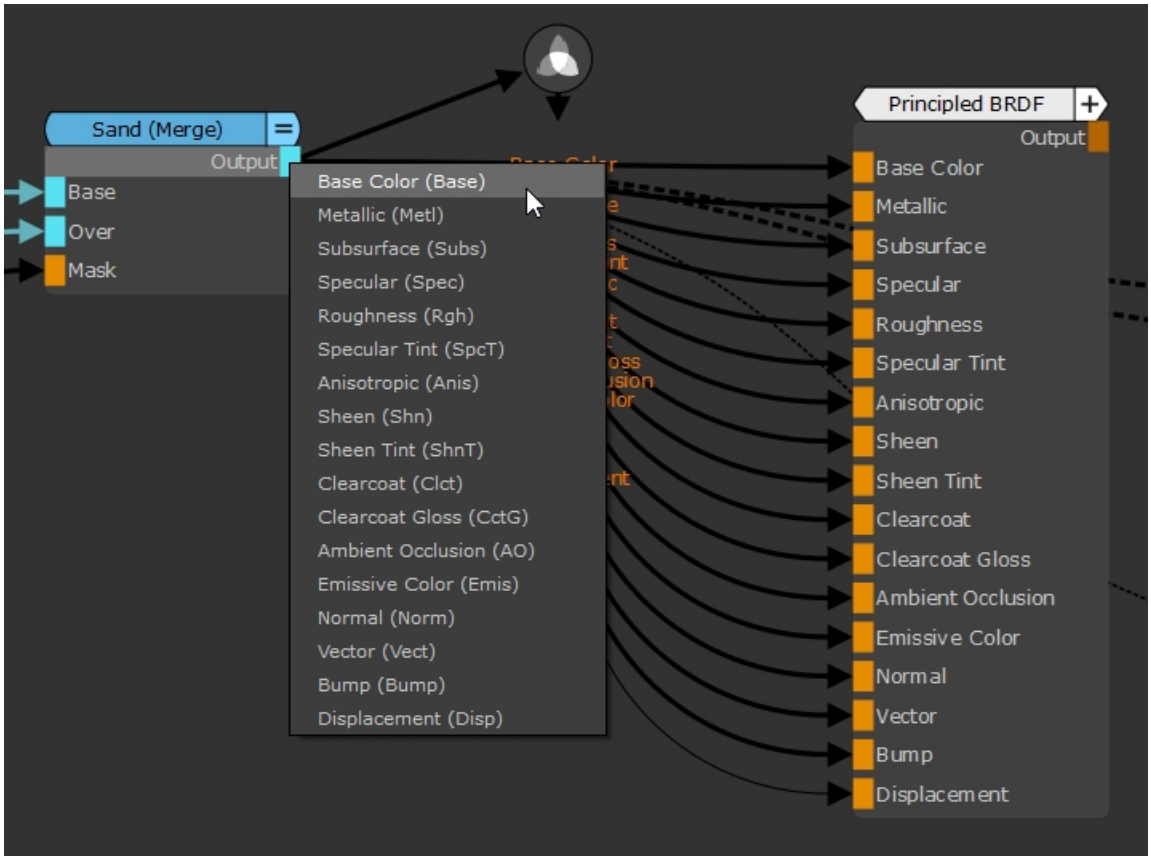
The View as Scalar node does not include any node properties.

View as Scalar Node Workflow Example

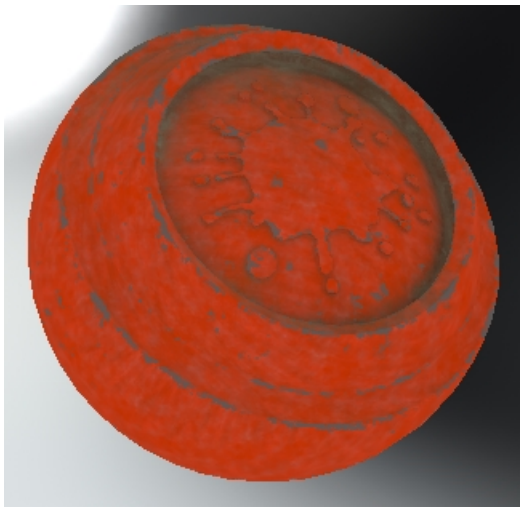
View as Scalar allows you to view individual streams in your node graph, rather than viewing the entire shader through the Viewer Transform.

1. Add a View as Scalar node to the node graph.
2. Drag the input from the View as Scalar node to the output of the node you want to view.

   The available streams are displayed.



3. Select the required stream from the list and connect the Viewer to the View as Scalar node.



Base viewed as color data.

Base viewed as scalar data.

Bottom Transparency Node

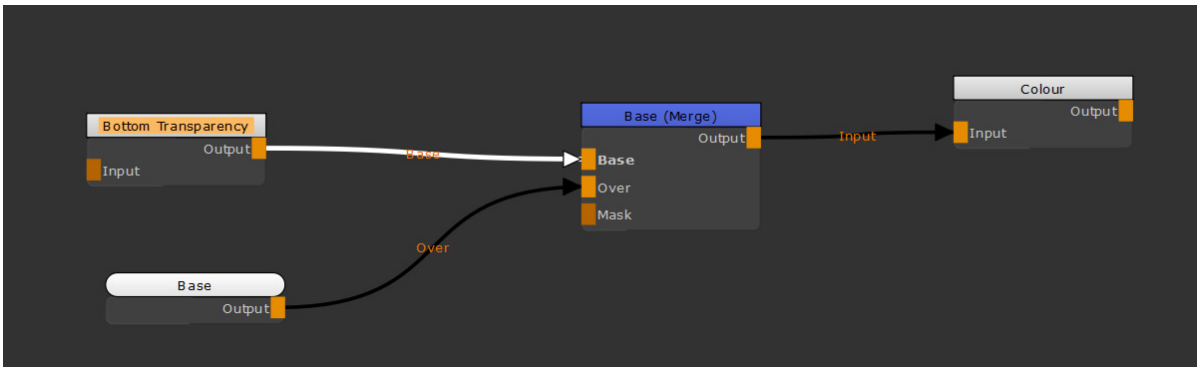
Access: **Nodes** > **Layer** > **Bottom Transparency**

The Bottom Transparency node is an empty node which has no alpha and functions to indicate the bottom of a layer stack. The value of the fill is 0,0,0,0 in the RGBA channels.

The node is automatically created if you add **Channels** in the new project creation UI or if you create a new channel inside the object.

When creating a Group Layer, the **Over** input stream is regarded as the "contents" of the Group. Then, a Bottom Transparency node is created to mark the bottom of this sub layer stack. Now, if you turn off the **Pass Through** mode on the Group Layer, the input of the Bottom Transparency node is connected back to main stream of the main layer stack.

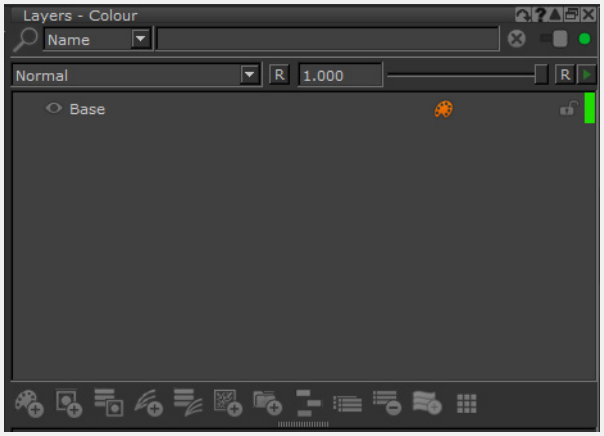
Mari by default creates the following nodes when creating a channel from the **Channels** palette.



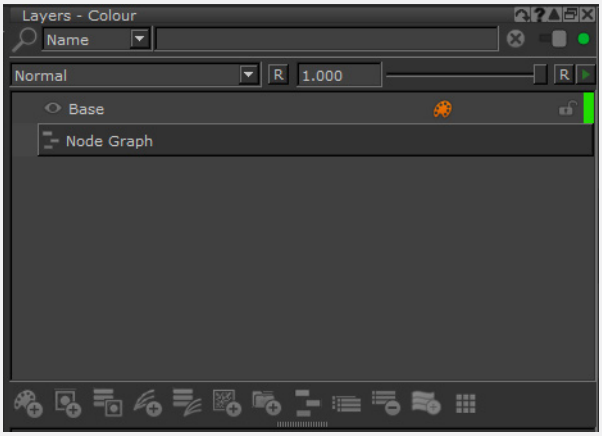
The Base (Merge) node shown here is called a **Layer** in the layer palette, whereas the individual nodes are not. When you add a new layer in the **Layers** palette, a new merge node is made with a paint node plugged into the **Over**, and the previous merge in the node tree is added into the **Base**.

When you create a channel from scratch however, there is no merge to use as a base. This is why Mari uses the empty Bottom Transparency node as it has no data and the Paint node (**Base** in the above screenshot) is the only thing shown.

If you decide to remove the Bottom Transparency node and replace it with something else then the **Layers** palette changes to the following example.



With Bottom Transparency



Without Bottom Transparency

If you decide to switch back to using layers while using the Node Graph and you see the example without Bottom Transparency, you can fix this by adding a Merge node at the start of your node tree with a Bottom Transparency in the **Base**.

Bottom Transparency Node Inputs

Input: The role of the input of a Bottom Transparency node is to implement the **Pass Through** function of a Group Layer.

Merge Node

Access: **Nodes** > **Layer** > **Merge**

The Merge node allows you to combine two nodes using blending operations and includes a **Mask** input to mask out regions from blending.

Under the hood, Mari passes around a set of four numbers along a connection between two nodes. The Merge node takes in a set of four numbers for each of its inputs and outputs a set of four numbers. When merging the **Base** and **Over** inputs, the blending operations are taking each number of the **Base** and **Over** inputs and apply the mathematics behind each blending mode to achieve a blended result.

Merge Node Inputs

- **Base** - Any node containing textures or paint, and any procedural node can be connected to the Merge node's **Base** input.
- **Over** - Any node containing textures or paint, and any procedural node can be connected to the Merge node's **Over** input.
- **Mask** - Any node containing textures or paint, and any procedural node can be connected to the Merge node's **Mask** input.

Note: See Merge: Node Graph Workflow Example for more information on the output.

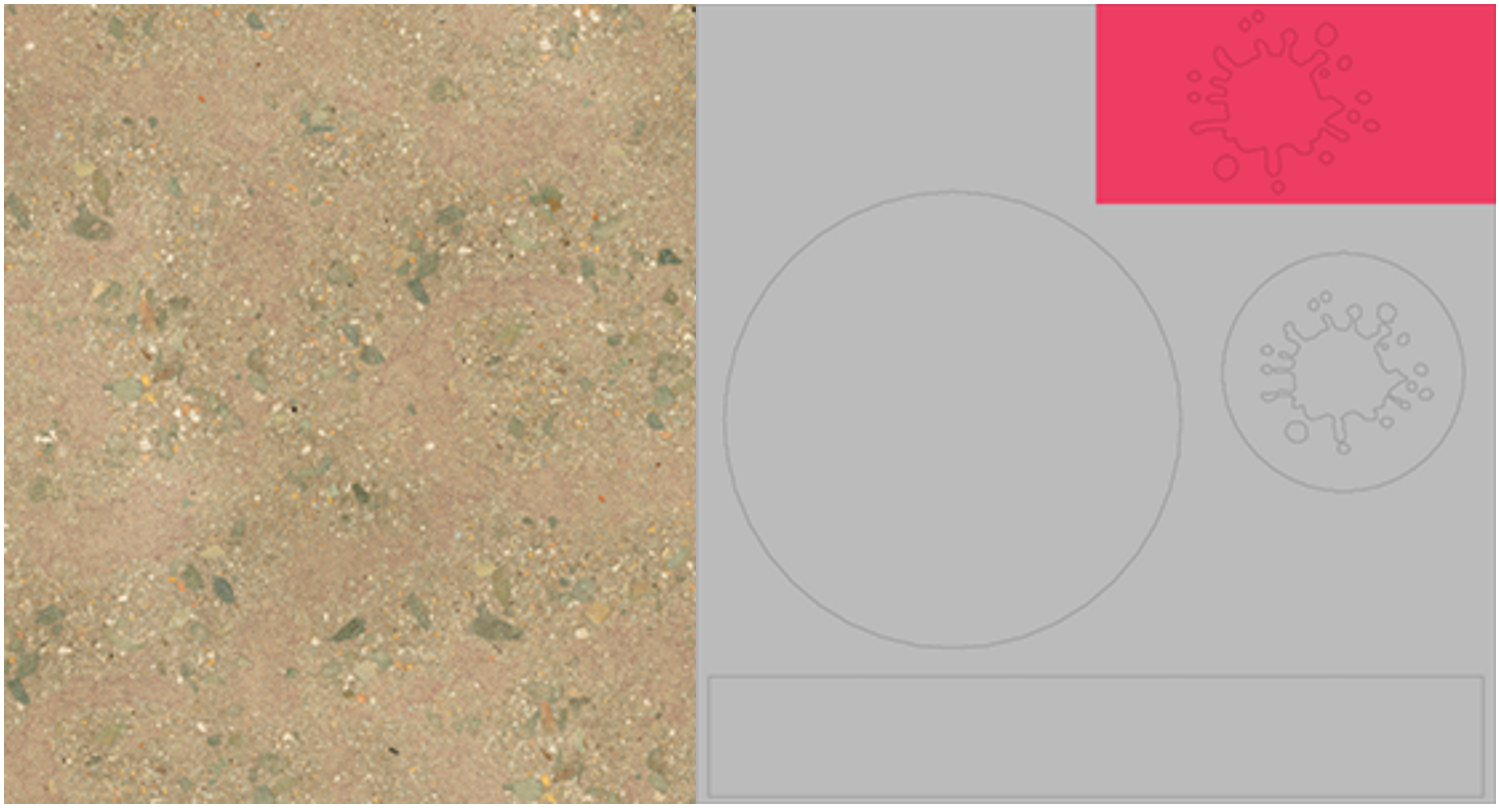
Merge Node Properties

Mode	*dropdown*	Selects the blend mode that you want to apply in this Merge node. See and **Advanced** for more information.
Amount	*text input/slider*	Adjusts the blend amount, **0** showing the **Base** input, and **1** showing the result of the blending operation.
Use Blending Colorspace	*dropdown*	When a Merge node is connected to an upstream node that uses its own size, depth, and colorspace properties, such as a Bake Point or Channel node, you need to consider whether **Use Blending Colorspace** is enabled or not. When enabled, it reduces color artifacts in the output of end point nodes that use color data. **Note:** See Bake Point Node and Channel Node for more information about the end point nodes. If your end point node has a **Depth** of **8bit (Byte)** and does not have **Scalar Data** enabled, it is recommended you set **Use Blending Colorspace** to **Enable** to reduce color artifacts in the output of the end point node.

		It is recommended you set **Use Blending Colorspace** to **Disable** if your end point node has a **Depth** of **16bit (Half)** or **32bit (Float)**, or if **Scalar Data** is enabled, as the blending operation is not required for non-color data or depths larger than 8-bit. **Note:** When connecting Merge nodes to end point nodes that 8-bit color data, a dialog appears asking if you would like to automatically enable **Use Blending Colorspace** on the Merge node.

Modes

The **Mode** option determines which mathematical calculation is used to merge the **Base** and **Over** inputs.



Base input **Over** input

Normal

Displays the **Over** input while the **Base** input stays hidden.

The **Normal** mode (Mari's default mode) is used for layering operations.

You can adjust the **Amount** to set how much of the **Over** input to show. Decreasing the **Amount** allows you to show the **Base** input.

Copy

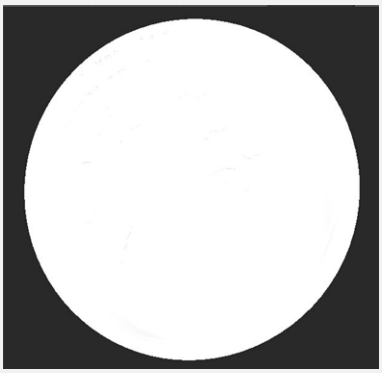
This copies the information from the **Over** input and doesn't involve any blending operations.

Note: Adjusting the **Amount** has no effect.

Clear

Subtracts the opacity of the **Over** input from the **Base** input. In other words, it erases texture from the **Base** input using the **Alpha** component of the **Over** input.

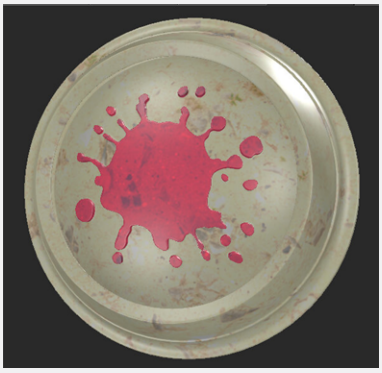
For example, an **Over** input with 80% opacity leaves a **Base** input with 20% opacity.



Darken

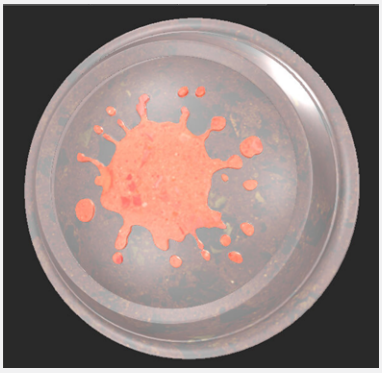
Darken

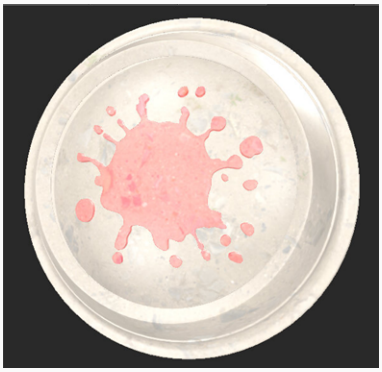
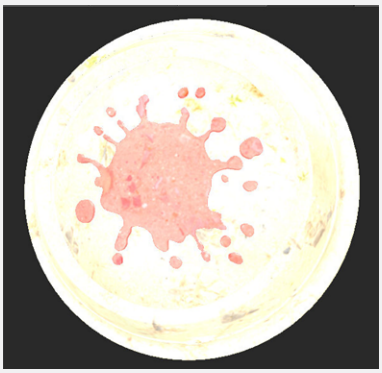
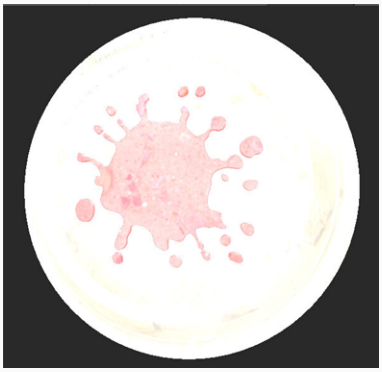
Gives each pixel the darker of two possible values: the **Base** or the **Over** (whichever is darker).



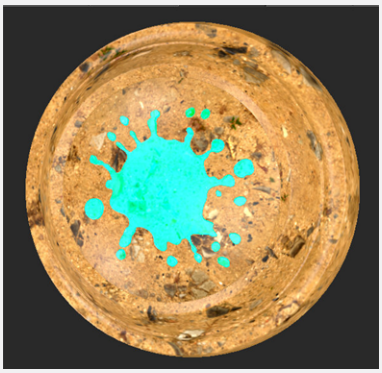
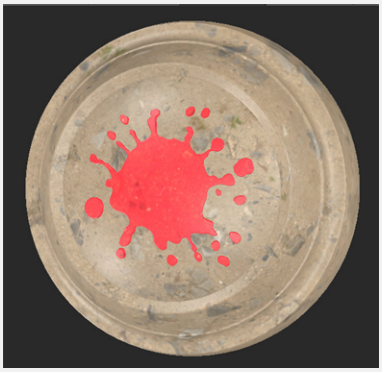
Multiply Darkens the color. Mari multiplies the **Base** color information by the **Over** color information.	
Color Burn Burns the **Base**, using the contents of the **Over** as input for the burn operation.	

Lighten

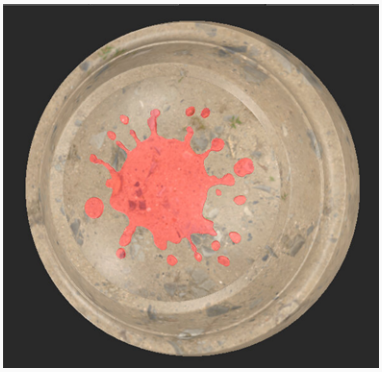
Lighten Gives each pixel the lighter of two possible values: the **Base** or **Over** (whichever is lighter).	

Screen Lightens the image. Mari divides the **Base** by the **Over**. (This is the opposite of Multiply.)	
Color Dodge Dodges the **Base**, using the contents of the **Over** to control the degree of dodging.	
Add Adds the values of the **Over** colors to the values on the **Base**.	

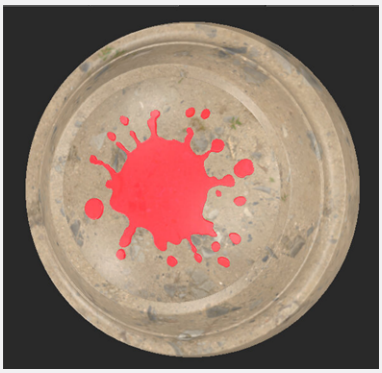
Contrast

Contrast The destination color moves away from the **Over** color by contrasting around the **Over** color. How much it contrasts depends on the alpha.	
Decontrast The destination color moves towards the **Over** color by decontrasting around the **Over** color. How much is decontrasts depends on the alpha.	
Overlay Uses the patterns and colors from the **Over**, but keeps the highlights and shadows from the **Base**.	

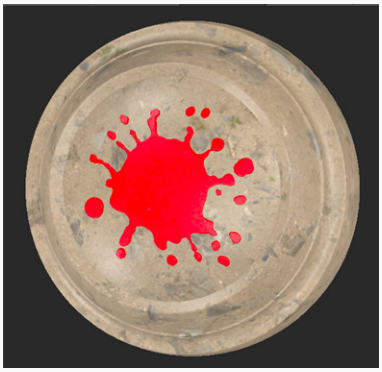
Soft Light
In dark areas in the **Base** burns the **Over** onto the **Base**. In lighter areas, dodges it on.



Hard Light
Mimics a harsh spotlight. Makes light areas lighter (using a screen effect), and dark areas darker (using a multiply effect).



Vivid Light
In dark areas in the **Base**, adds the **Over** with more contrast. In light areas, adds it with less contrast.

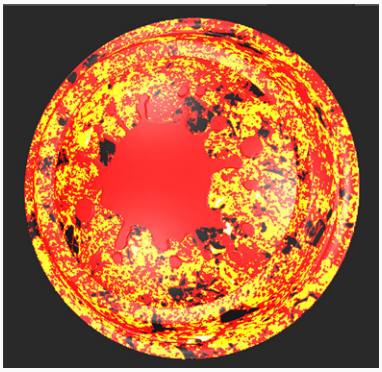


Pin Light
For light areas in the **Over**, replaces pixels in the **Base** that are darker than the corresponding pixels in the **Over**.

For dark areas in the **Over**, replaces pixels in the **Base** that are lighter than the corresponding pixels in the **Over**.

(Otherwise, leaves the **Base** unchanged.)

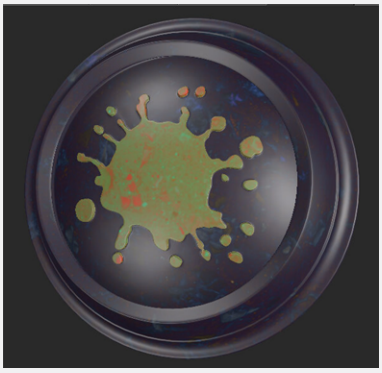
Hard Mix
First performs a vivid blend, then thresholds the colors to their extreme values. The result is that the pixels can be one of eight colors - black, white, red, green, blue, cyan, magenta, or yellow.

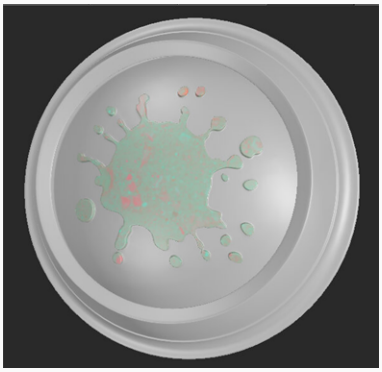
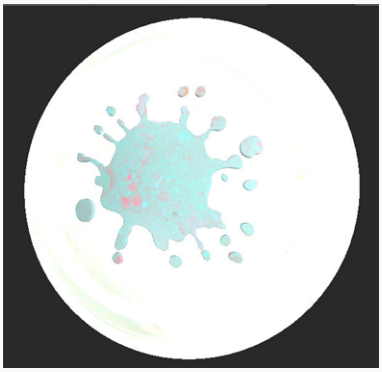


Sponge Desaturate
Uses the **Over** as the input to desaturate the pixels on the **Base**.

Inversion

Difference
Inverts the colors. Mari subtracts the values in the **Over** from those in the **Base**.

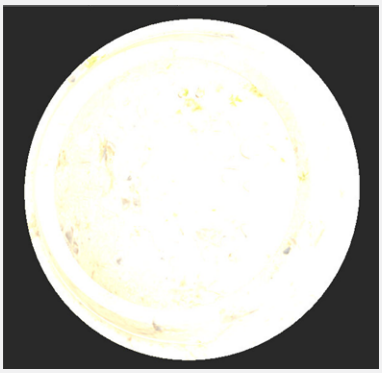


Exclusion Inverts the colors, but with a lowered contrast.	
Invert Uses the **Over** as the input to invert the pixels on the **Base**.	
Inverse Difference Subtracts the colors in the **Over** from the **Base**, and then inverts the result.	

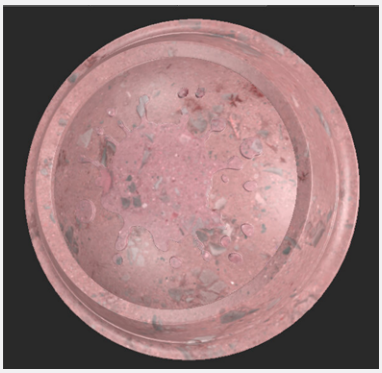
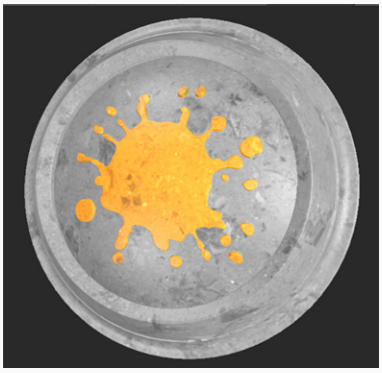
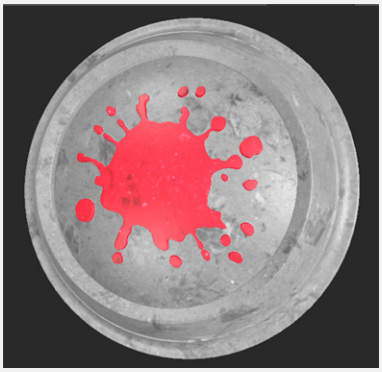
Burn

Burn Highlights Burns the **Base** based on the highlights in the **Over**.	
Burn Midrange Burns the **Base** based on the middle range (that is, not the lightest or darkest pixels) of the **Over**.	
Burn Shadows Burns the **Base** based on the contents of the **Over**, with darker pixels being burned harder.	

Dodge

Dodge Highlights Dodges the **Base** based on the highlights in the **Over**.	
Dodge Midrange Dodges the **Base** based on the middle range (that is, not the lightest or darkest pixels) of the **Over**.	
Dodge Shadows Dodges the **Base** based on the contents of the **Over**, with darker pixels being dodged harder.	

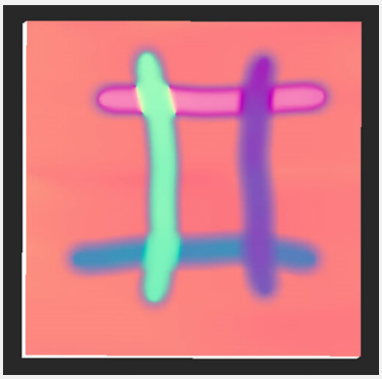
Component

Hue Takes the luminance and saturation from the **Base** pixels, with the hue from the **Over**.	
Saturation Takes the luminance and hue of the **Base** color, with the saturation from the **Over**.	
Color Takes the luminance of the **Base**, and the color and saturation of the **Over**	
Luminance Takes the hue and saturation of the **Base** color, and the luminance of the **Over**. (This is the opposite of the **Color** blend mode.)	

Vectors

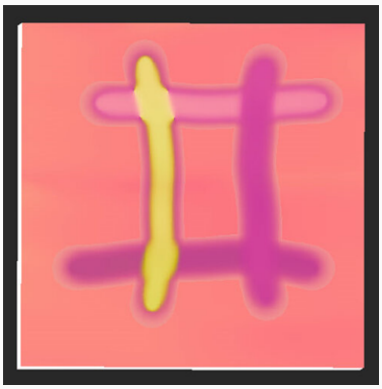
Paint Flow Vectors

Changes the **Over** to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. Unlike **Nudge Flow Vectors**, the **Paint Flow Vectors** mode blends between the vectors instead of adding them. This mode is best for use with flow maps.



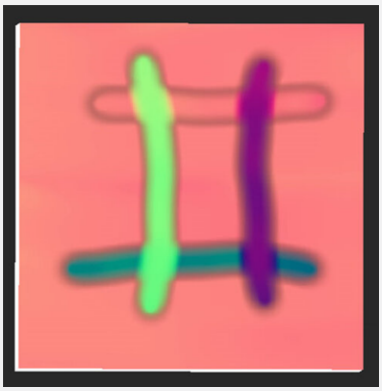
Nudge Flow Vectors

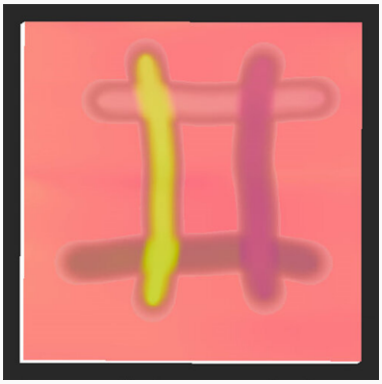
Changes the **Over** to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. The **Nudge Flow Vectors** mode adds the vectors together to get the blend result, and is best for use with flow maps.



Paint Normal Vectors

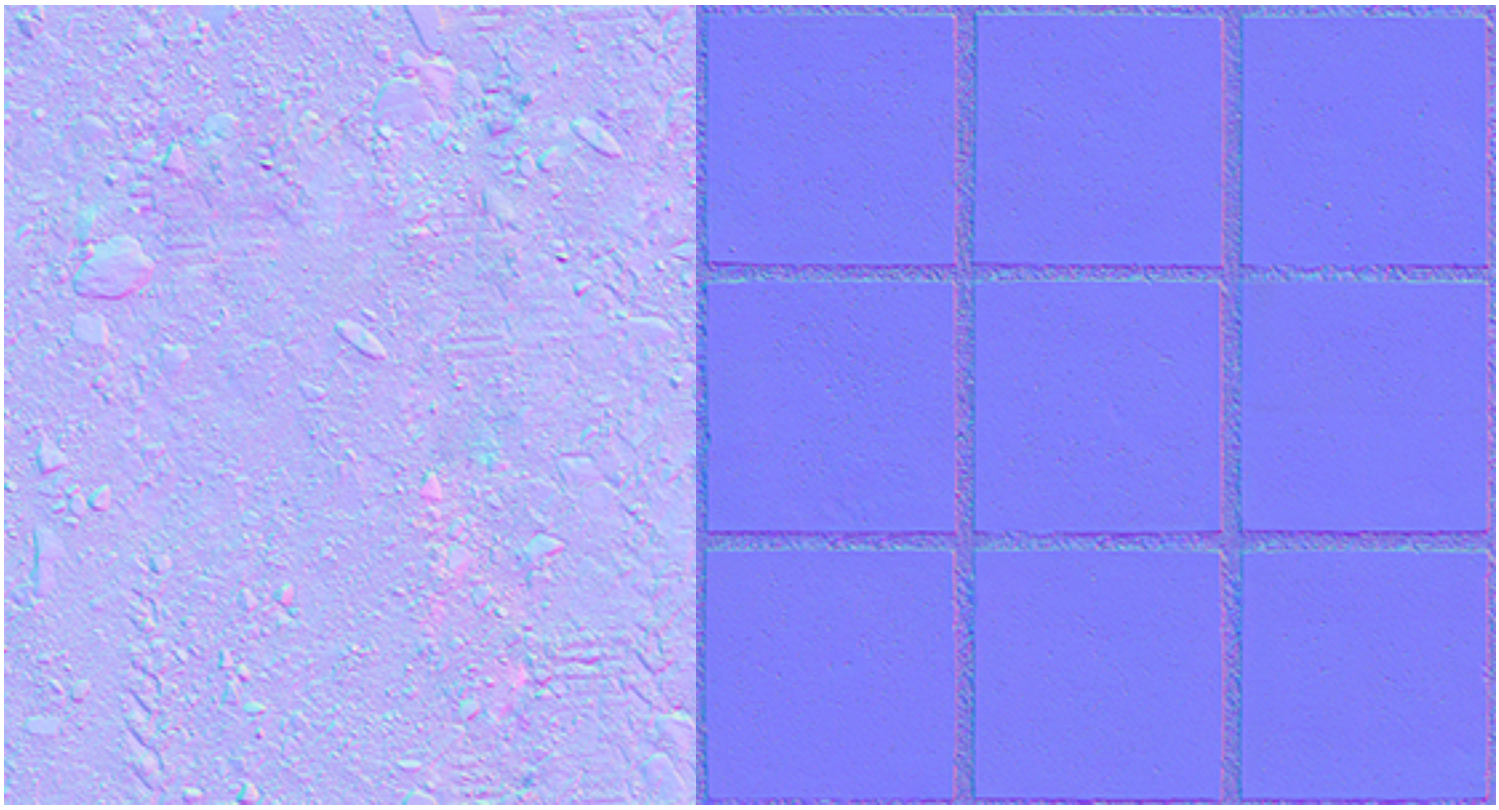
Changes the **Base** to tangent vector space. Allows the z value to face outward in 3D space, so that vectors can face out from the object's surface anywhere on the object. Unlike **Nudge Normal Vectors**, the **Paint Normal Vectors** mode blends between the vectors instead of adding them. This mode is best for use with normal maps.



Nudge Normal Vectors Changes the **Over** to tangent vector space. Allows the z value to face outward in 3D space, so that vectors face out from the object's surface anywhere on the model. The **Nudge Normal Vectors** mode adds the vectors together to get the blend result, and is best for use with normal maps.	
Scale Vectors This is most useful with the basic **Paint** tool, while preferably using grayscale paint. **Scale Vectors** uses the intensity of color to control the length of the normal in 3D space.	

Normal Maps

These are the original normal maps used for the following examples:

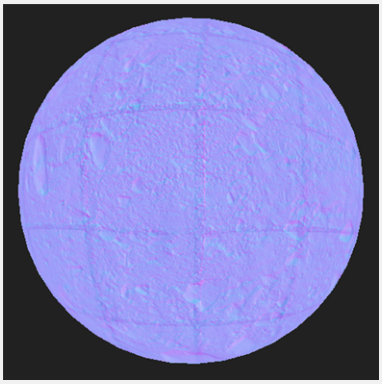


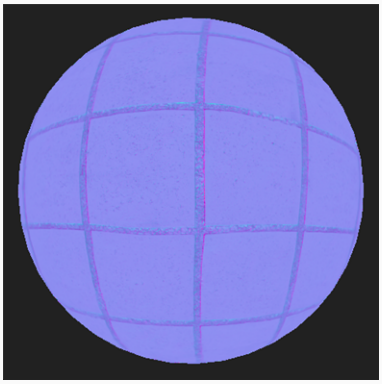
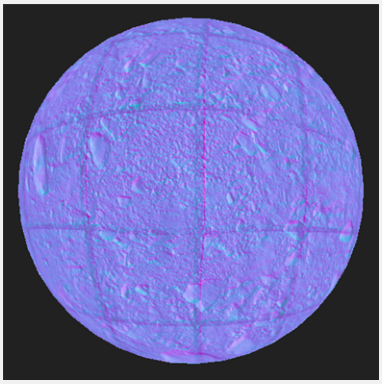
Base input

Over input

Add Normal Maps

Blends two RGB-encoded vector layers with the **Add** input by converting the raw values from the 0.0 to +1.0 colorspace-encoded range to the -1.0 to +1.0 vector range, applying the blend factor to the **Over** input, adding the values, then normalizing the values before converting it back to the 0.0 to +1.0 colorspace-encoded range.

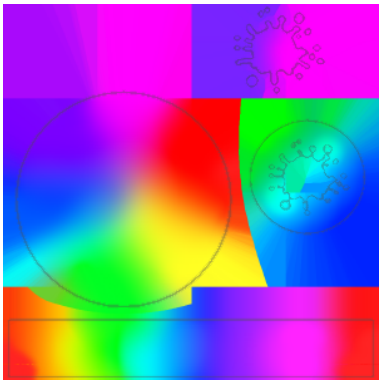


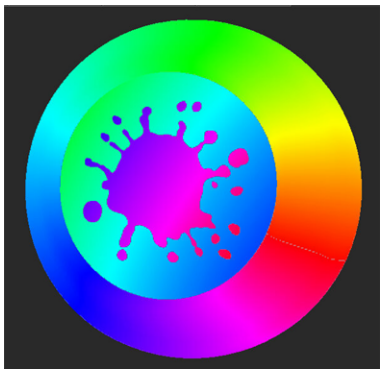
Mix Normal Maps Blends two RGB-encoded vector layers with the **Add** input by converting the raw values from the 0.0 to +1.0 colorspace-encoded range to the -1.0 to +1.0 vector range, applying the blend factor to the **Over** input, adding the values, then normalizing the values before converting it back to the 0.0 to +1.0 colorspace-encoded range.	
Mix Reoriented Maps Blends two RGB-encoded vector layers in a way that preserves the strength and details of both the base and the overlay inputs.	

Swizzle

Swizzling forces modifications to each color component of the blended result and applies to each pixel of the object.

This is the original texture used for the following examples.

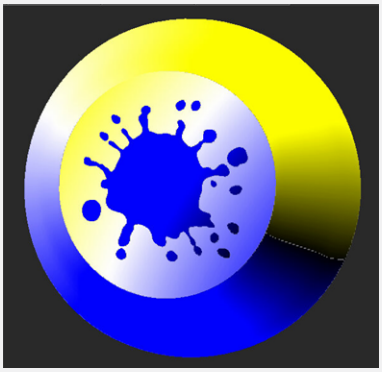




R

Modifies the value that is output from the **Red** color component.

- **Red** - By default, the **Red** component is set to output red color values.
- **Green** - Applies the value from the **Green** component.
- **Blue** - Applies the value from the **Blue** component.
- **Alpha** - Applies the value from the **Alpha** component.
- **One** - overrides the component's value and set it to **1** (white).
- **Zero** - overrides the component's value and set it to **0** (black).

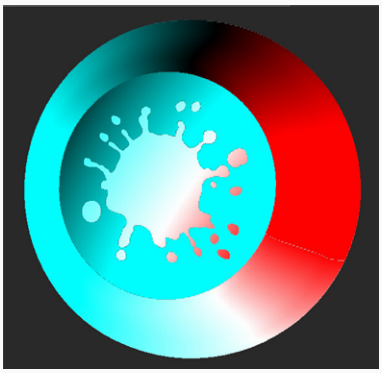


Set to **Green**.

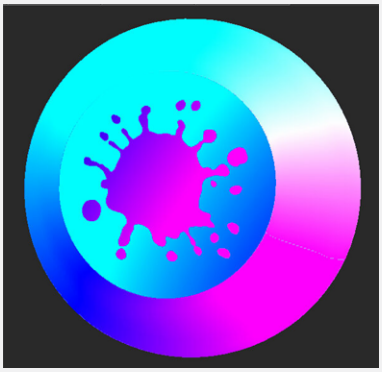
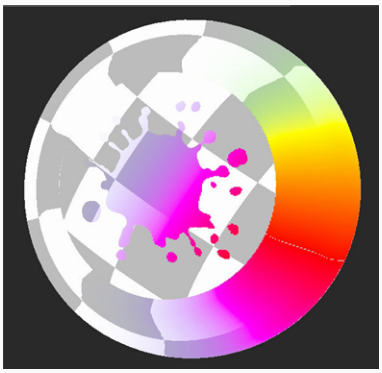
G

Modifies the value that is output from the **Green** color component.

- **Red** - Applies the value from the **Red** component.
- **Green** - By default, the **Green** component is set to output green color values.
- **Blue** - Applies the value from the **Blue** component.
- **Alpha** - Applies the value from the **Alpha** component.
- **One** - overrides the component's value and set it to **1** (white).
- **Zero** - overrides the component's value and set it to **0** (black).



Set to **Blue**.

B Modifies the value that is output from the **Blue** color component. • **Red** - Applies the value from the **Red** component. • **Green** - Applies the value from the **Green** component. • **Blue** - By default, the **Blue** component is set to output blue color values. • **Alpha** - Applies the value from the **Alpha** component. • **One** - overrides the component's value and set it to **1** (white). • **Zero** - overrides the component's value and set it to **0** (black).	 Set to **Alpha**.
A Modifies the value that is output from the **Alpha** color component. • **Red** - Applies the value from the **Red** component. • **Green** - Applies the value from the **Green** component. • **Blue** - Applies the value from the **Blue** component. • **Alpha** - By default, the **Alpha** component is set to output alpha color values. • **One** - overrides the component's value and set it to **1** (white). • **Zero** - overrides the component's value and set it to **0** (black).	 Set to **Red**.

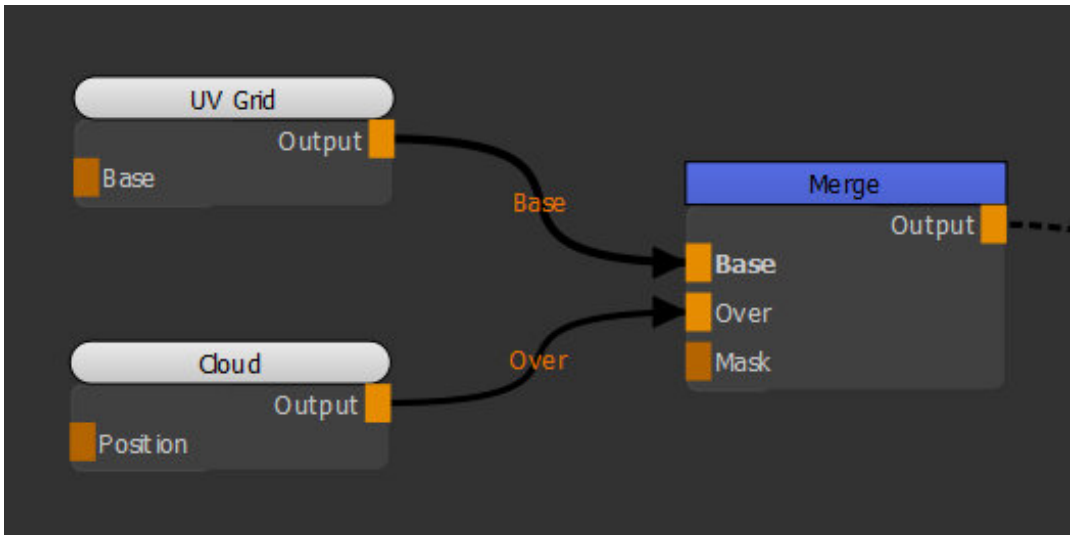
Advanced

Amount Enabled *dropdown*	**Enable** or **Disable** the ability to change the blend mode **Amount**. If **Amount** (see **Merge** tab) has already been modified and **Amount Enabled** is set to **Disable**, then the amount is not reset to the default. It is locked to the amount entered when **Disable** was set. However, when disabled, Mari considers that the **Amount** is set to 1.
Blending *dropdown*	Choose whether to modify the blending mode with **Basic** or **Advanced** settings. If you select **Basic**, the **Component** field and the **Over** and **Base** fields cannot be modified.
Component *dropdown*	Select either the **Grayscale**, **Red**, **Green**, **Blue**, or **Luminance** components to adjust when modifying the curve editors and applying the **Blending** mode.

Over *curve editor*	Based on the **Over** input values, this adjusts the amount of blending, using black, white and shades or gray. Drag to move points on the graph. Add points by clicking on the curve. Edit a point by right-clicking it. **Note:** In the **Merge** tab, the **Amount** adjusts the blend mode amount globally and not per input.
Base *curve editor*	Based on the **Base** input values, this adjusts the amount of blending, using black, white and shades or gray. Drag to move points on the graph. Add points by clicking on the curve. Edit a point by right-clicking it. **Note:** In the **Merge** tab, the **Amount** adjusts the blend mode amount globally and not per input.

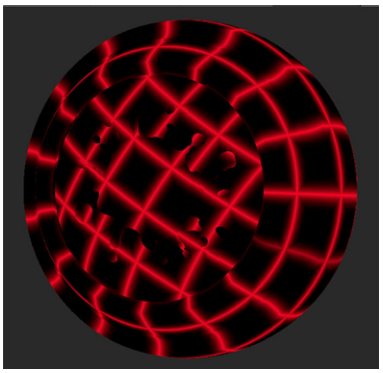
To illustrate the effects of **Base** and **Over** in the **Advanced Settings**, let's use the following example.

1. Create a UV Grid node and a Cloud node and connect them respectively in the **Base** and **Over** inputs of the Merge node.

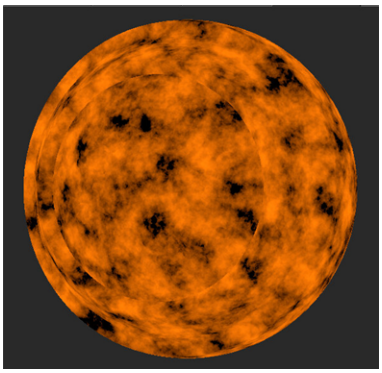


Adding a UV Grid node and a Cloud node to the scene.

These are the visual outputs of each separate node.

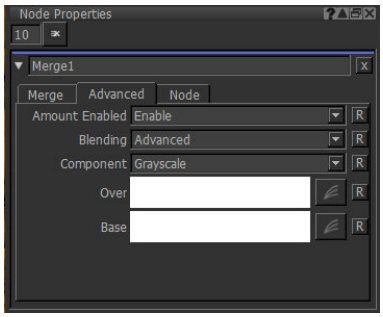


UV Grid node connected to **Base**.
Texture hidden under the Cloud node.

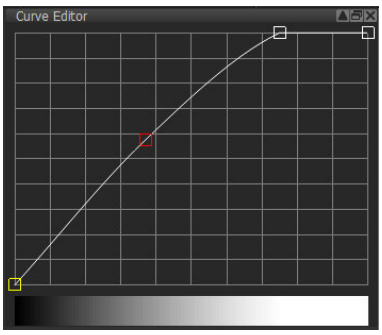


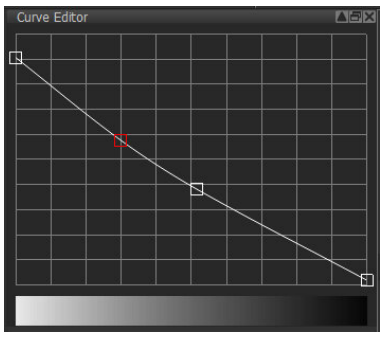
Cloud node connected to **Over**.
Visible texture.

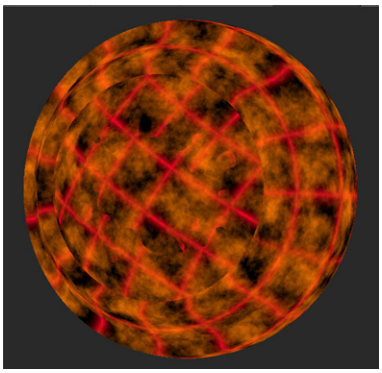
2. In the Merge node's **Node Properties**, open the **Advanced** tab, and set the **Blending** to **Advanced**. This allows you to modify the **Component**, **Over** and **Base** fields.



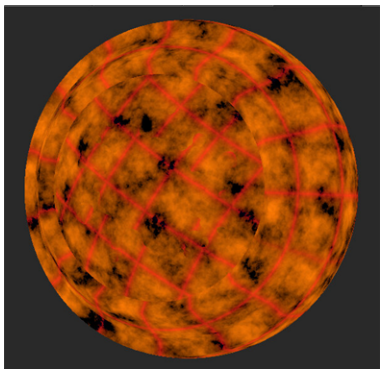
3. Tweak the **Over** and **Base** using the curve editor.







This adjusts the amount of blending of the **Over** input.

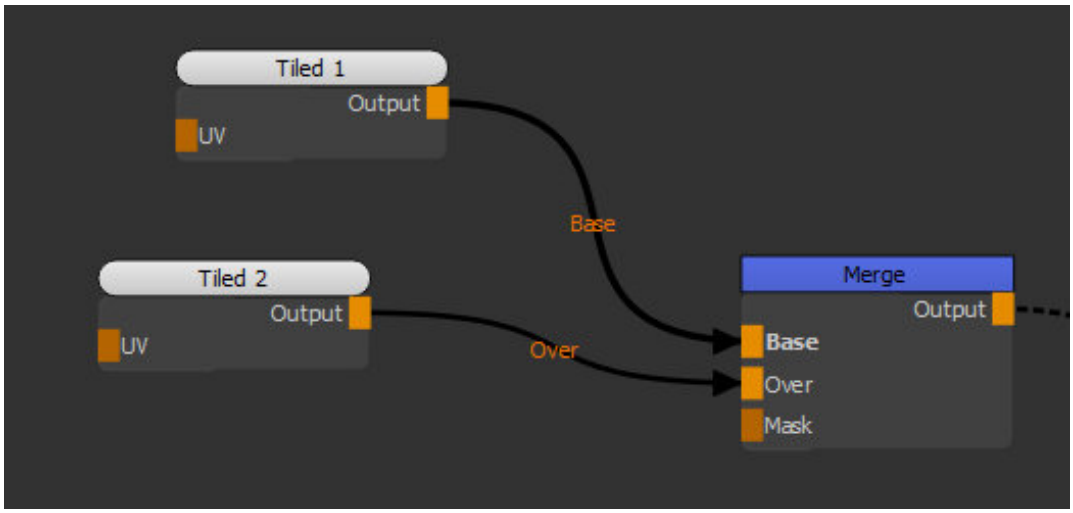


This adjusts the amount of blending of the **Base** input.

Merge: Node Graph Workflow Example

In this example, let's use the Merge node as a layering operation.

1. Create two Tiled nodes and connect them to the **Base** and **Over** inputs of the Merge node.



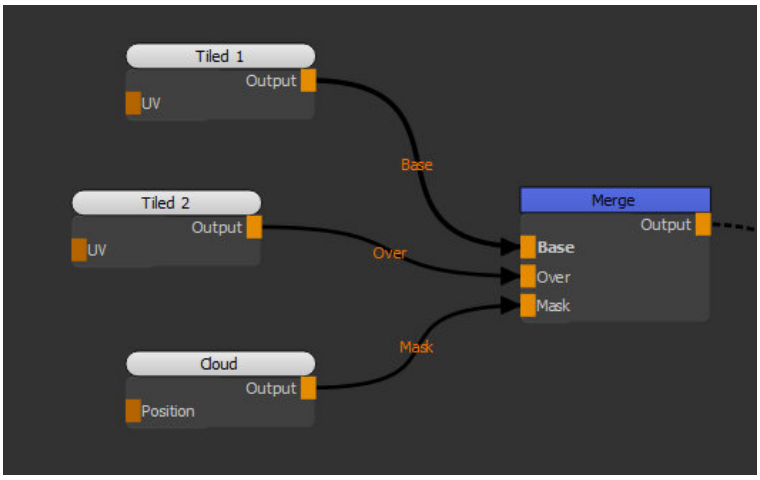
Adding two Tiled nodes to the scene.

*Texture in Tiled 1 node connected to **Base** input.*

*Texture in Tiled 2 node connected to **Over** input.*

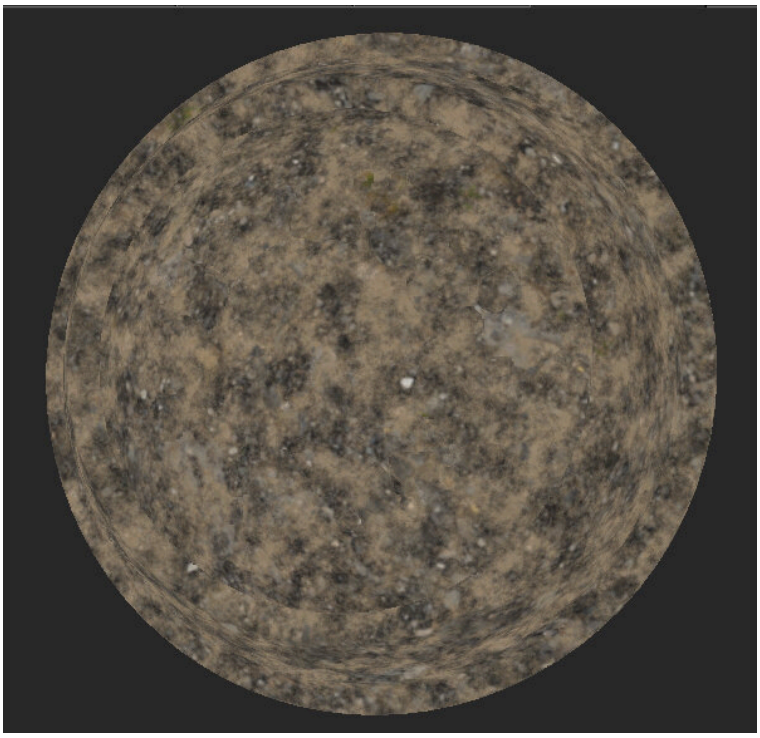
The **Blend** mode is set to **Normal**, which displays the **Over** input only.

2. Connect a Cloud node to the **Mask** input to control how much blending is applied.



Adding a Cloud node as a mask to control how much blending is applied.

Notice how both textures show on the object.



Tiled 1 and Tiled 2 are blended together.
This is achieved by using the Cloud node as a mask.

Ceil Node

Access: **Nodes** > **Math** > **Ceil**

$$\mathrm{ceil}(x) = \lceil x \rceil$$

The Ceil node rounds up values to the nearest whole number or integer. For example, 0.3 becomes 1, 1.7 becomes 2, 3.2 becomes 4.

Note: The Ceil node is similar to the Floor node operation, except, instead of rounding down to the nearest whole number, Ceil rounds up. For more information, see the Floor node.

Input	Output
3.75, 5.08, 2.49, 1.0	4, 6, 3, 1
5.0, 6.5, 7.2, 1.0	5, 7, 8, 1
0.0, 0.3, 0.7, 0.0	0, 1, 1, 0

Ceil Node Inputs

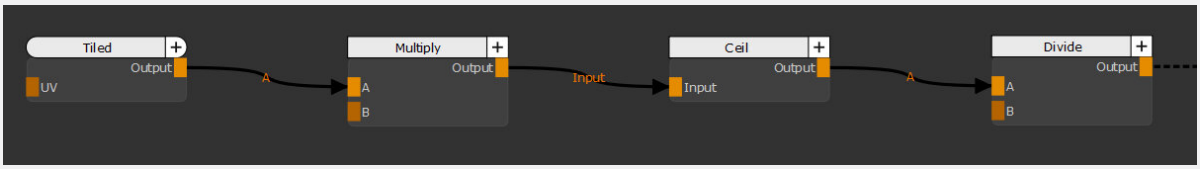
Input	The input values to be rounded up to the nearest integer. x

Ceil Node Outputs

Output	The input after rounding up to the nearest whole number. $\lceil x \rceil$

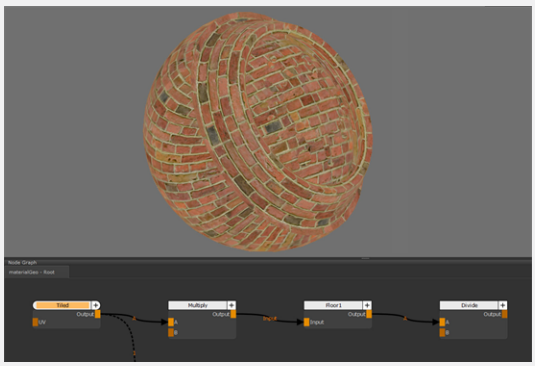
Workflow Example

The Ceil node can be used to create a simple posterize effect from a texture color input. To define the number of levels in the posterize effect, you can use the Multiply and Divide nodes, in the following way, where the Multiply and Divide **B** inputs are the number of levels you would like in the posterize effect:

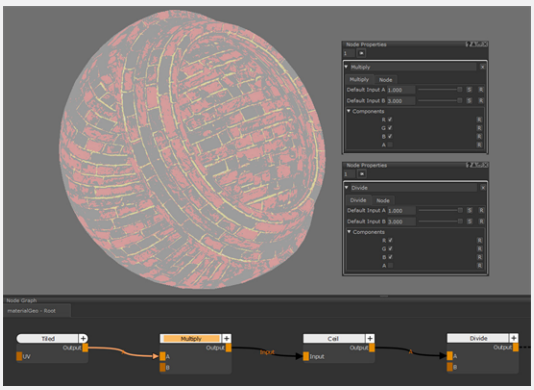


Workflow to create a posterize effect using a Ceil node

> **Note:** The Ceil node produces a similar posterize effect to the Floor node but because it rounds the input values up, the resulting colors will appear lifted.

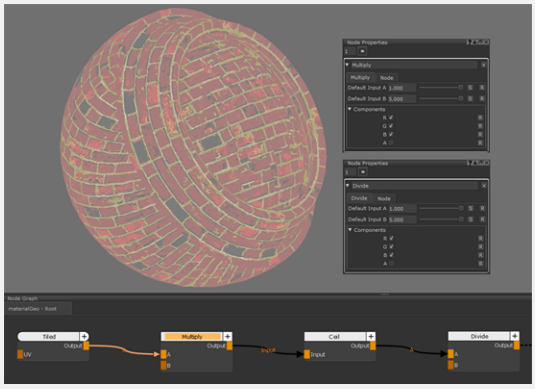


Input Texture



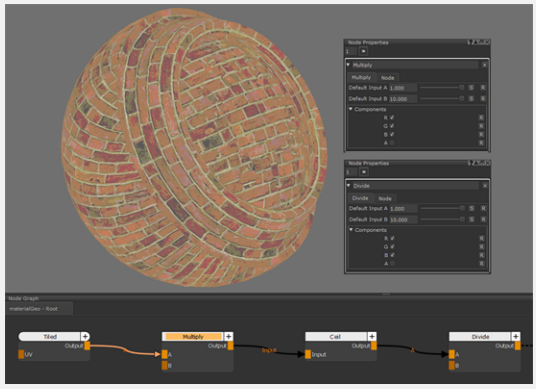
Multiply and Divide **B** input of 3.

Resulting in 3 posterize levels.



Multiply and Divide **B** input of 5.

Resulting in 5 posterize levels.



Multiply and Divide **B** input of 10.

Resulting in 10 posterize levels.

Distance Node

Access: **Nodes** > **Math** > **Distance**

The Distance node outputs the distance between two input vectors using the calculation:

$$d = \sqrt{((r_2 - r_1)^2 + (g_2 - g_1)^2 + (b_2 - b_1)^2 + (a_2 - a_1)^2)}$$

The Distance node can be especially useful when calculating the distance between 2D or 3D positions. The output is a single value copied across each component for the vector output, except for the alpha component, which retains its original input value.

Distance Node Example

Input	Calculation	Output
InputA = 0.0, 0.0, 0.0, 0.0 InputB = 1.0, 0.0, 0.0, 0.0	$d = \sqrt{((1.0 - 0.0)^2 + (0.0 - 0.0)^2 + (0.0 - 0.0)^2 + (0.0 - 0.0)^2)}$ $d = 1.0$	1.0, 1.0, 1.0, 1.0
InputA = 0.0, 0.0, 0.0, 0.0 InputB = 0.0, 3.0, 4.0, 1.0	$d = \sqrt{((0.0 - 0.0)^2 + (3.0 - 0.0)^2 + (4.0 - 0.0)^2 + (0.0 - 0.0)^2)}$ $d = 5.0$	5.0, 5.0, 5.0, 1.0

Note: Alpha components retain their original value even if they are included in the calculation.

Tip: To learn more about the distance equation solution, see the **3D Distance Calculator** on **calculatorsoup**.

Distance Node Inputs

InputA	The first vector from which the distance is calculated.
InputB	The second vector to which the distance is calculated.

Distance Node Properties

R *checkbox*	When checked, includes the red component in the distance calculation.
G *checkbox*	When checked, includes the green component in the distance calculation.
B *checkbox*	When checked, includes the blue component in the distance calculation.
A *checkbox*	When checked, includes the alpha component in the distance calculations. However, the alpha component is passed through to the output as its original value. For example, an input alpha value of 1.0 is output as an alpha value of 1.0.

Floor Node

Access: **Nodes** > **Math** > **Floor**

$$\text{floor}(x) = \lfloor x \rfloor$$

The Floor node rounds down values to the nearest whole number or integer.

Note: The Floor node is similar to the Ceil node operation, except instead of rounding up to the nearest whole number, Floor rounds down. For more information, see the Ceil node.

Input	Output
3.75, 5.08, 2.49, 1.0	3, 5, 2, 1
5.0, 6.5, 7.2, 1.0	5, 6, 7, 1
0.0, 0.3, 0.7, 0.0	0, 0, 0, 0

Floor Node Inputs

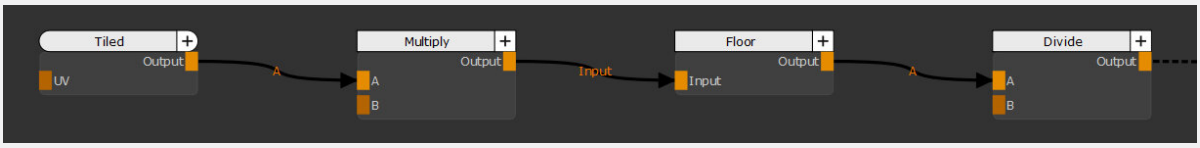
Input	The input values to be rounded down to the nearest integer. x

Floor Node Outputs

Output	The input after rounding down to the nearest whole number. $\lfloor x \rfloor$

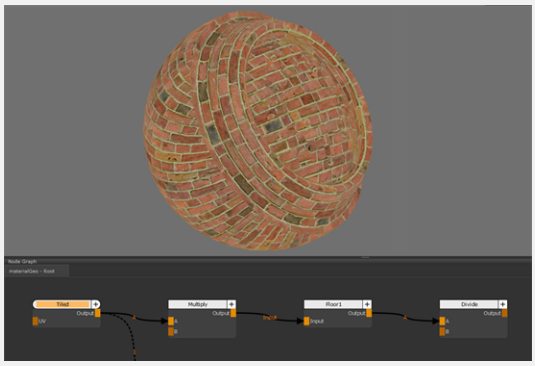
Floor Node Workflow Example

The Floor node can be used to create a simple posterize effect from a texture color input. To define the number of levels in the posterize effect, you can use the Multiply and Divide nodes, in the following way, where the Multiply and Divide **B** inputs are the number of levels you would like in the posterize effect:

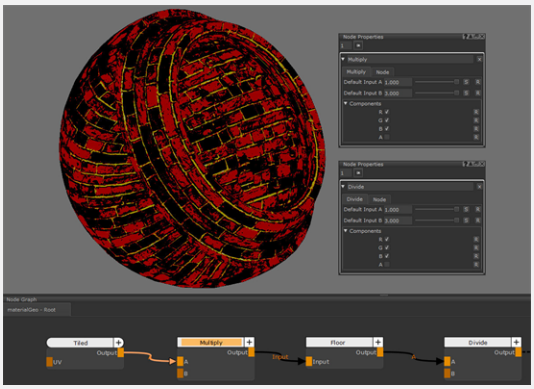


Workflow to Create a Posterize Effect using a Floor Node

Note: The Floor node produces a similar posterize effect to the Ceil node but because it rounds the input values down, the resulting colors will appear darker.

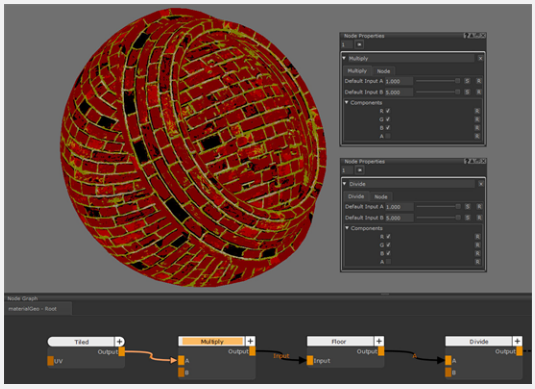


Input Texture



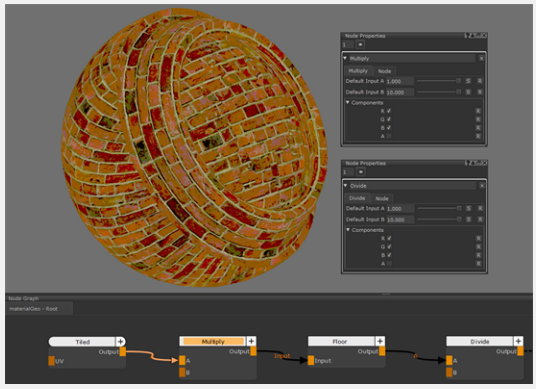
Multiply and Divide **B** input of 3.

Resulting in 3 Posterize Levels.



Multiply and Divide **B** input of 5.

Resulting in 5 Posterize Levels.



Multiply and Divide **B** input of 10.

Resulting in 10 Posterize Levels.

Fract Node

Access: **Nodes** > **Math** > **Fract**

$$\mathrm{frac}(x) = x - \lfloor x \rfloor$$

The Fract node calculates the fractional part (frac) of an input value by subtracting the floor $\lfloor x \rfloor$ of the input value from the input value. The Fractnode then outputs the result which is only the decimal place remainder of the input.

Input	Output
5.385, 0.164, 3.065, 1.0	0.385, 0.164, 0.065, 0.0
267.19, 105.096, 25.483, 1.0	0.19, 0.096, 0.483, 0.0
3.0, 7.0, 4.0, 1.0	0.0, 0.0, 0.0, 0.0

Note: For more information on the frac calculation, see **here** for an explanation provided by **MathWorks**. For information about the floor $\lfloor x \rfloor$ calculation, see Floor node.

The Fract node can be used in multiple ways, one workflow is to use the node to determine where a certain UV coordinate is from within a patch. Using a Fract node after a UV will output just its position within the patch.

Another workflow is to use the Fract node to repeat a part of your texture that's looked up by UV coordinates across each patch. For example, using a Fract node before a Gradient node to repeat the gradient across each patch.

Fract Node Inputs

Input	The input value of which the fractional part is calculated. x

Fract Node Outputs

Output	The decimal remainder of the number only. $x - \lfloor x \rfloor$

Fract Node Properties

No node properties.

Length Node

Access: **Nodes** > **Math** > **Length**

The Length node takes an input vector and outputs the length (or magnitude) of the vector. The following formula is used to calculate the length of the vector:

$$|\mathbf{u}| = \sqrt{\mathbf{u}_x{}^2 + \mathbf{u}_y{}^2 + \mathbf{u}_z{}^2 + \mathbf{u}_w{}^2}$$

Where $\mathbf{u}$ is the vector [$\mathbf{u}_x$ $\mathbf{u}_y$ $\mathbf{u}_z$ $\mathbf{u}_w$] and $|\mathbf{u}|$ is the length of the vector.

The output is a vector and the alpha channel always has a value of 1.0. Therefore the output is in the format:

[$|\mathbf{u}|$ $|\mathbf{u}|$ $|\mathbf{u}|$ 1]

Tip: For more information on the vector length equation solution, see the **Vector Magnitude Calculator** by OnlineMSchool.

Length Node Examples

Input	Calculation	Output
$\mathbf{u}$ = [$\mathbf{u}_R$ $\mathbf{u}_G$ $\mathbf{u}_B$ $\mathbf{u}_A$] = [3 1 2 0]	$\sqrt{3^2 + 1^2 + 2^2 + 0^2}$	$\lvert\mathbf{u}\rvert$ = 3.742 [3.742 3.742 3.742 1.0]
$\mathbf{u}$ = [$\mathbf{u}_R$ $\mathbf{u}_G$ $\mathbf{u}_B$ $\mathbf{u}_A$] = [10 5 4 0.5]	$\sqrt{10^2 + 5^2 + 4^2 + 0.5^2}$	$\lvert\mathbf{u}\rvert$ = 11.885 [11.885 11.885 11.885 1.0]
$\mathbf{u}$ = [$\mathbf{u}_R$ $\mathbf{u}_G$ $\mathbf{u}_B$ $\mathbf{u}_A$] = [12 25 6 1]	$\sqrt{12^2 + 25^2 + 6^2 + 1^2}$	$\lvert\mathbf{u}\rvert$ = 28.390 [28.390 28.390 28.390 1.0]

Length Node Inputs

Input	The vector to calculate the length of. $\mathbf{u}$

Length Node Outputs

Outputs	The length of the vector. $\lvert\mathbf{u}\rvert$

Length Node Properties

R *checkbox*	When checked, the red component is included in the length calculation.
G *checkbox*	When checked, the green component is included in the length calculation.
B *checkbox*	When checked, the blue component is included in the length calculation.
A *checkbox*	When checked, the alpha component is included in the length calculation. **Note:** When the alpha is checked, its value is included in the length calculation however the alpha output will always be 1.0.

Mix Node

Access: **Nodes** > **Math** > **Mix**

The Mix node linearly interpolates between two color inputs based on an input blend value. If the **Blend Factor** is 0, color **A** is output. If the **Blend Factor** is 1, color **B** is output. A **Blend Factor** value between 0 and 1 outputs a relative mix between **A** and **B**.

Mix Node Inputs

A: The color to use as **A**. Defaults to **Default Input A** when no node is connected.

B: The color to use as **B**. Defaults to **Default Input B** when no node is connected.

Blend: The value to be use as the **Blend Factor**. Defaults to the value defined in **Blend Factor** when no node is connected.

Mix Node Properties

Default Input A *swatch*	The default color to be used as color **A**.
Default Input B *swatch*	The default color to be used as color **B**.
Blend Factor *text field, slider*	The blend value between color **A** and color **B**.
R *checkbox*	Specify whether to blend the red channel. When unchecked, the value from **A** is used.
G *checkbox*	Specify whether to blend the green channel. When unchecked, the value from **A** is used.

B *checkbox*	Specify whether to blend the blue channel. When unchecked, the value from **A** is used.
A *checkbox*	Specify whether to blend the alpha channel. When unchecked, the value from **A** is used.

Modulo Node

Access: **Nodes** > **Math** > **Modulo**

The Modulo node outputs the input A modulo the input B. The output is the remainder after dividing A by B, where A is the dividend and B is the divisor. For example:

Input A	1.0, 2.5, 4.25, 3.0
Input B	2.0, 2.0, 2.0, 2.0
Output (input A modulo input B)	= 1.0, 0.5, 0.25, 1.0

Modulo Node Inputs

Input	Takes the input as the dividend, or input A of the division operation. The remainder of the result of input A divided by input B is the modulo output.
Modulo	Takes the input as the divisor, or input B of the division operation. The remainder of the result of input A divided by input B is the modulo output.

Normalize Node

Access: **Nodes** > **Math** > **Normalize**

The Normalize node takes an input vector, of any length, and normalizes it to retain its direction, but display its length as unit length which is always a value of 1.0. This normalization process is useful to describe a vector's direction without regard to its length. The following formula is used to calculate the normalized vector:

$$\hat{\mathbf{u}} = \frac{\mathbf{u}}{|\mathbf{u}|}$$

Where $\mathbf{u}$ is the vector $[\mathbf{u}_x\ \mathbf{u}_y\ \mathbf{u}_z\ \mathbf{u}_w]$, $|\mathbf{u}|$ is the length of the vector and $\hat{\mathbf{u}}$ represents the normalized vector.

Normalize Node Examples

Input	Calculation	Output
$\mathbf{u} = [\mathbf{u}_R\ \mathbf{u}_G\ \mathbf{u}_B\ \mathbf{u}_A] = [3\ 1\ 2\ 0]$ $\lvert\mathbf{u}\rvert = 3.742$	$\hat{\mathbf{u}}_R = 3 / 3.742 = 0.802$ $\hat{\mathbf{u}}_G = 1 / 3.742 = 0.267$ $\hat{\mathbf{u}}_B = 2 / 3.742 = 0.534$ $\hat{\mathbf{u}}_A = 0 / 3.742 = 0$	$\hat{\mathbf{u}} = [0.802\ 0.267\ 0.534\ 0]$ $\lvert\mathbf{u}\rvert = 1.0$
$\mathbf{u} = [\mathbf{u}_R\ \mathbf{u}_G\ \mathbf{u}_B\ \mathbf{u}_A] = [10\ 5\ 4\ 0.5]$ $\lvert\mathbf{u}\rvert = 11.874$	$\hat{\mathbf{u}}_R = 10 / 11.874 = 0.842$ $\hat{\mathbf{u}}_G = 5 / 11.874 = 0.421$ $\hat{\mathbf{u}}_B = 4 / 11.874 = 0.337$ $\hat{\mathbf{u}}_A = 0.5 / 11.874 = 0.042$	$\hat{\mathbf{u}} = [0.842\ 0.421\ 0.337\ 0.042]$ $\lvert\mathbf{u}\rvert = 1.0$
$\mathbf{u} = [\mathbf{u}_R\ \mathbf{u}_G\ \mathbf{u}_B\ \mathbf{u}_A] = [12\ 25\ 6\ 1]$	$\hat{\mathbf{u}}_R = 12 / 28.373 = 0.423$	$\hat{\mathbf{u}} = [0.423\ 0.881\ 0.211\ 0.035]$

$\lvert \mathbf{u} \rvert = 28.373$	$\hat{\mathbf{u}}_G = 25 / 28.373 = 0.881$ $\hat{\mathbf{u}}_B = 6 / 28.373 = 0.211$ $\hat{\mathbf{u}}_A = 1 / 28.373 = 0.035$	$\lvert \mathbf{u} \rvert = 1.0$

Tip: For more information about normalized and unit vectors, see **Unit Vectors** provided by **Wikipedia**. For information about calculating the length, or magnitude, of a vector, see Length Node.

The use of the Normalize node is specific to normal map textures, or any textures that represent a unit length vector.

The Normalize node can be used to ensure that a normal map is behaving correctly after it has been edited manually or after it has been generated from a section of the Node Graph.

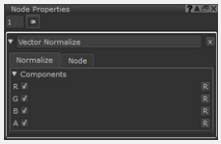
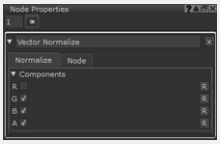
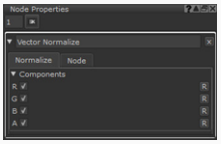
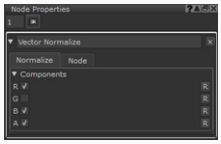
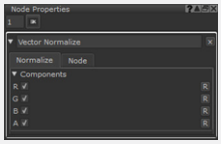
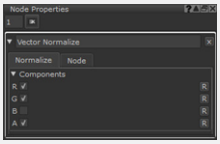
Normalize Node Inputs

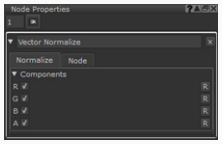
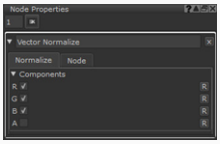
Input	The vector to be normalized. $\mathbf{u}$

Normalize Node Outputs

Output	The normalized vector. $\hat{\mathbf{u}}$

Normalize Node Properties

R *checkbox*	When checked, the red component is included in the normalization output. When unchecked, the red component will be left out of the normalization calculation and the input red value will pass through unchanged.  RGBA are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ \hat{u}_B\ \hat{u}_A]$  GBA are enabled so the output is: $[u_R\ \hat{u}_G\ \hat{u}_B\ \hat{u}_A]$
G *checkbox*	When checked, the green component is included in the normalization output. When unchecked, the green component will be left out of the normalization calculation and the input green value will pass through unchanged.  RGBA are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ \hat{u}_B\ \hat{u}_A]$  RBA are enabled so the output is: $[\hat{u}_R\ u_G\ \hat{u}_B\ \hat{u}_A]$
B *checkbox*	When checked, the blue component is included in the normalization output. When unchecked, the blue component will be left out of the normalization calculation and the input blue value will pass through unchanged.  RGBA are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ \hat{u}_B\ \hat{u}_A]$  RGA are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ u_B\ \hat{u}_A]$
A *checkbox*	When checked, the alpha component is included in the normalization output. When unchecked, the alpha component will be left out of the normalization calculation and the input alpha value will pass through unchanged.

	 RGBA are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ \hat{u}_B\ \hat{u}_A]$	 RGB are enabled so the output is: $[\hat{u}_R\ \hat{u}_G\ \hat{u}_B\ u_A]$

Power Node

Access: **Nodes** > **Math** > **Power**

$$a^b = a * \underbrace{\ldots}_{b\ times} * a$$

The Power node outputs the value of **A** raised to the power **B**. This can be useful to raise base color values to become darker or lighter, depending on the exponent.

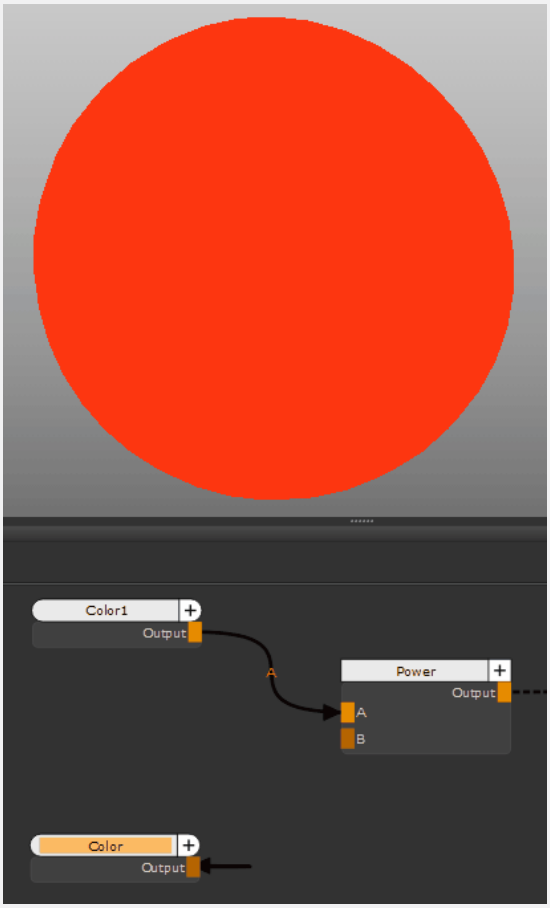
For the inputs **A** = [R G B A] and **B** = [X Y Z W], the Power node performs the following calculation:

R^X, G^Y, B^Z, A^W.

Power Node Examples

Input (RGBA)	Calculation	Output (RGBA)
Input **A** = 1.0, 0.213, 0.063, 1.0 Input **B** = 1.0, 0.454, 0.454, 1.0	$R = 1.0^{1.0}$ $G = 0.213^{0.454}$ $B = 0.063^{0.454}$ $A = 1.0^{1.0}$	1.0, 0.495, 0.285, 1.0

Input **A** = 0.5, 0.9, 0.1, 1.0 Input **B** = 0.1, 0.1, 0.1, 1.0	R = $0.5^{0.1}$ G = $0.9^{0.1}$ B = $0.1^{0.1}$ A = $1.0^{1.0}$	0.933, 0.989, 0.794, 1.0



A Color node with a value of 1.0, 0.213, 0.063 raised to the power of a Color node with a value of 1.0, 0.454, 0.454.

Power Node Inputs

A	The value to raise to the power **B**. When no node is connected, the value defaults to the value specified in **Default Input A**.
B	The power to raise **A** to. When no node is connected, the value defaults to the value specified in **Default Input B**.

Power Node Properties

Default Input A *text field, slider*	The value to use for **A** if no node is connected to the **A** input.
Default Input B *text field, slider*	The value to use for **B** if no node is connected to the **B** input.
R *checkbox*	If checked, the red component of **A** is raised to the red component of power **B**. If unchecked, the red component is passed through to the output as the original value of input **A**.
G *checkbox*	If checked, the green component of **A** is raised to the green component of power **B**. If unchecked, the green component is passed through to the output as the original value of input **A**.
B *checkbox*	If checked, the blue component of **A** is raised to the blue component of power **B**. If unchecked, the blue component is passed through to the output as the original value of input **A**.
A *checkbox*	If checked, the alpha component of **A** is raised to the alpha component of power **B**. If unchecked, the alpha component is passed through to the output as the original value of input **A**.

Square Root Node

Access: **Nodes** > **Math** > **Square Root**

$$\sqrt{x}$$

The Square Root node outputs the square root of the input. Using the Square Root node is a quick method to obtain a gain effect on your textures to lift and lighten the base color values.

Input (RGBA)	Calculation	Output (RGBA)
1.0, 1.0, 1.0, 1.0	√1.0, √1.0, √1.0, 1.0	1.0, 1.0, 1.0, 1.0
0.5, 0.5, 0.5, 1.0	√0.5, √0.5, √0.5, 1.0	0.716, 0.716, 0.716, 1.0
0.0, 0.0, 1.0, 1.0	√0.0, √0.0, √1.0, 1.0	0.0, 0.0, 1.0, 1.0

Note: Alpha components retain their original value even if they are included in the calculation.

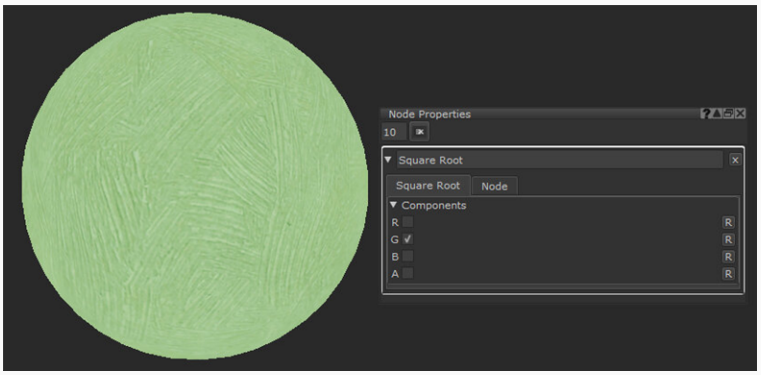
Square Root Node Inputs

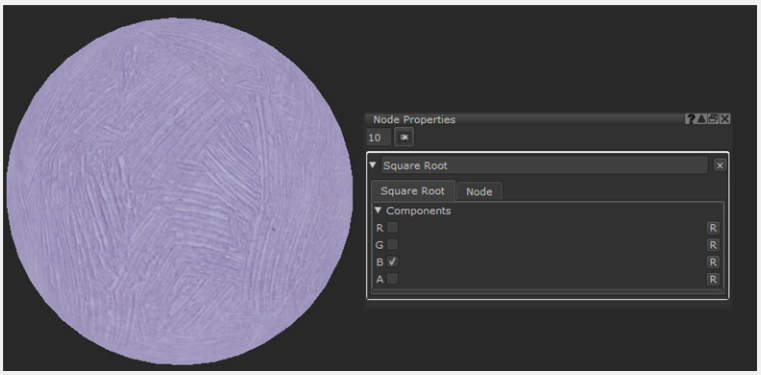
Input	The RGBA components of a vector input.

Square Root Node Outputs

Output	The square root of the input components which are selected in the **Node Properties**. **Note:** If any RGBA components are unchecked in the **Node Properties**, they pass through to the output unchanged.

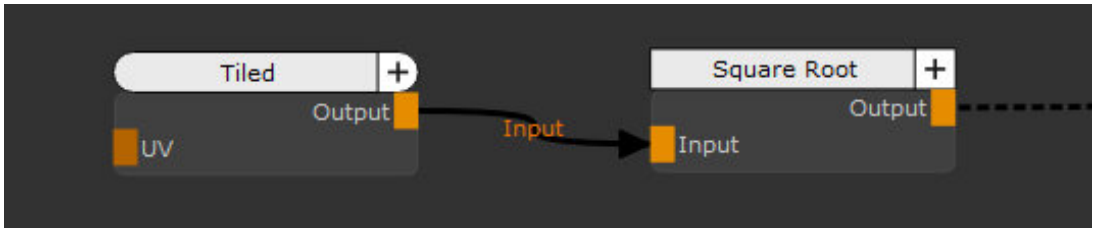
Square Root Node Properties

R *checkbox*	When checked, the red component float value of the input is square-rooted and the result is output as the red component. A Tiled node with an image before it has been plugged into a Square Root node. A Tiled node with an image fed into a Square Root node with the **R** component checked on.
G *checkbox*	When checked, the green component float value of the input is square-rooted and the result is output as the green component. A Tiled node with an image before it has been plugged into a Square Root node.  A Tiled node with an image fed into a Square Root node with the **G** component checked on.
B	When checked, the blue component float value of the input is square-rooted and the result is

checkbox	output as the blue component. A Tiled node with an image before it has been plugged into a Square Root node.  A Tiled node with an image fed into a Square Root node with the **B** component checked on.
A *checkbox*	This is checked off by default. The alpha component is passed through to the output as its original value. For example, an input alpha value of 1.0 is output as 1.0.

Square Root Node Workflow Example

Using the Square Root node in the Node Graph allows you to get the square root of floating point color values, which lightens the selected RGBA components. In the example, a Tiled node with an image is fed into a Square Root node to lighten the tiled image.



A Tiled node fed into a Square Root node with all four RGBA components selected results in a lighter image.

The texture from the Tiled node, before it has been fed into a Square Root node.

The results of the texture from feeding the Tiled node into the Square Root node.

If you are familiar with the Grade node, the **Gain** property of that node works in a similar way and gives you more overall control. However, if you need to quickly lighten your input colors, a Square Root node may prove more useful.

Vector Dot Node

Access: **Nodes** > **Math** > **Vector Dot**

The Dot Product applies the directional growth of one vector to another vector. The result is how much stronger the original vector has been made with a positive, negative, or zero scalar result. The Vector Dot node calculates and outputs the dot product (a single number) of two input vectors, which are in the format (*x, y, z, 1*).

Vector Dot Node Inputs

A	The vector input in the form (dot, dot, dot, 1) to be used as vector **A** in the dot product calculation of

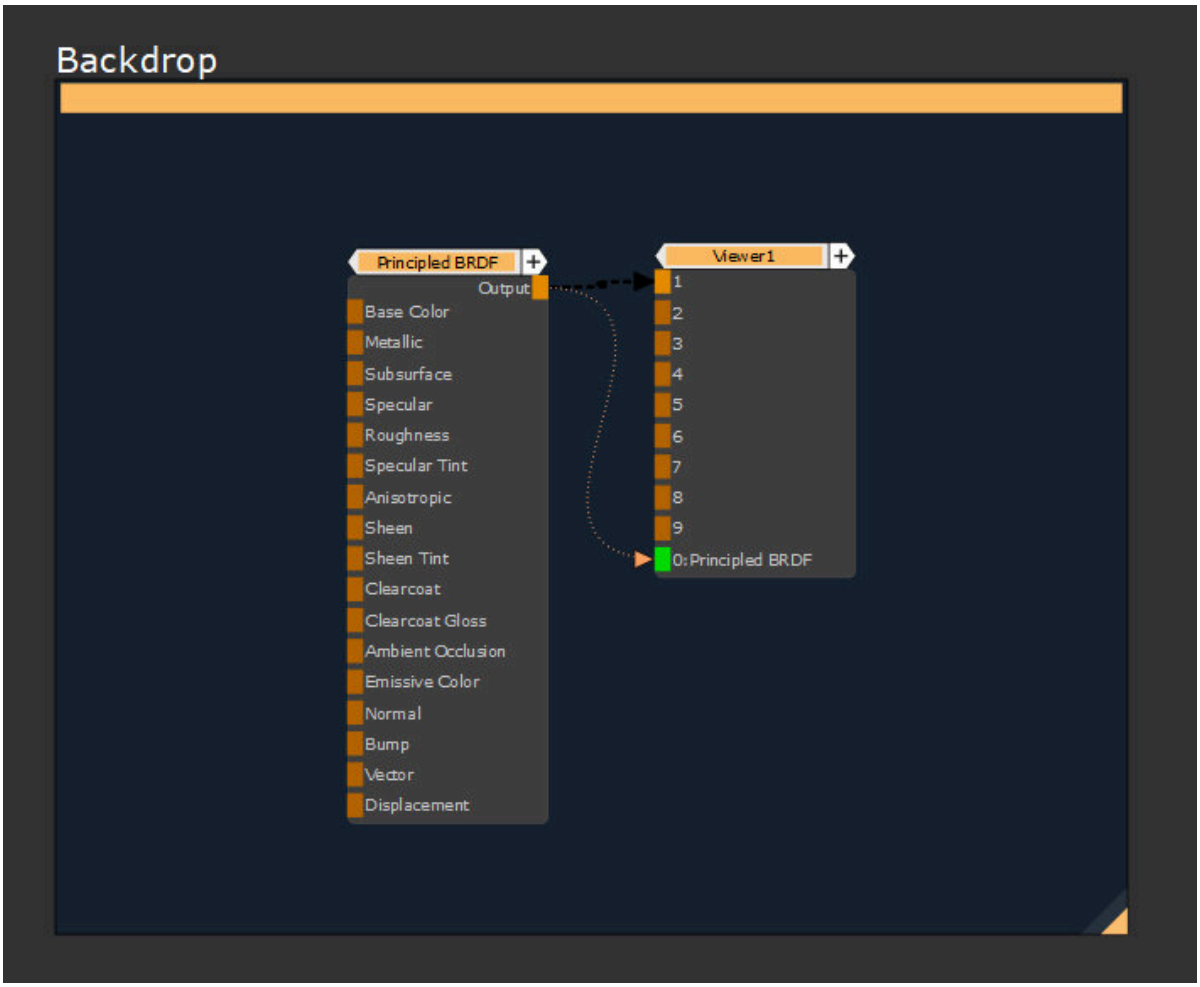
	$A \cdot B = (A * B) + (A * B)$.
B	The vector input in the form (dot, dot, dot, 1) to be used as vector **B** in the dot product calculation of $A \cdot B = (A * B) + (A * B)$.

Vector Dot Node Properties

Default Input A *text field, slider*	Values to be used as the input for vector **A** to calculate the dot product if no node is connected to the **A** input. **Tip:** Click **S** next to the slider to expand all 4 components of the vector.
Default Input B *text field, slider*	Values to be used as the input for vector **B** to calculate the dot product if no node is connected to the **B** input. **Tip:** Click **S** next to the slider to expand all 4 components of the vector.
R *checkbox*	When turned on, the red component is included in the dot product.
G *checkbox*	When turned on, the green component is included in the dot product.
B *checkbox*	When turned on, the blue component is included in the dot product.
A *checkbox*	When turned on, the alpha component is included in the dot product.

Backdrop Node

Access: **Nodes** > **Misc** > **Backdrop**



The Backdrop node is a great way to organise things inside your Node Graph. When you place a Backdrop node down, you create a colored rectangle which groups its contents.

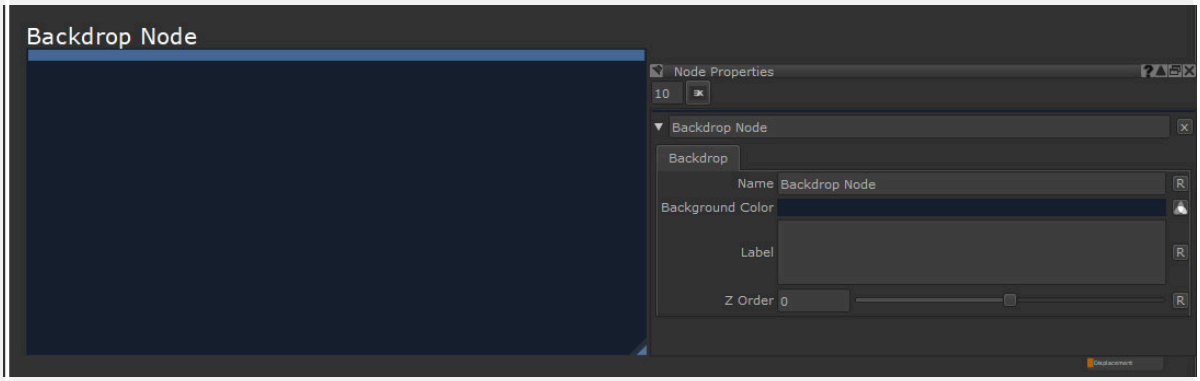
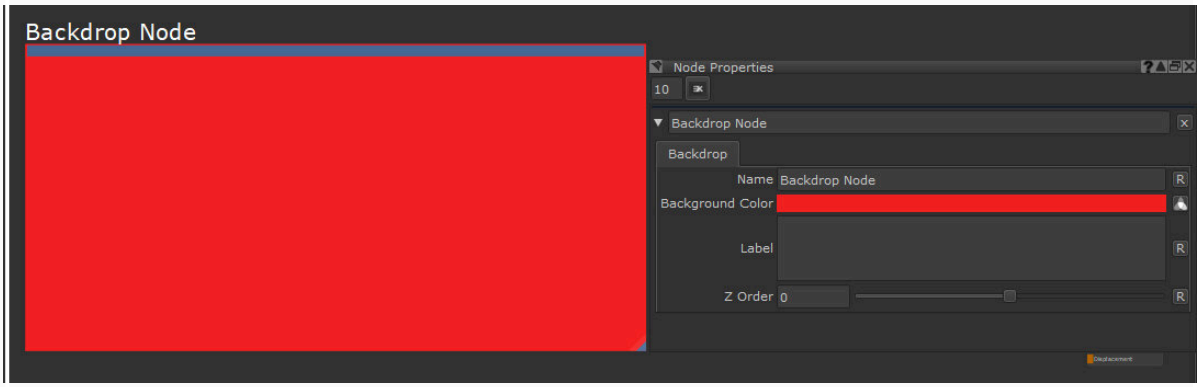
The name of the Backdrop is easily visible even when you zoom out, as it scales with the view. Since the Node Graph can get quite busy, this node is perfect for finding nodes quickly and organizing sections of your material creation to make them easily accessible.

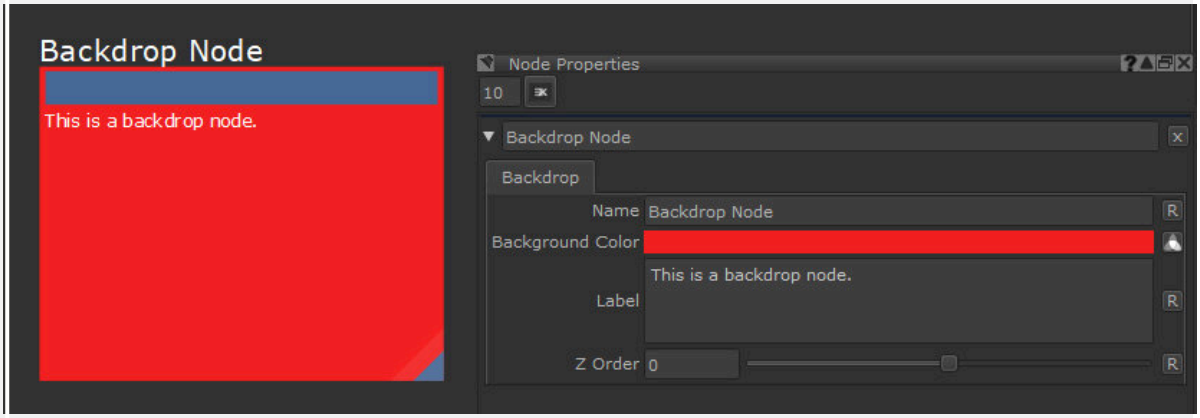
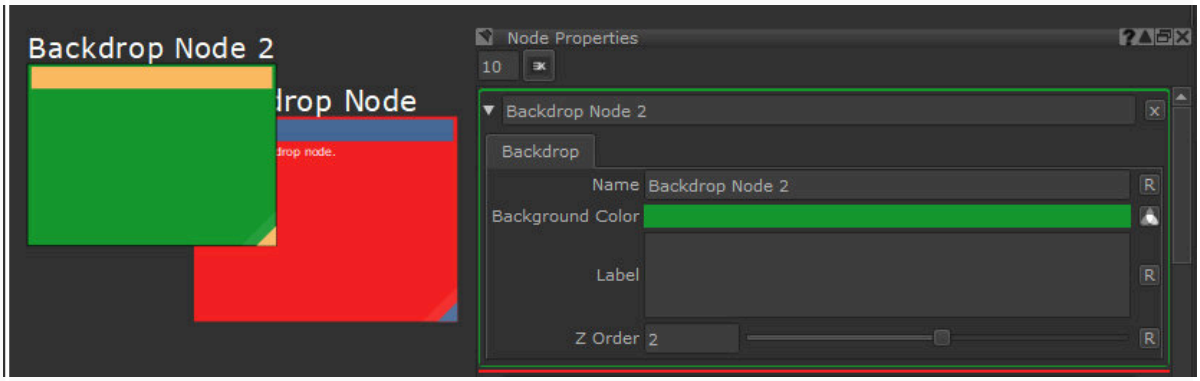
It can be helpful to use Backdrop nodes to group sections of your node tree. For example, all the masks you want to use in multiple places, or a section of your node tree that creates the Specular Roughness Channel.

Tip: To easily move a Backdrop node with all of its contained nodes, press and hold [**W**] and drag the whole Backdrop to a new location. [**Shift**] and [**Ctrl**] modifiers can be used to add or remove from the current selection.

Tip: Selecting a Backdrop while holding [**B**] will select the Backdrop but not any of its contained nodes. The shortcut can be customized in the Shortcut Manager.

Backdrop Node Properties

Name *text field*	Changes the name of the node. When zoomed in this **Name** is displayed in the top of the node. 
Background Color *text field, slider*	Changes the color of the Backdrop node background. If you make the **Background Color** a very light color the **Name** text color will change from white to black.  Default is a navy blue (0.09, 0.132, 0.19).
Label *text field*	Adds text along the top of the Backdrop. This text does not scale like the **Name** property. HTML code can be used for formatting.

	 Default is empty.
Z Order *slider*	When you have two Backdrop nodes overlapping each other, the highest **Z Order** number is shown on top of the lower one.  Default is 0.

Backdrop Node Workflow Example

Clean Scenes with the Backdrop Node

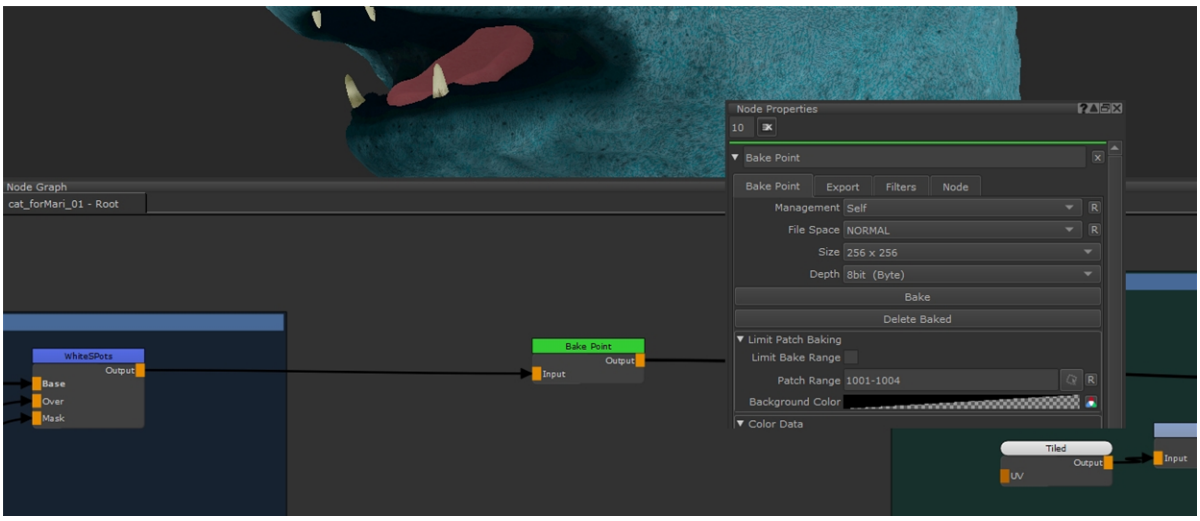
The Backdrop node is great for keeping a project from getting cluttered. When you start sharing nodes across your tree for the sake of optimization you get a lot of crossed pipes. But straight away you can see the Backdrop nodes when you zoom out, and because of that you know exactly where to go if you want to make a change to areas of the material creation.

Due to the nature of the Node Graph flowing from left to right, if you want to find the base color or tiling texture of a certain area, you know to look on the left side of a Backdrop node as they are going to have been set up at the start.

Clean scenes are important in case you ever need to hand your project over to someone else. You want them to be able to understand how to use your setup. Or maybe you come back to a project a few months after starting it, it's going to be a lot easier if everything is nicely organized and easy to find.

Bake Point Node

Access: **Nodes** > **Misc** > **Bake Point**



Using a Bake Point node to optimize textures in a scene.

The Bake Point node is a great way to optimize your scene and keep it running even when your node graph and project gets overly complex or heavy.

When you first create a Bake Point you will be asked to select the settings of the bake. The same options are displayed as when creating a Paint node. After selecting the size, bit depth and any other settings you may want to change, the Bake Point will be created. From there you can bake anything you input into the node to cache the input and keep your scene running smoother.

The Bake Point can be essential if your machine is starting to run slow, and can also speed up your workflow if you know you don't need to change any nodes that come before it.

Tip: Geo-Channels can be used as an intermediate caching point with the Bake Point node to allow complex node sharing without the difficulty of navigating complex shared connections in the Node Graph, by exporting the current bake data to a Geo-Channel on the object. See Adding Geo-Channels to Objects and Syncing Bake Points to Geo-Channel Nodes.

Bake Point Node Inputs

Input: Bakes the input.

Bake Point Node Properties

Note: For more information on the **Add Bake Point** dialog, see Add Paint/Bake Point Dialog.

Bake Point Tab

Management *dropdown list*	This option decides where the nodes properties come from. The default of **Self** allows the user to change them, the other option **Channel** defines the properties from the channel the Bake Point is connected to.
File Space *dropdown list*	This defines how values are imported or exported. The default of **Normal** changes nothing. For **Vector**, the data is treated as vector data and remapped. **Vector_Y_Flip** does the same but inverts the Y component. **Normal_Y_Flip** changes nothing but inverts the Y component.
Size *dropdown list*	Changes the size of the bake. Changing this scales the image up or down.
Depth *Dropdown list*	Changes the bit depth of the bake.

	Warning: Changing this converts the image and may result in lost texture information.
Bake *button*	Starts the baking process of the input into the node. **Note:** You can also bake Bake Point nodes directly in the Node Graph, right-click and select **Edit** > **Bake Points** > **Smart Bake And Update** or **Bake Selected**. See the *Node Graph Functions* section in the *Mari Reference Guide*.
Delete Baked *button*	Clicking this button clears any information held in the Bake Point node.

Limit Patch Baking

Limit Bake Range *checkbox*	Enable to only bake paint data within the specified **Patch Range** with this Bake Point node, allowing you to reduce bake times by only baking required patches. Once enabled, the Bake Point node displays an icon to indicate the bake range is limited to certain patches. Disable to bake all patches on your object with this Bake Point node.

Patch Range
 	
text field
	Specify patches on your object to be baked by this Bake Point node by typing either a single patch number, or a range of patches.
	
	For example, if you want this Bake Point node to only bake patches 1001 to 1005, enter:
	`1001-1005`
	
	If you want this Bake Point node to only bake patches 1008, enter:
	`1008`
	
	If you want this Bake Point node to bake patches 1003 to 1006, and patch 1009, enter:
	`1003-1006 1009`

  patch selection button	Click this button to view the following patch selection options: • **Set to Current Patch Selection** - The currently selected patch(es) is assigned to the **Patch Range**.

	• **Set to Patch Selection Group** - Opens a dialog allowing you to select a patch selection group from a dropdown. Once chosen, the selection group is added to the **Patch Range**. **Tip:** See the *Saving Selections on the Model* section in the *Mari User Guide* for more information about creating Patch selection groups. • **Select Patches in Range** - Automatically selects all patches specified in **Patch Range**.
R *reset button*	Click to reset the **Patch Range**. This resets to select all available patches on the object.
Background Color *color swatch*	Specify a background color onto which the baked image is composited. **Note:** Patches outside the specified **Patch Range** have a uniform, transparent background and are not affected by the **Background Color**.

Color Data

Colorspace *dropdown list*	If the **Raw Data** checkbox is disabled then the color space of the data stored in the bake is converted to the specified **Colorspace**.

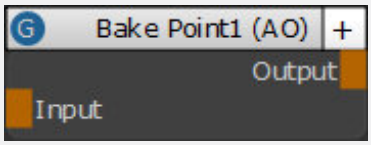
Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.

Raw Data *checkbox*	Specify if the data is raw or is in a different colorspace.

	Note: Disabling this allows you to specify the colorspace in the Bake Point, resulting in a conversion to the bake.
Scalar Data *checkbox*	Enable **Scalar Data** to indicate that pixel data is not affected by color management. Enabling this changes the view transform to **None**.

Export Tab

Geo-Channel

Geo-Channel *dropdown*	Select which Geo-Channel to use with the Bake Point node. Enter a new Geo-Channel name to create and export the baked data to a new Geo-Channel.  The Bake Point node updates to display the Geo-Channel name in the title bar once a Geo-Channel is selected.
Sync on Bake *checkbox*	Enabling this automatically syncs the latest bake data to the selected **Geo-Channel**.
Export Baked to Geo-Channel *button*	Exports the latest bake data on the Bake Point node to the selected **Geo-Channel**.

Disk

Export on Bake	Enable to automatically export images every time a bake occurs.

checkbox	**Warning:** Enabling this option overwrites previously exported images on disk each time a bake is triggered.
Export Path *text field / button*	The path where bake data from the Bake Point node is exported. See Exporting Textures from Layers for more information on the **Set Export Options** dialog.
Post Process	A post process command which triggers once the bake point image(s) has been exported. For example, you can enter a post process command which moves your images to a different folder once they have been written to disk, or a command to convert the image(s) to Renderman, Arnold, or 3Delight texture map files. Enter the following command to move the written images to a different path: `mv $EXPORTED "[drive:][path]"` **Note:** Hover over the **Post Process** field to see a tooltip with the accepted command attributes. They are the same attributes used by the **Export Manager**. See Export Manager Dialog.
Export Baked *button*	Clicking this button brings up an **Export UI** window to let you export the baked image to disk. **Note:** An overwrite check is performed if there are already images on disk using the same name.

Filters Tab

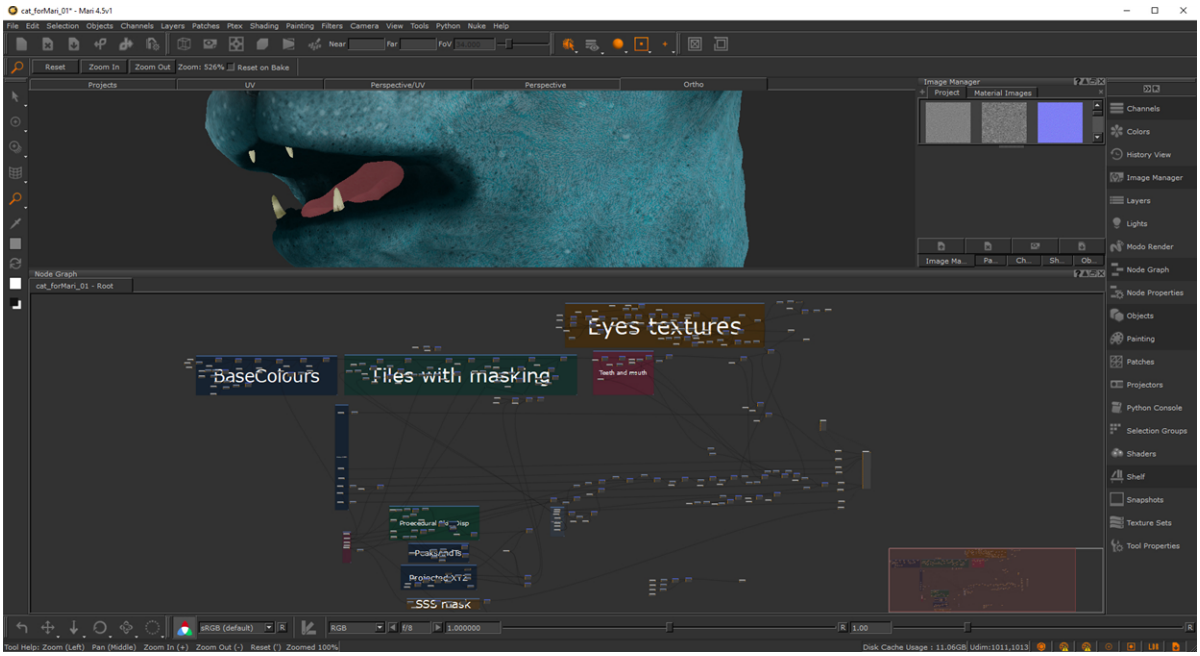
Filter Baked Patches *checkbox*	Enables or disables the visibility of filters applied to the Bake Point node.
Edit Filters	Click to open the **Apply Bake Point Filters to Bake Point** dialog. Select a filter to apply to the Bake Point node or preview the result of the filter.

button	
Summary *read-only text field*	Displays a list summary of all filters applied to the Bake Point node.

Bake Point Node Workflow Examples

Node Tree Optimization with the Bake Point Node

In the example we have a heavy node graph to texture this asset. Even on a powerful computer this scene is heavy to compute. If you have a busy scene or computer that struggles to keep up, Bake Points can be great for optimizing your node tree. In the example, we are seeing the Node Graph and a nearly finished **Base Color** texture:



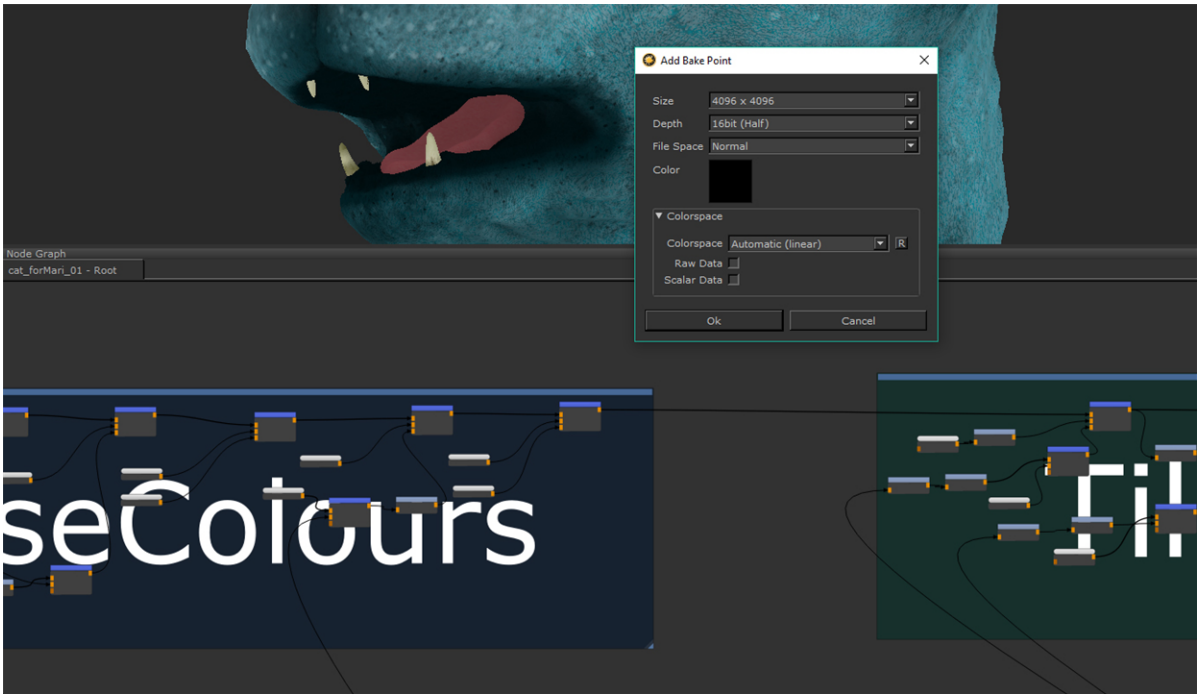
A nearly finished asset's Node Graph.

Let's use some Bake Nodes to optimize the **BaseColours** section on the left of this node tree. When you are finished with a part of your node tree, it can be good practice to use a Bake Point node to stop Mari calculating all the nodes that come before it. By doing this you can speed up your scenes.

> **Note:** The downside to this workflow however is you need to either deactivate the node if you want to make changes or rebake it after.

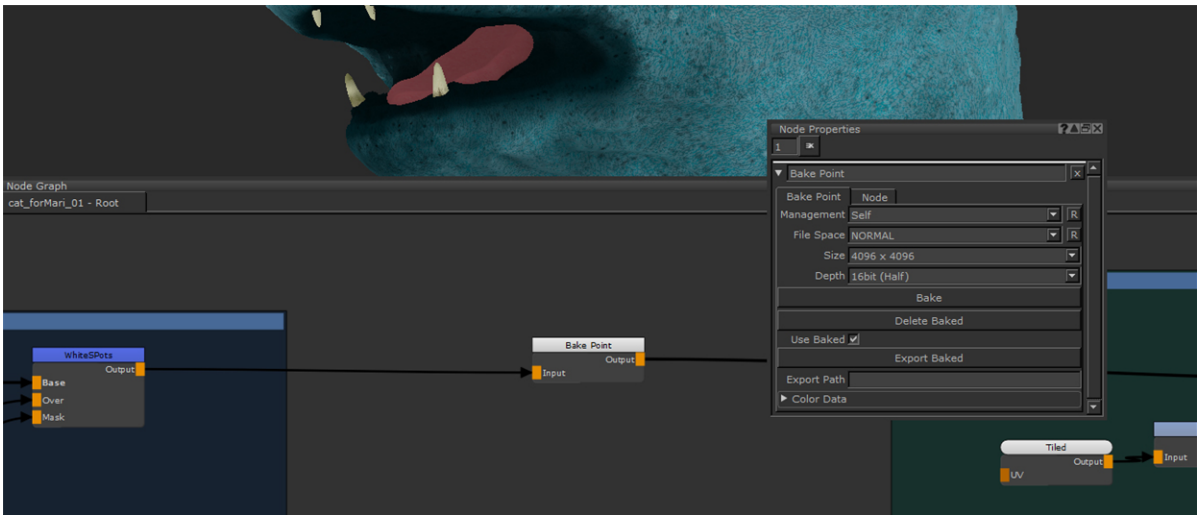
Since the base colors of this object are working and we're just finessing the smaller details which come after it in the Node Tree then a Bake Point can be placed to the right of this Backdrop node.

By placing down a Bake Point we will get the creation UI pop up. Since the UDIMs of this texture are 4k then we want to use that size but if you want to just use it for previewing or speed up your project even more you could select a size smaller than your patches. The rest of the settings will be dependant on your project and also the Channel this node will end up being used for. Since we have color data here we don't want to select Scalar Data on.



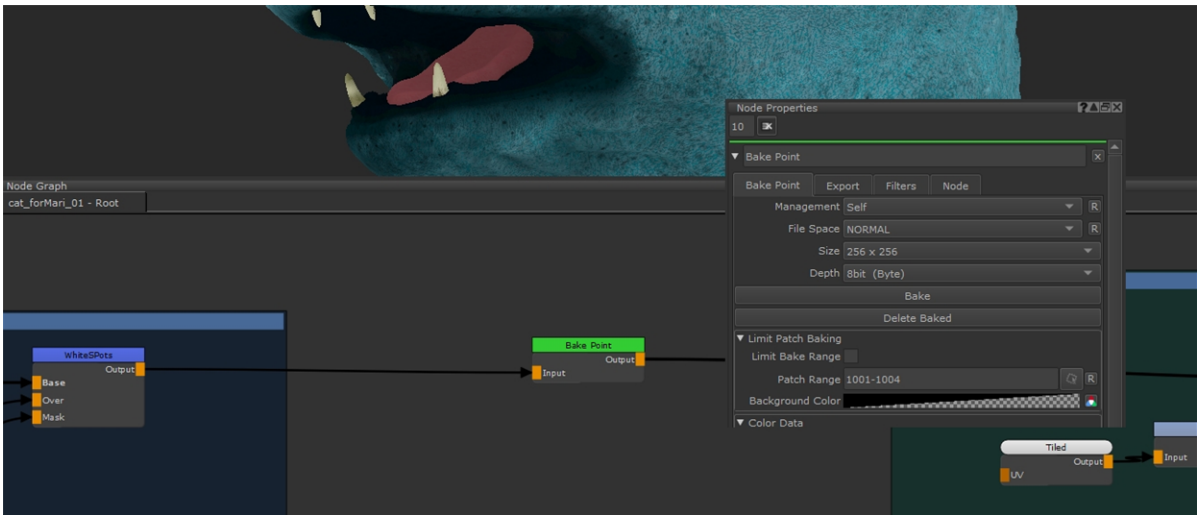
The Bake Point node **Add Bake Point** popup appears when first creating a Bake Point node.

After clicking OK the node will be created. It doesn't bake until you plug something in and select the **Bake** button.



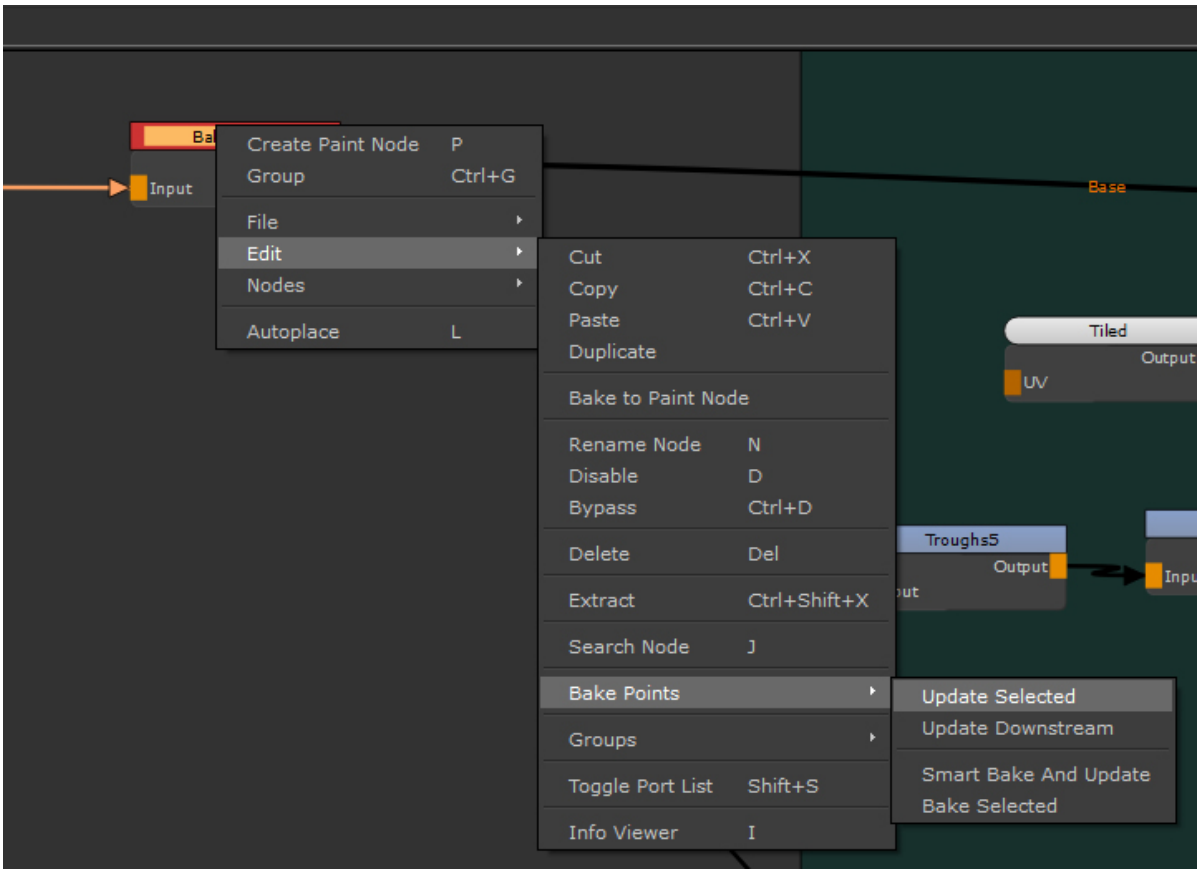
An unbaked Bake Point node appears gray on the Node Graph.

After you select an input for the Bake Point, Mari will perform a calculation, and depending on your scene this could take a while. When the calculation is done, the node will turn green to signify it has bake data inside of it that is up to date.



Once baked and up to date the Bake Point turns green.

If it has gone red that means something being plugged into it is out of date. You can either select it and click **Bake** again to update it, or you can shift and click multiple Bake Point nodes, right click and go to **Edit** > **Bake Points** > **Update Selected** to update multiple bakes at once.



When out of date the node turns red. You can update selected Bake Points by going through these menus.

When using Bake Points along your node tree you should notice your scene speeding up, especially when using a lot of nodes. Bake Points are a great tool for scene optimization.

Tip: Geo-Channels can be used as an intermediate caching point with the Bake Point node to allow complex node sharing without the difficulty of navigating complex shared connections in the Node Graph, by exporting the current bake data to a Geo-Channel on the object. See Adding Geo-Channels to Objects and Syncing Bake Points to Geo-Channel Nodes.

Limiting Patches in a Bake Point Node

You can reduce bake time in a Bake Point node by limiting the amount of patches/UDIMs being baked by that node to only patches that require a bake.

To limit patches/UDIMs in a Bake Point node:

1. Double-click the Bake Point node to access the **Node Properties**.
2. Enable **Limit Bake Range** in the **Bake Point** tab.
3. Using one of the following methods, select a **Patch Range**:
   - Enter a value into the **Patch Range** field. The following example selects patches 1001, 1002, and 1003:

     ```
     1001-1003
     ```

   - If you have a patch already selected, click the patch selection button and select **Set to Current Patch Selection**.
   - If you have set up patch selection groups, click the patch selection button and select **Set to Patch Selection Group**.

   Tip: If you're not sure which patches are currently selected, click **Select Patches in Range**.

4. Pick a color using **Background Color** onto which the baked image is composited.
5. Click **Bake**.

   The selected patches in **Patch Range** are baked.

Filtering Non-Destructively Using Bake Point Filters

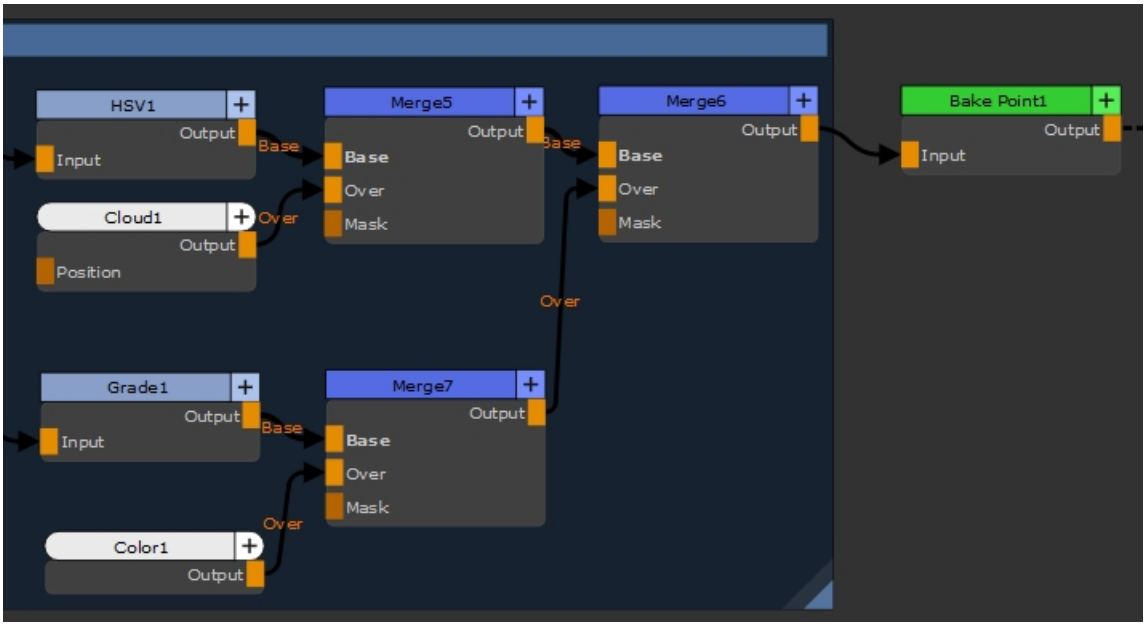
Bake Point Filters allow you to apply a stack of filter effects, such as **Gaussian** blur or **Emboss**, to a node without baking your network to a single Paint node. In older versions of Mari, you would bake down your node graph to a Paint node to apply filters, which is a destructive workflow that prevents any upstream changes being made to procedural nodes or filters.

Bake Point filters remove the need to bake parts of your node network by applying filters to a Bake Point node cache, rather than directly to the node or network of nodes you're working with, so you retain the option to alter them later on. When an out-of-date Bake Point with filters enabled is updated, its filter stack is reapplied to the newly baked result automatically.

1. Add a Bake Point node down stream of your node network and connect the node to the network.
2. Double-click the Bake Point node to open its **Properties** panel.
3. Click **Bake** to process all the upstream nodes in the network and cache the output to disk.

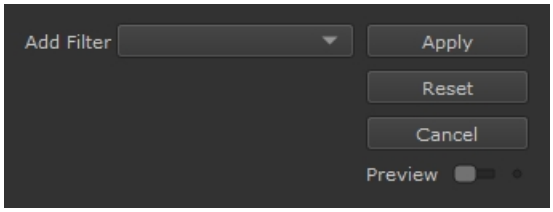
> **Note:** Bake Point filters apply to whatever patches you're using on your mesh, so if you want the filters to only apply to a particular UDIM tile of your asset then you can specify this in the **Limit Patch Baking** section of the node properties. See Limiting Patches in a Bake Point Node for more information.

The Bake Point node turns green when the bake is complete.



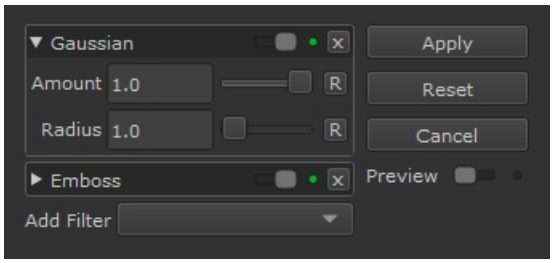
4. In the **Properties** panel, click the **Filters** tab, and then click **Edit Filters** to add a Bake Point filter.

   The **Apply Bake Point Filters** dialog is displayed.



5. Click **Add Filter** to display a list of available filters you can apply to the Bake Point node.

6. Select a filter to apply it to the Bake Point. In this example, a **Gaussian** blur and an **Emboss** filter are applied.

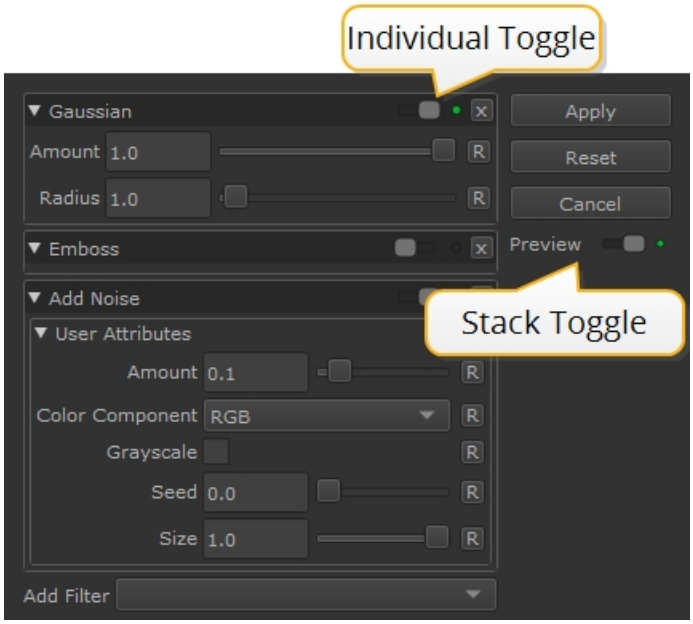


> **Note:** Some filters, such as **Emboss** and **Sharpen**, are snapshot-based preview filters that are applied to a snapshot of the canvas, rather than the output of the Bake Point, and may not look as expected. Applying these filters displays a warning if they are added to a filter stack containing non-preview filters such as **Gaussian**.

7. Use the controls associated with each filter to adjust the appearance of the filter. For example, you can increase the **Radius** of the blur applied by a **Gaussian** filter.

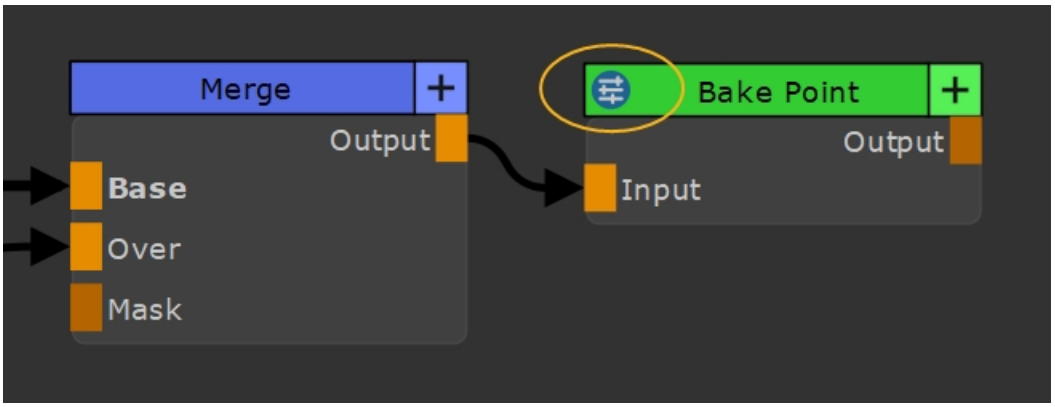
   Filters are applied from the top of the stack down, in the order that they are listed. In the example, this would mean that the **Emboss** filter is applied to the canvas after the blur is applied. If the order was reversed, the result would be very different because the embossing would occur before the blur.

8. You can enable and disable individual filters in a stack using the toggle adjacent to the filter name or enable and disable the entire stack using the **Preview** toggle.



9. Click **Apply** when you've finished editing your filter stack.

Bake Point nodes containing a filter stack display a stack icon so you can see at a glance where filters applied in the network.



10. You can incorporate changes in nodes upstream and downstream of Bake Point nodes by opening the Bake Point node's **Properties** and clicking Bake or by right-clicking a node in the network, selecting **Edit** > **Bake Points**, and then selecting the required action.

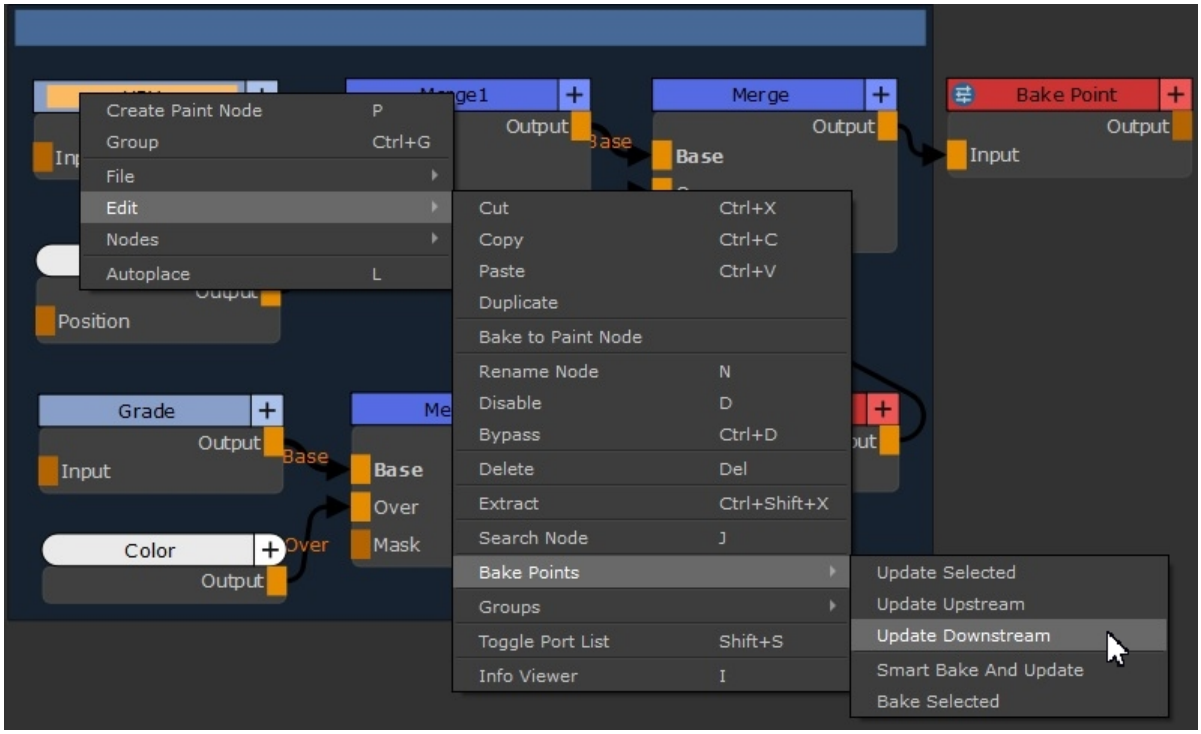


Image Input node

Access: **Nodes** > **Misc** > **Image Input**

A node that takes an image as a parameter and can be connected to nodes that need images as parameters/inputs, providing a more modular approach to usage of image nodes in material/smart masks. This is helpful for driving multiple image nodes from a single point.

For example, if you are creating a Material with multiple projection options (UV/Camera Projection/Tri-Planar Projection), you could use a single Image Input node to control the image being projected.

Image Input node outputs

• **Image Out** — Outputs the image token. This will only be visible when connected to Image inputs.

> **Note: Image Out** does not represent the sampled image, but an image token intended for passing to other nodes to sample. It is not intended to be viewed directly. Viewing, baking or connecting it into a non-image node input will not show expected results.

• **Preview** — A preview view of the image in UV space.

Image Input Node Properties

Image	*image picker*	The image you want to use to drive image nodes.

Material Node

Access: **Nodes** > **Misc** > **Material**

The start of the texturing process of a watch asset with 2 Material nodes and a single mask.

The Material node allows you to texture with a material based workflow. Instead of texturing your individual texture channels one at a time, you define an entire material all at once and then merge multiple together to quickly build up all your texture maps.

Not only does this speed up your workflow, but the ability to share, import and export materials helps productivity among multiple texture artists.

The Material node sits at the heart of Mari's material based workflow and is vital to creating and editing your own materials for texturing. It may be a new style of working inside of Mari so check out the examples to see how it works.

For some quick video tutorials of the new 4.5 Material System, check out Mari 4.5 Fundamentals video series here.

Material Node Inputs

The Material Node by default has no inputs, however when inside of the node you can use an Input node to allow textures and data from your Node Graph to be shared inside the Material node.

Material Node Properties

Property	Description
p *button*	Clicking the **p** button allows you to promote attributes and properties from the nodes you use inside the Material node.
s *button*	Clicking the **s** button opens the Material node sub-graph in the Node Graph.
x *button*	Clicking the **x** button closes the Material node properties.

Export Tab

Property	Description
Export Path *text field, button*	Specify a path to export your material **.mma** file to using this field. **Note:** You must specify an **Export Path** in order to export your material.
Thumbnail Path *text field, button*	Specify an image path to be used as the thumbnail image for this material **.mma** file.
Tags *button*	Add metadata tags to your material, if required. Start typing a description into the field to display a list of existing tags or create your own and press **Enter** to add the tag. Tags enable Mari to display information about an **.mma** file before you apply the material. For example, if the material represents a rusty metal surface, you could added descriptive tags to identify the material more easily.
Export As Material *button*	Exports the material as an **.mma** file to the file path specified in **Export Path**, using the thumbnail image specified in **Thumbnail Path**.

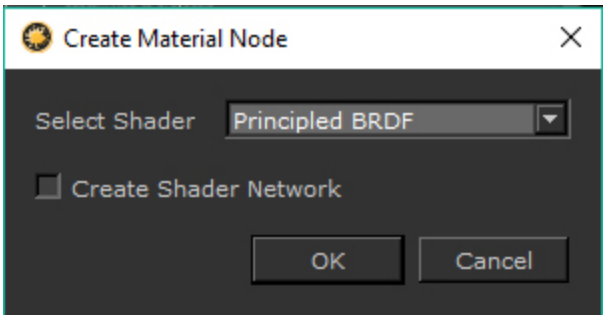
Material Node Workflow Example

Creating a Material in Mari

Lets create a quick wooden material that defines every channel you need.

First you need to put down a Material node. A dialog box appears on the UI requiring you to select a shading model. In this example **Principled BRDF** is selected, but if you want to use another then the principles remain the same.

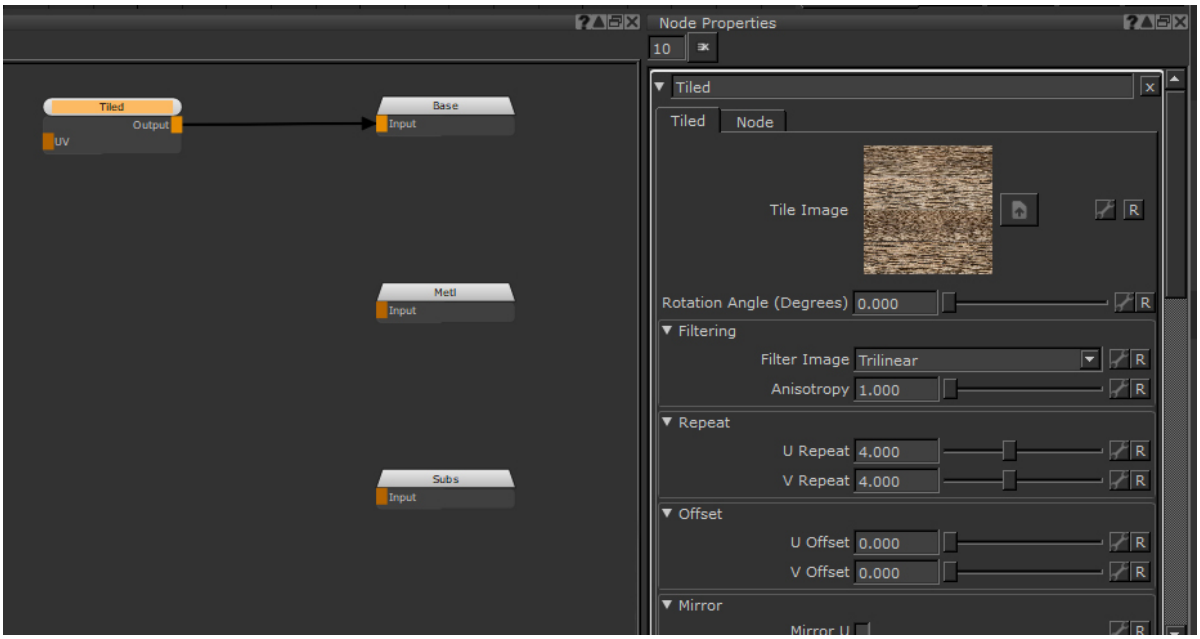
> **Note:** If you turn on **Create Shader Network** then a shader is also created that connects the material's outputs to the shader.



Then to edit it you need to **Ctrl** + double-click the Material node. This opens up a second tab in the Node Graph containing the Material node contents. To return to your main Node Graph just click the other tab.

Inside this new graph, you can see a number of Output nodes, all named after different texture channels. In here, you can assign textures and inputs to any of the Material node's outputs.

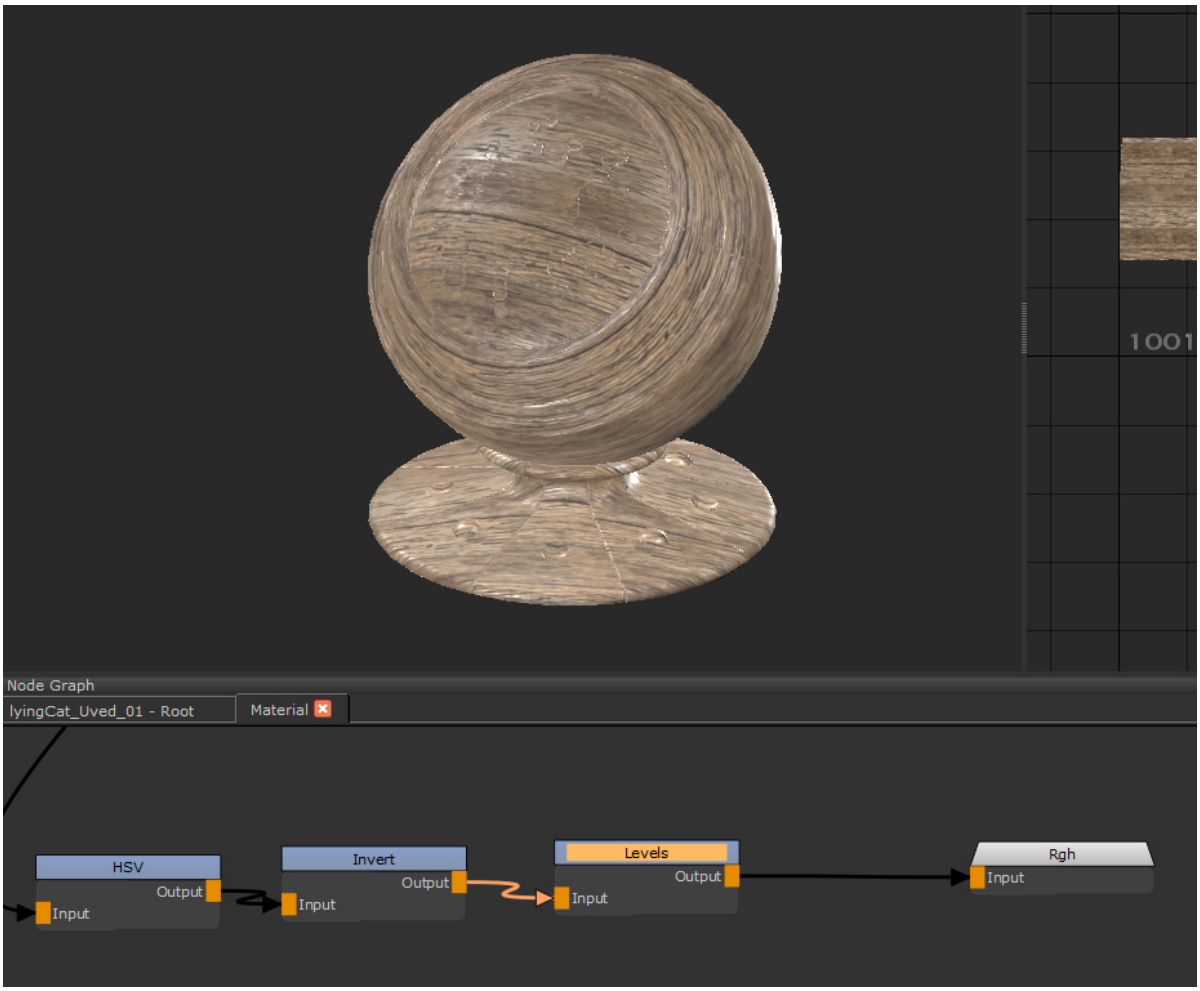
Next we need a tiled texture. You can create one by dragging an image from the **Image Manager** into the Node Graph. This creates a Tiled node containing your selected image. From there you can change your scale so it looks right for you and make any other edits to the Tiled node until you get something you are happy with. Then it's time to plug that into the **Base Color** output.



Note: If you are connecting a source node with shader model driven output ports to the Output nodes, you can connect just one stream of the source node's output ports to its corresponding Output node, and all sibling output ports automatically connect to the sibling stream Output nodes.
For more information, see Connecting Multi-Channel Nodes.

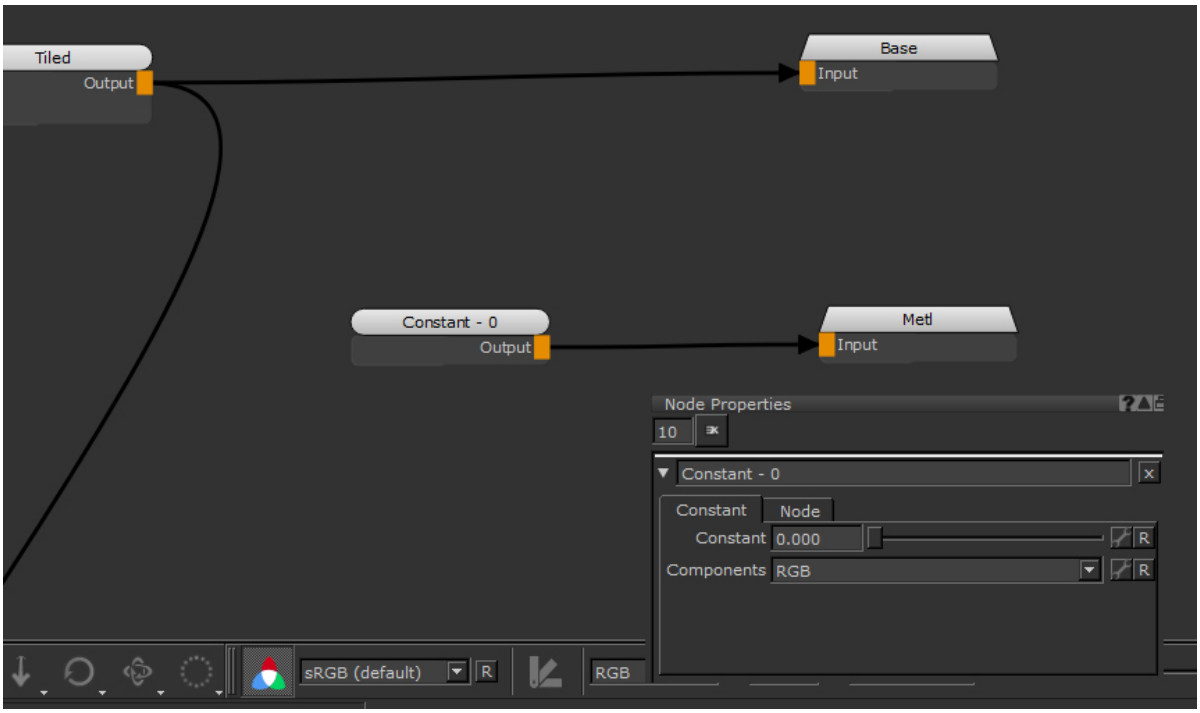
Next use an HSV node and an Invert node to de-saturate and flip the Tiled node to create a rough map that we can use for the **Specular Roughness**. You can use a Levels node to adjust the values until you have something that looks right. The easiest way to tell is by going back to your Node Graph and hooking the material up to a shader and viewing it so you can Lookdev values on the fly.

Tip: If you're unfamiliar with creating and customizing shaders, check out the example documentation for the **BRDF** and **Principled BRDF** nodes.

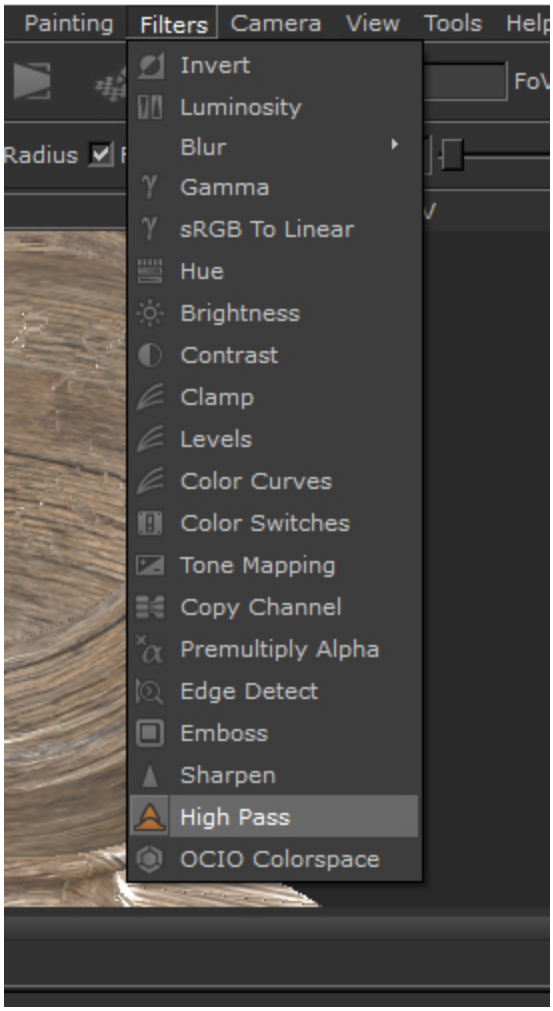


The **Base Color** and Roughness (**Rgh**) being viewed through a shader. Going back into the Material node and tweaking the **Levels** means you can Lookdev as you go.

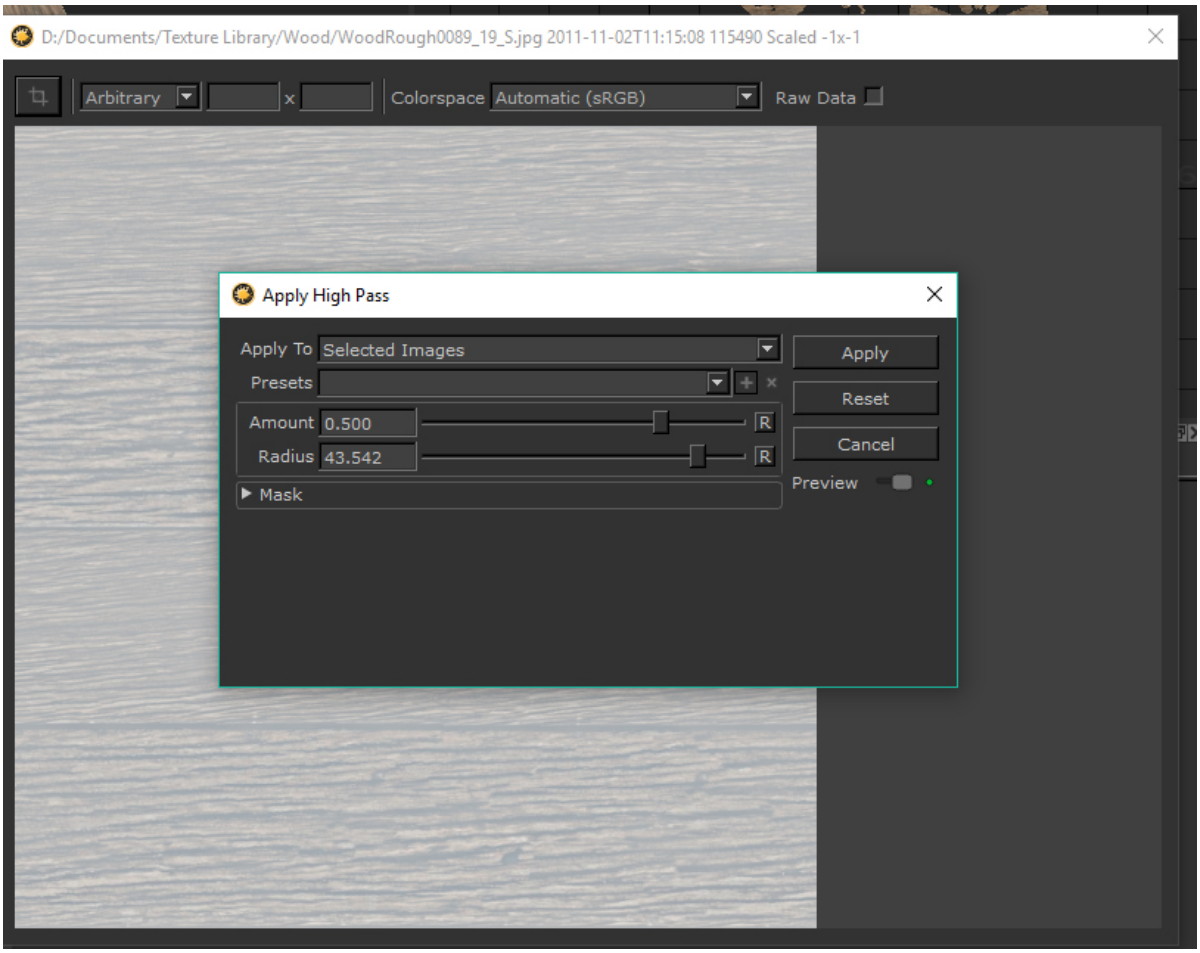
Adding a Constant node with a **Constant** value of **0** with **RGB** components selected can be used for the **Metallic** output of the material, as the wood for our example is not metallic in appearance.



To make a bump map you can use a **Filters** > **High Pass** on the wood texture in your **Image Manager** to get a quick height map from it.

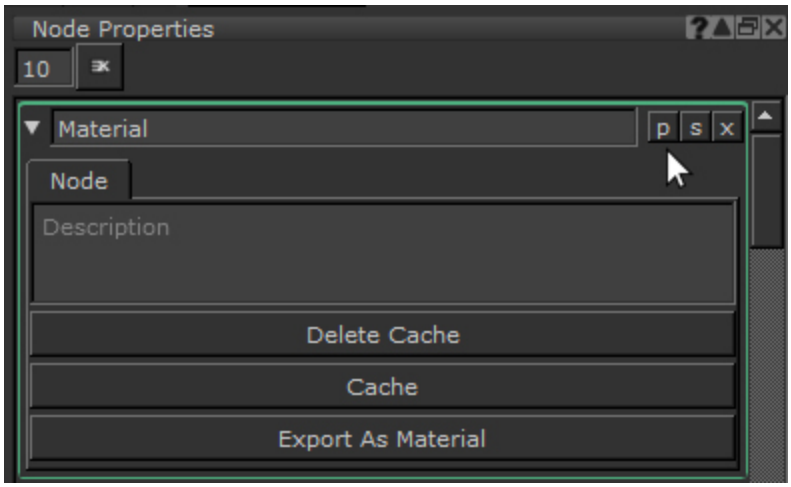


Make sure the **Amount** stays at **0.5** to get an image with mid-gray as the mid-point as this is what a bump map needs.



You can then use that with another Tiled node that plugs into the **Bump** output of the Material node. Copying and pasting the first Tiled node used is a good idea so the tiling amount lines up with the base color. In the example is the final result, a quickly built up material you can export and reuse in other scenes.

Finally lets promote the Tiled node properties so they can be edited on the fly without having to jump in and out of the node every time. If you jump back to your Node Graph and click the **p** button on the **Node Properties** a **Group Node Knobs** dialog box appears.

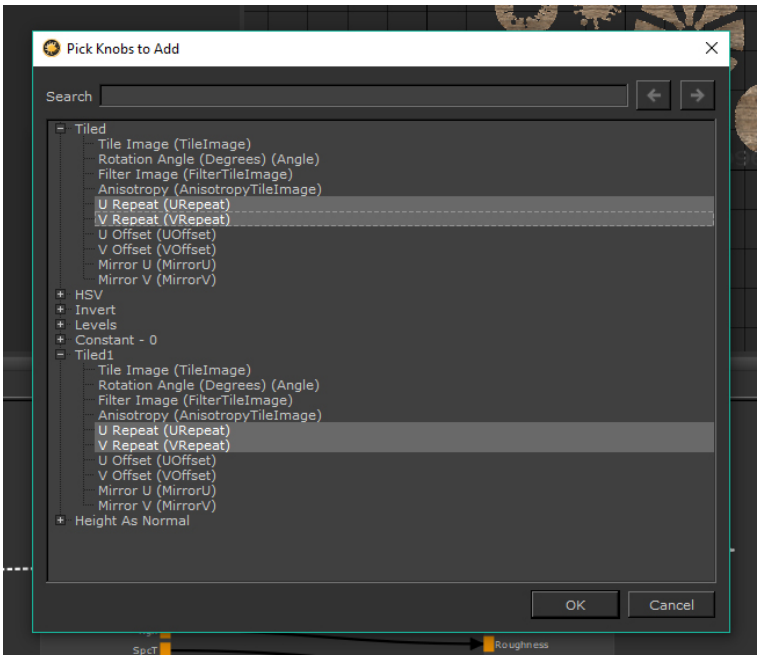


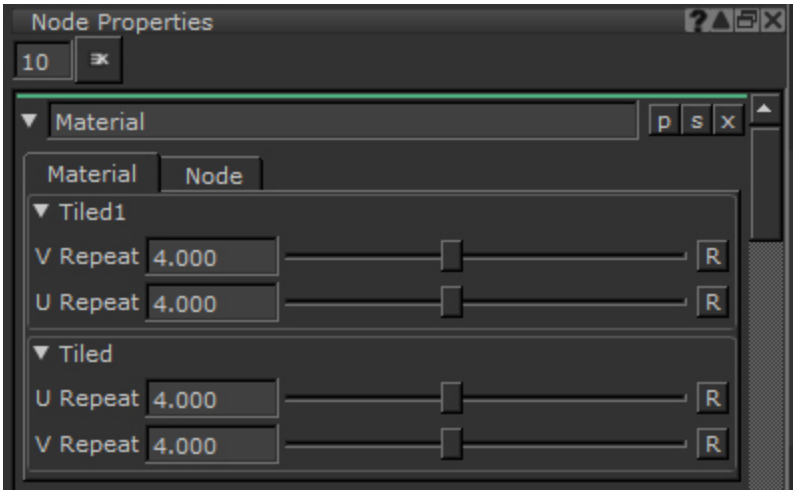
From here you need to choose which properties to promote.

Promoting Node Properties

1. Click the **Pick** option and open up the two Tiled nodes with the plus icon next to their names.
2. Click to select the **U Repeat** and **V Repeat** of both Tiled nodes and click **OK**.

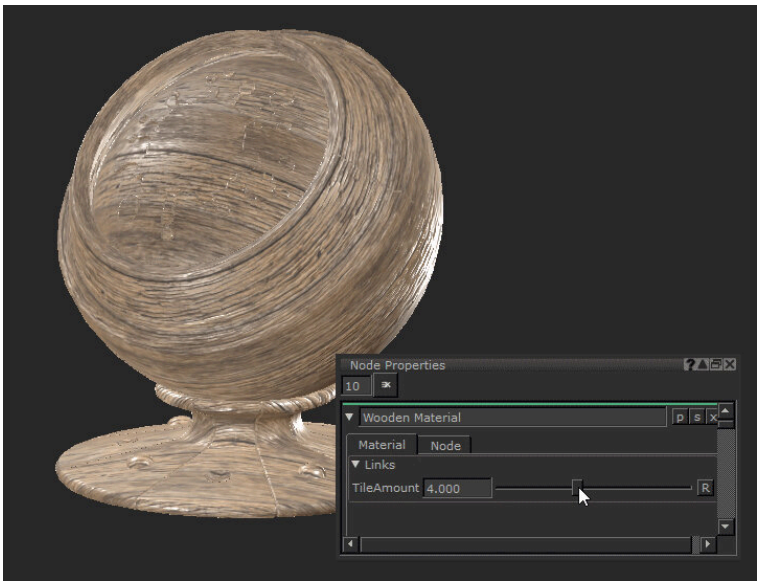
   They are promoted to the Material node. Now they can be edited on the go without going in and out of the Material node.





The promoted **U Repeat**, and **V Repeat** properties on the Material node.

If you select those promoted properties in the **Pick Knobs to Add** again, and click the **Link** button in the same dialog you can merge them all into a single slider of your choosing. Name it something obvious, like 'TileAmount', and you can now see the new linked attribute in the **Node Properties**.

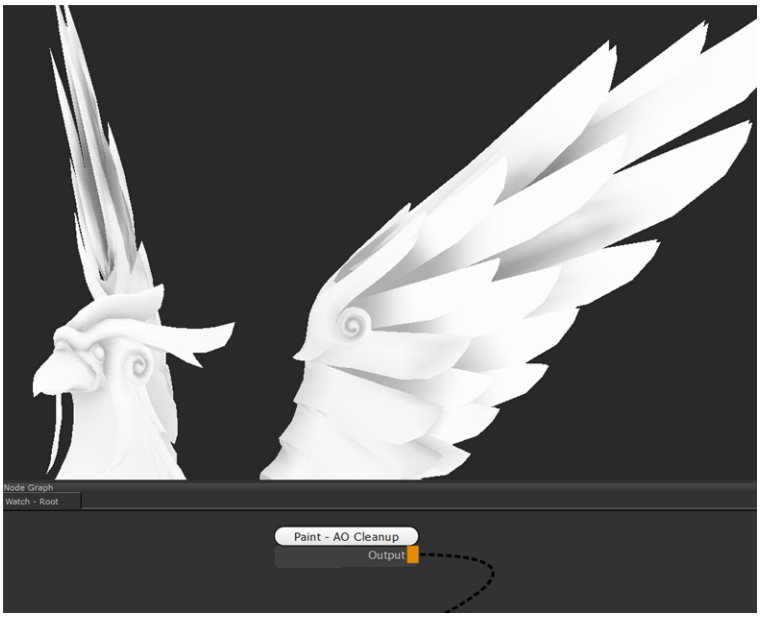


That is a way to quickly build up some of the output channels of a Material node from a single tiled texture. You can use this as a base and tweak things or add additional details with other procedural nodes like noises to add detail and complexity.

When you have multiple materials in a scene you can use the Multi-Channel Merge node to combine them together in a similar way to the Merge node.

Paint Node

Access: **Nodes** > **Misc** > **Paint**



A cleaned up Ambient Occlusion map, using the **Bake to Paint node** feature.

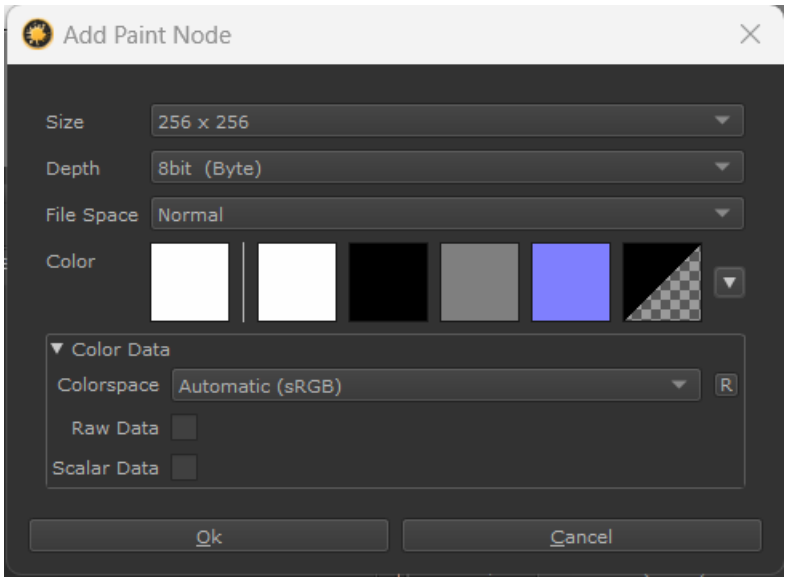
The Paint node allows you to paint, import, or project your own textures into it. When creating textures, depending on your workflow, it can be one of the most commonly used nodes.

When creating textures, if you want to paint a mask in a specific area, or project some reference onto your mesh you can use the Paint node to store that data. You can then use other nodes to manipulate it and add to it.

Since Paint nodes are stored data they can vary in size and bit depth to help optimize your scene, which is a big advantage nodes have over using a layer based workflow. If you need a rough, non-detailed mask, then you can make a low resolution paint node to fill areas of your mesh. If you need high fidelity, hand-painted break-up, you can create a high resolution Paint node to store it in.

Tip: If you need to import images and textures from software other than Mari, then you need a Paint node to do so.

Paint Node Properties



The **Add Paint Node** dialog.

When you first create a Paint node, you will be greeted with this UI dialog. This defines the settings for the Paint node that is created.

You can select the leftmost Color square to spawn color picker, or use any of the preset colors (white, black, mid grey, normal, transparent alpha).

> **Tip:** You can also store more colors in the Color Presets shelf, and then access them via the button here.

The settings on this dialog can be changed from the **Node Properties** after creation. Below is a list of all the properties that can be found there.

Management *Dropdown list*	This option decides where the node's properties come from. The default of **Self** allows the user to change them. The other option **Channel** defines them from the channel the Bake Point is connected to.
File Space *Dropdown list*	This defines how values are exported. The default of **Normal** changes nothing. For **Vector**, the data is treated as vector data and remapped. **Vector_Y_Flip** does the

	same but inverts the Y component. **Normal_Y_Flip** changes nothing but inverts the Y component.
Size *Dropdown list*	Chooses the resolution of the images when exported.
Depth *Dropdown list*	Chooses the bit depth of the images that get exported.

Color Data

Colorspace *Dropdown list*	If the **Raw Data** checkbox is disabled then you can define color space of the color data of the channel using this property. By default, Mari selects the colorspace determined by the value set in **Default Import Colorspace** in the **Mari Preferences** dialog under the **Color** tab. See Mari Preferences Dialog for more information.

Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.

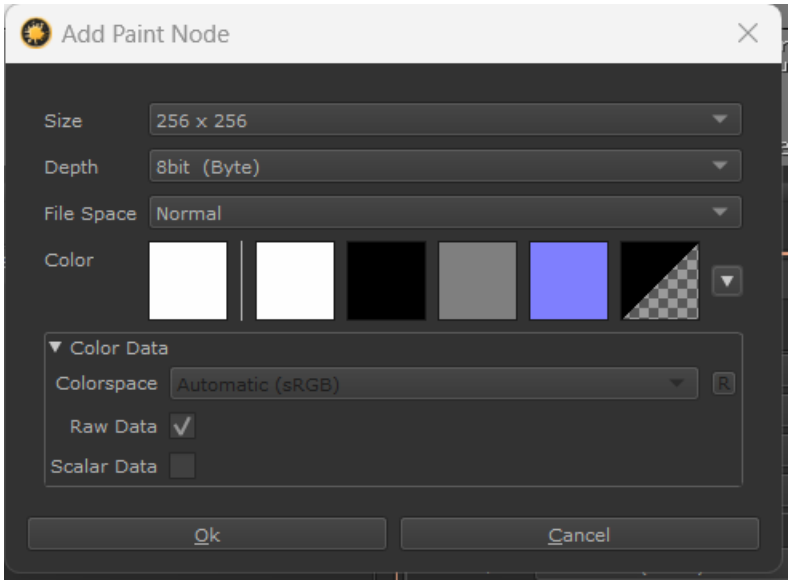
Raw Data *Checkbox*	Determines if the data is raw or is in a different color space. Disabling this will allow you to change the **Colorspace** dropdown, potentially resulting in a conversion when displayed and exported.
Scalar Data *Checkbox*	Determines if values going through the channel are **Color** or **Scalar Data**. If you enable this the view transform is changed to **None**.

Paint Node Workflow Example

Importing External Images into a Paint Node

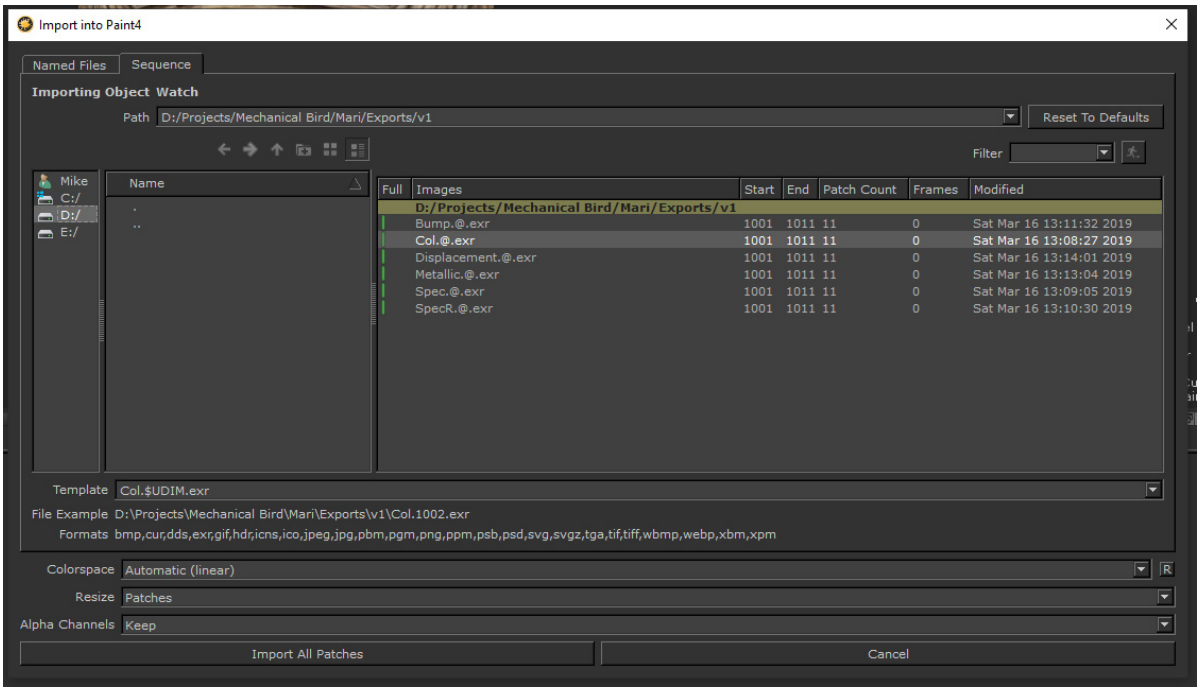
The Paint node is how you can import images into your UDIM patches, for example images that have been created in external software. In this example images that have been edited in Nuke are going to be reimported into Mari.

First you need to create a Paint node that matches up to the **Size** of your images and has the same **Bit Depth**. You can also disable the **Raw Data** checkbox if you would like to override the **Colorspace**.



The **Add Paint Node** dialog.

Once the node has been created you need to right click the Paint node, and select **File** > **Import**. This brings up a file browser where you can select the images you would like to import.



The **Import into Paint** node dialog.

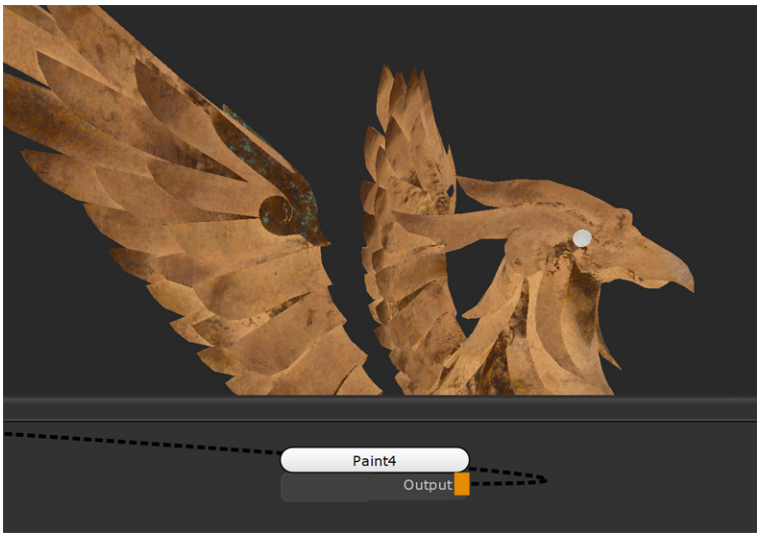
Here you need to go to the location of the files and select the image or image sequence you wish to import.

> **Tip:** If for some reason you are in the right folder but the files do not appear try clearing the **Template**.

From here you can override the **Colorspace** of the images to be imported. By default, Mari selects the colorspace determined by the value set in **Default Import Colorspace** in the **Mari Preferences** dialog under the **Color** tab. See Mari Preferences Dialog for more information.

You can determine if Mari should resize the images or patches if the import and Paint node don't match up. You can also determine what to do with the imported image's **Alpha Channels**.

When you are done with that you can click **Import All Patches** to process them. If you used the selection tool to select only certain patches you are also be given the option to **Import Selected Patches**.

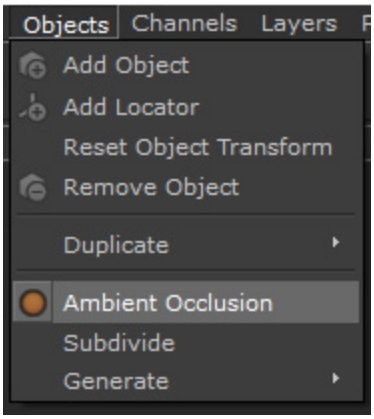


After that's done, you have a Paint node containing your images, which you use in your texturing.

Baking AO to a Paint Node to Clean Up

Being able to bake a procedural node, such as a noise, tiled image or color to a Paint node is a handy feature of Mari and makes it possible to then paint on top of a procedural.

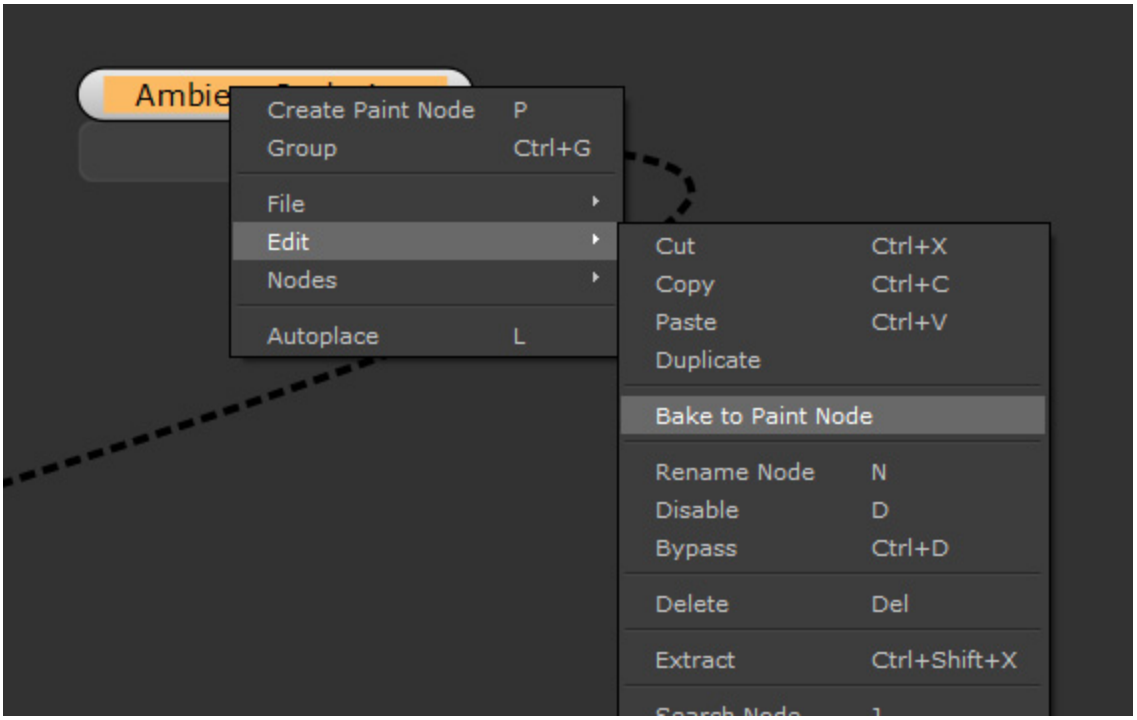
Tip: To create Ambient Occlusion on your model, first go to the **Objects** menu then click on the **Ambient Occlusion** option to start baking it for your model.



You can now see Ambient Occlusion when you create the node.

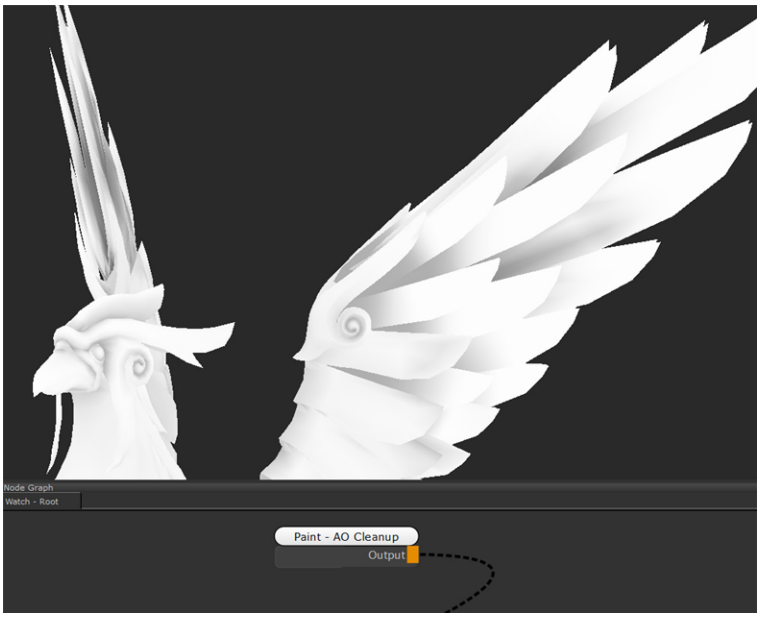
The Ambient Occlusion node gives good results but depending on the resolution of the mesh it can give blocky results as seen here on this geometry's wings. After baking it down to a Paint node, you can use the **Paint** tool to clean it up and add extra Ambient Occlusion to places you feel need it.

After creating an Ambient Occlusion node, or any procedural, you can right click the node and go to **Edit** > **Bake to Paint Node**. This pops up the **Add Image Set** dialog so you can choose the **Size** and **Bit Depth**. After that Mari will calculate and create a separate Paint node with that information baked into.



Baking to a Paint node.

From there, you can use any of the painting tools to add extra detail to it. Since this is an Ambient Occlusion, it needs black and white colours. Using the paint brush tool with the **p** key, you can clean up and smooth out the blocky areas of the procedural version.

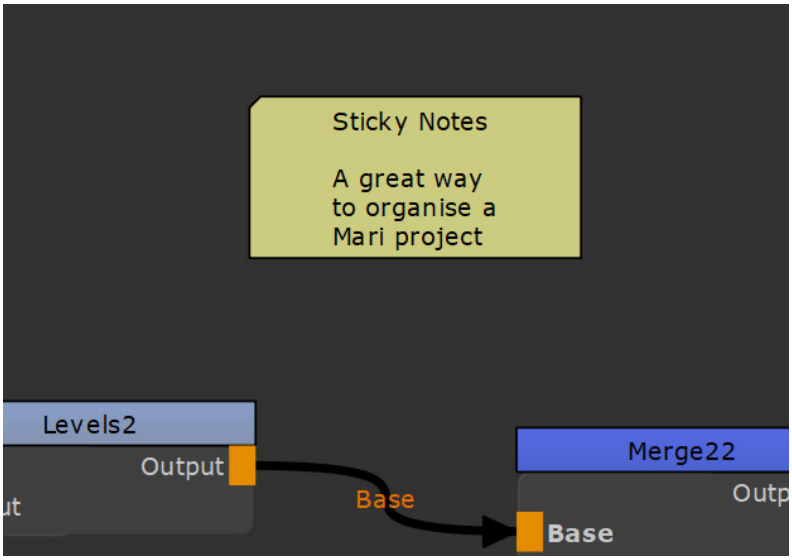


Since that data is now in a baked down node, you are free to paint and clean up parts of the texture

> **Tip:** Since baking a node to a Paint node creates a copy it's always good to rename it afterward so you know what the Paint node is for!

Sticky Note Node

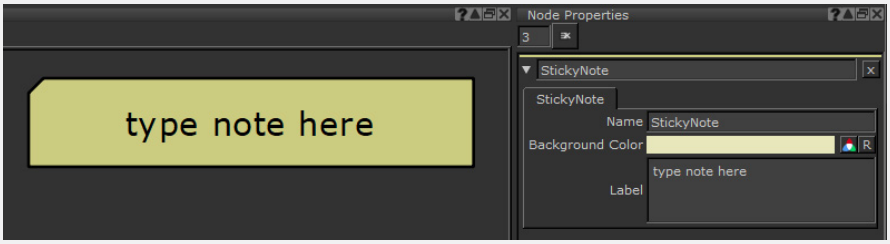
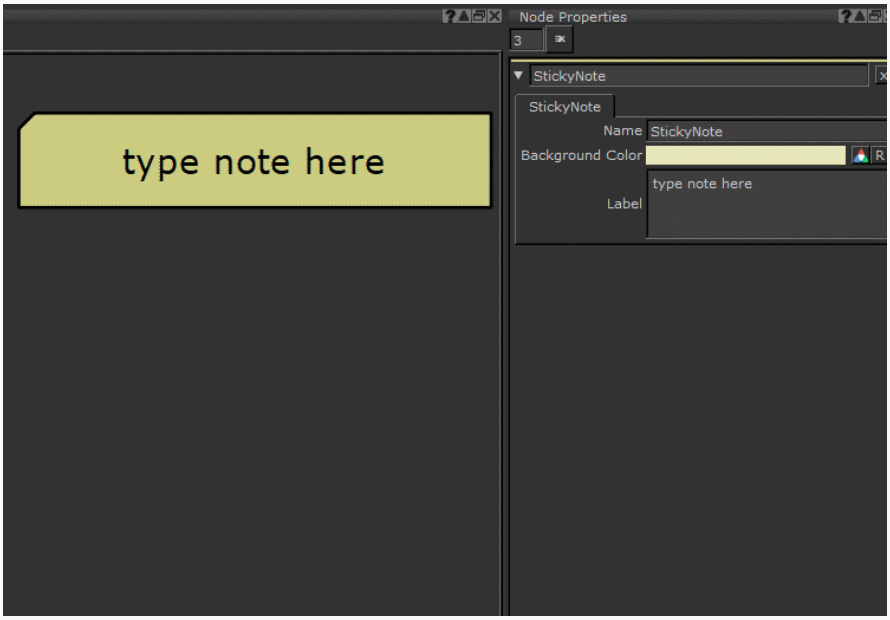
Access: **Nodes** > **Misc** > **Sticky Note**

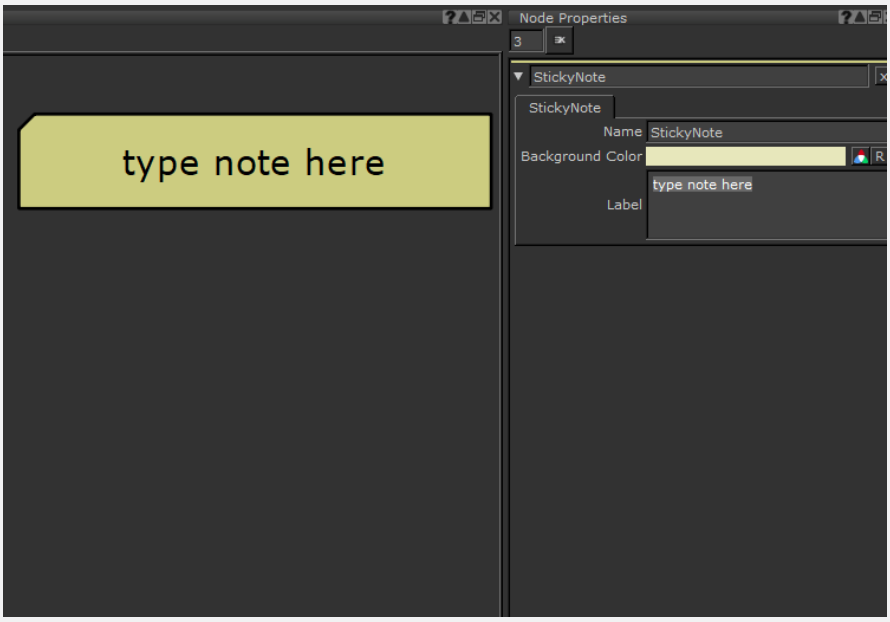


The Sticky Note is a great way to organise your project and leave memos inside your Node Graph. It works differently to other nodes when you create one, by dropping one of these down you create a coloured rectangle which you can type text into.

Since the Node Graph can get quite busy and cluttered this node is perfect for organizing sections of your texture creation or to leave comments for yourself and others to help understand how or why you have constructed parts of your node tree. The Sticky Note node is a great way to leave comments and notes for your self and other people that may pick up your Mari projects.

Sticky Note Node Properties

Name *text field*	Changes the name of the node. This does not display on the node but can be useful with the **Search Nodes** function (**Edit** > **Search Nodes**). 
Background Color *text field, slider*	Changes the color of the Sticky Note node's background. If you make the background a very light color the **Name** text color changes from white to black. 

	Default is a pale yellow (0.8, 0.8, 0.5).
Label *text field*	Adds text along the top of the Sticky Note. This text does not scale with the viewer. It scrolls infinitely along one line, use a line break with the **Enter** key to add a new line.  Default is **type note here**.

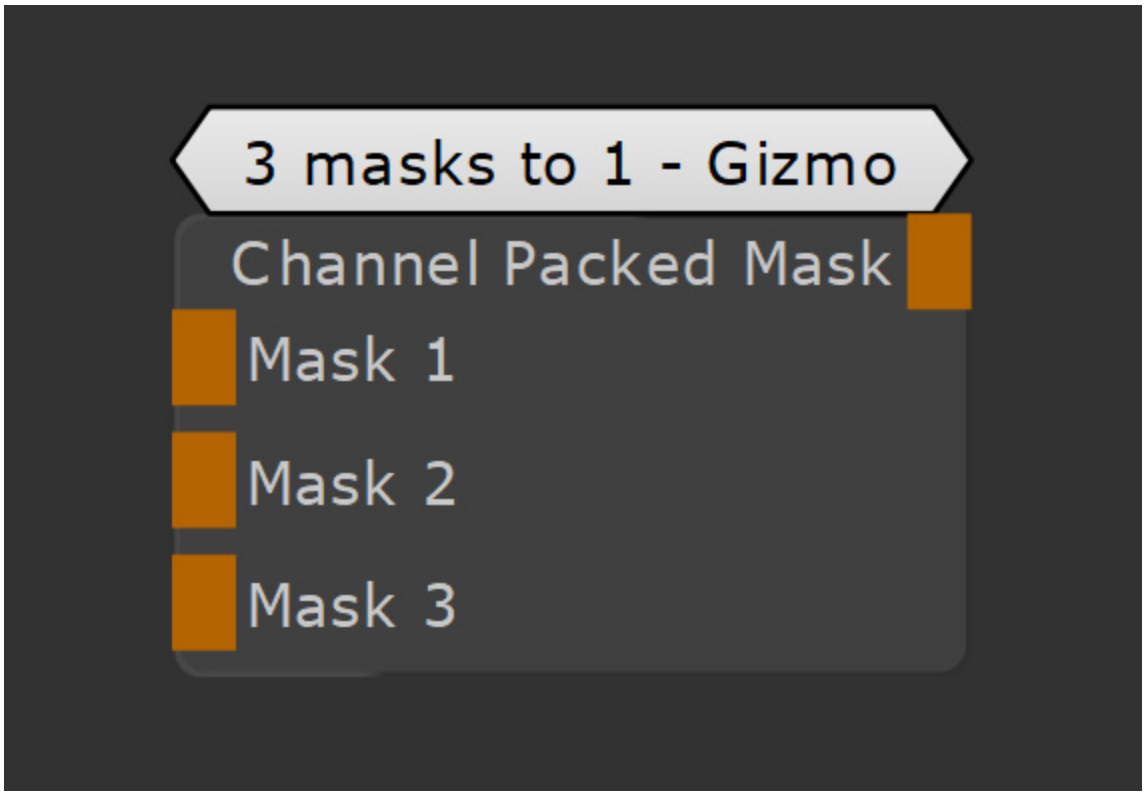
Sticky Note Node Workflow Example

Annotating a Group with the Sticky Note Node

In the example we have a custom gizmo inside a Group node that turns three black and white channels into a single channel packed node and outputs it.

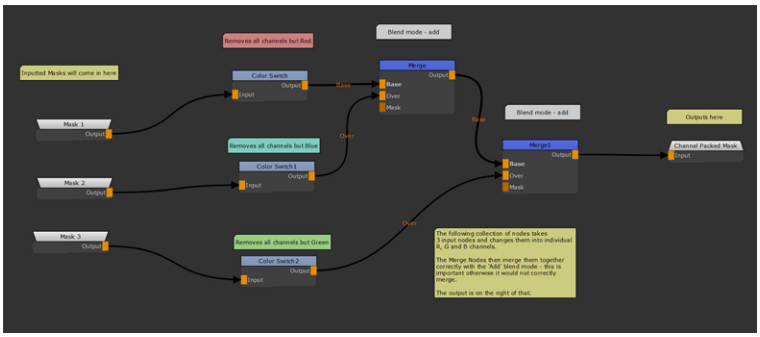
> **Tip:** For more on the technical creation of the channel packing process see the documentation for the Color Switch Node.

When you jump into someone's groups, gizmos, or projects it can be a bit overwhelming what is going on, so Sticky Note nodes are a great way to annotate your Node Graph.



If this was passed to someone and they jumped inside of the group they may not understand how it was created. This is a fairly simple example, but the more complex a project or group gets the more it helps to label things correctly.

By using the Sticky Note node we can explain to users that jump inside of the Group node how it has been created and how to edit it if they wish.



Now if I export this Gizmo and give it to anyone else, even if they're a beginner Mari user it will all make sense to them.

Teleport Broadcaster and Teleport Receiver Nodes

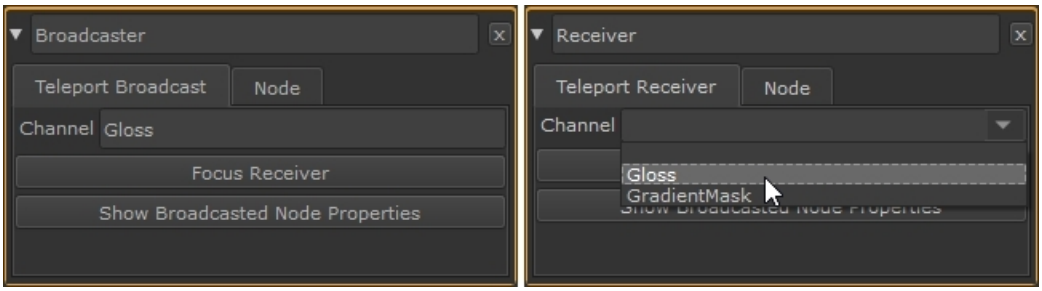
Access: **Nodes** > **Misc** > **Teleport Broadcaster** or **Teleport Receiver**

The Teleport nodes are designed to help declutter your node graph by transmitting data across hidden connections. Any node can be plugged into the **Input** of a Teleport Broadcaster which is then transmitted to one or more Teleport Receiver nodes, even from within nested groups and materials.

As of Mari 7, The Teleport Broadcaster includes an output, so that in addition to the broadcast, you have the option to chain the node in the usual way as part of the node graph data flow.

Tip: A broadcast can be made to multiple receivers, but a receiver can only listen to one broadcast.

To create a connection between a Teleport Broadcaster and a Teleport Receiver node, just use the same **Channel** name in both nodes. You can enter a name into the **Channel** control in the node **Properties** manually or select an existing **Channel** from the dropdown in the receiver's **Properties**.



The broadcast and received **Channel** name is displayed on the nodes in brackets, **(Gloss)** in the example.

Tip: You can change the node color of broadcasters and receivers in the **Preferences** under **Node Graph** > **Node Category Colors** > **Teleport**.

Teleport Broadcast Node Properties

Channel *text field*	Enter the **Channel** name broadcast from this node to Teleport Receivers in the node graph. The name can be anything, but it's good practice to use a descriptive name so that you can quickly pick the correct broadcast in the receiver.
Focus Receiver *button*	Click to display a list of receivers with the **Channel** control set to this broadcast. Select the required receiver to center the node graph on the chosen Teleport Receiver node.
Show Broadcasted Node Properties *button*	Click to quickly display the **Properties** of the node connected to the Teleport Broadcaster node's input.

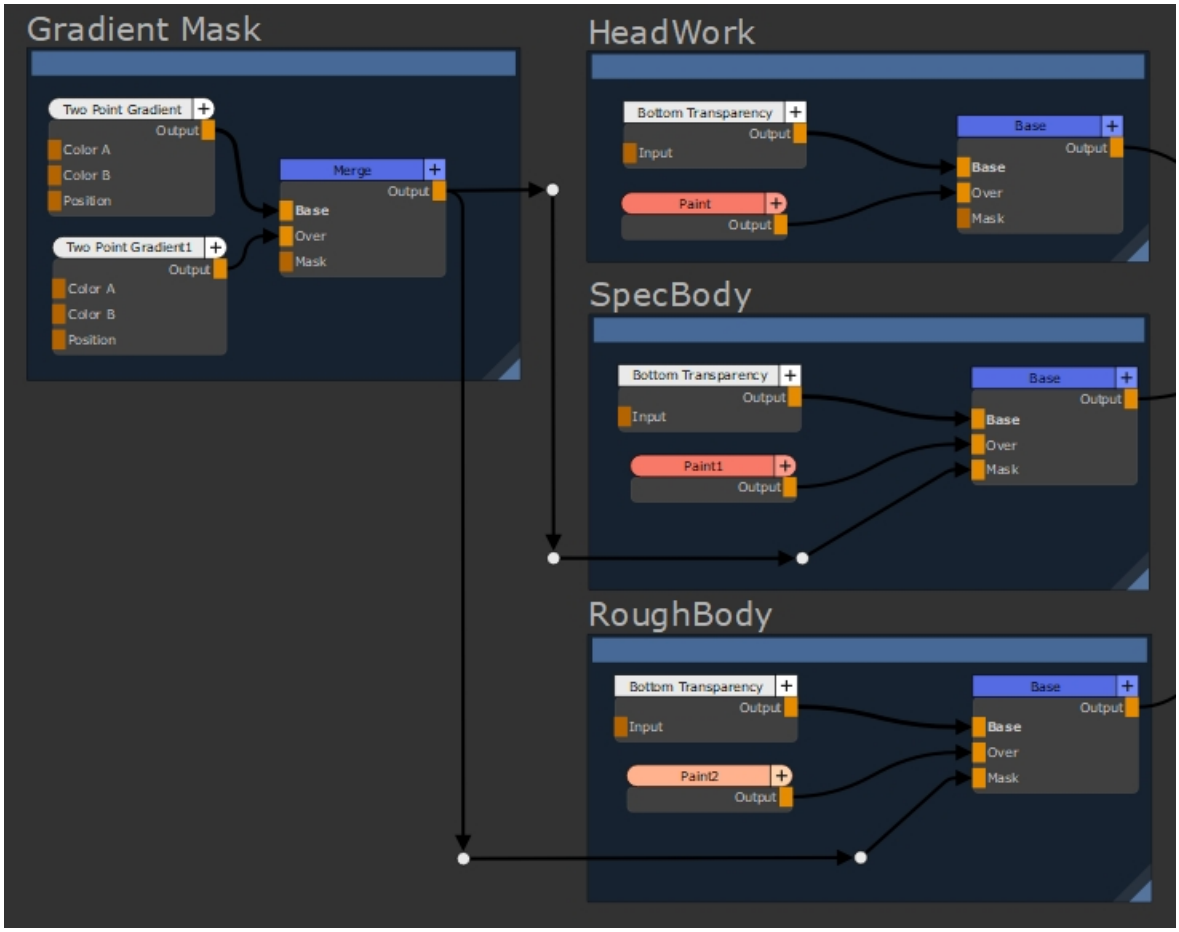
Teleport Receiver Node Properties

Channel *text field, dropdown*	Enter the **Channel** name to receive from a Teleport Broadcaster node in the node graph or use the dropdown to choose from the available broadcasts.
Focus Connected Broadcaster *button*	Click to center the node graph on the Teleport Broadcaster node transmitting the **Channel** this node is receiving.
Show Broadcasted Node Properties *button*	Click to quickly display the **Properties** of the node connected to the Teleport Broadcaster node's input.

Teleport Broadcaster and Teleport Receiver Nodes Workflow Example

Broadcast a Mask to Multiple Node Graph Locations

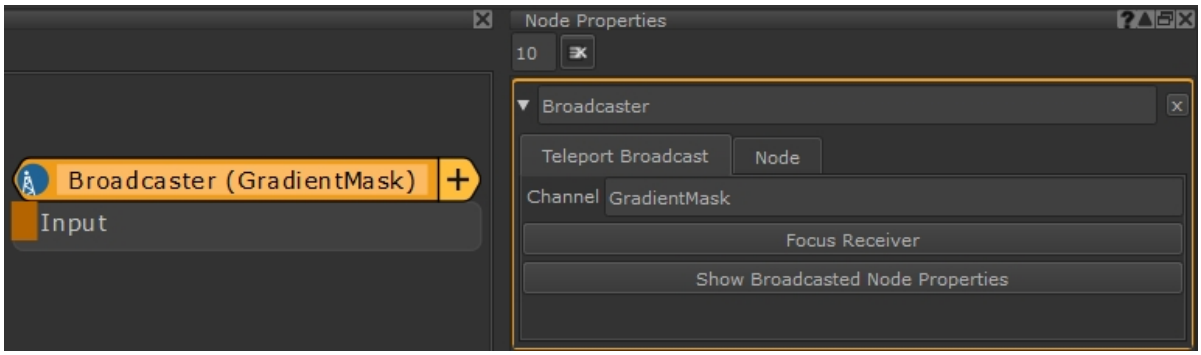
In this simple example, we'll reuse a gradient mask in specular and roughness operations without cluttering up the node graph with Dot nodes and connectors. The image shows the **Gradient Mask** connected up to the **Mask** inputs of two Merge nodes using standard Mari connections.



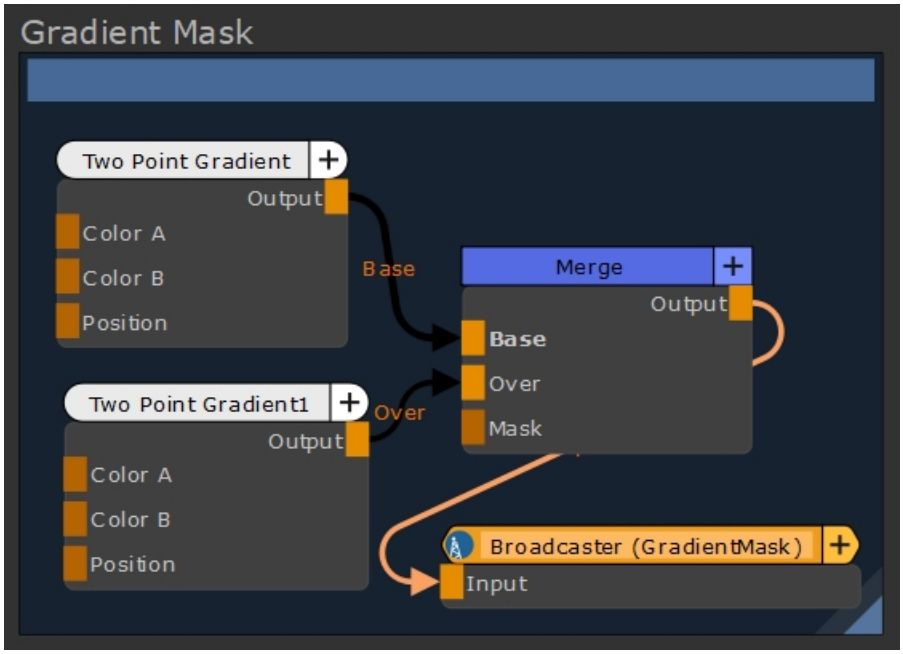
To tidy up the node graph, let's replace the connecting pipes and Dot nodes by broadcasting the mask and then receiving the data closer to the **Mask** inputs.

1. Press **Tab** in the node graph, type **Teleport**, and then select **Teleport Broadcaster**.
2. Double-click the node to open its **Properties** panel.

3. In the **Channel** control, enter the name of the broadcast. Let's call the broadcast **GradientMask**. The name can be anything, but it's good practice to use a descriptive name so that you can quickly pick the correct broadcast in the receiver.

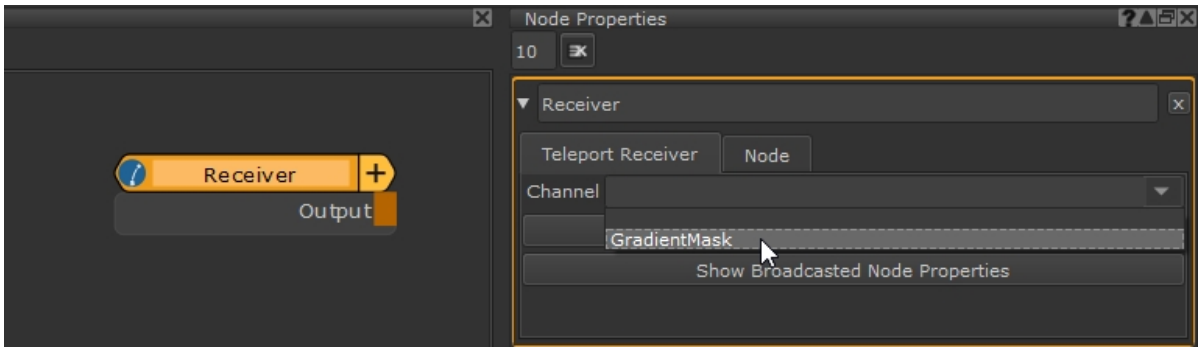


4. Connect the Gradient Mask **Output** to the broadcaster.



5. Press **Tab** in the node graph, type **Teleport**, and then select **Teleport Receiver**.
6. Double-click the node to open its **Properties** panel.
7. In the **Channel** control, click the dropdown and select the broadcast **GradientMask**.

> **Tip:** You can type the **Channel** name manually if you prefer.

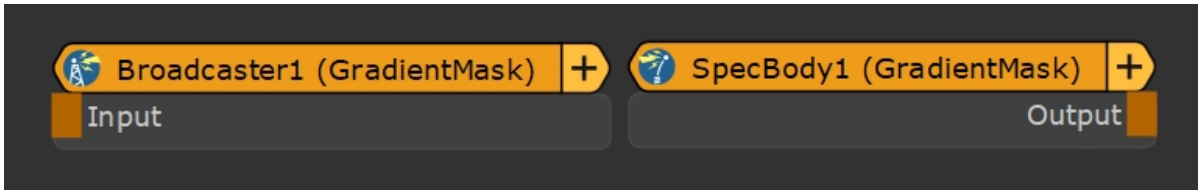


You can change the name of the receiver for clarity in the node graph. For example, you could name this receiver **SpecBody** and create another receiver called **RoughBody** to differentiate between the two.

> **Tip:** You can also change the node color of broadcasters and receivers in the **Preferences** under **Node Graph** > **Node Category Colors** > **Teleport**.

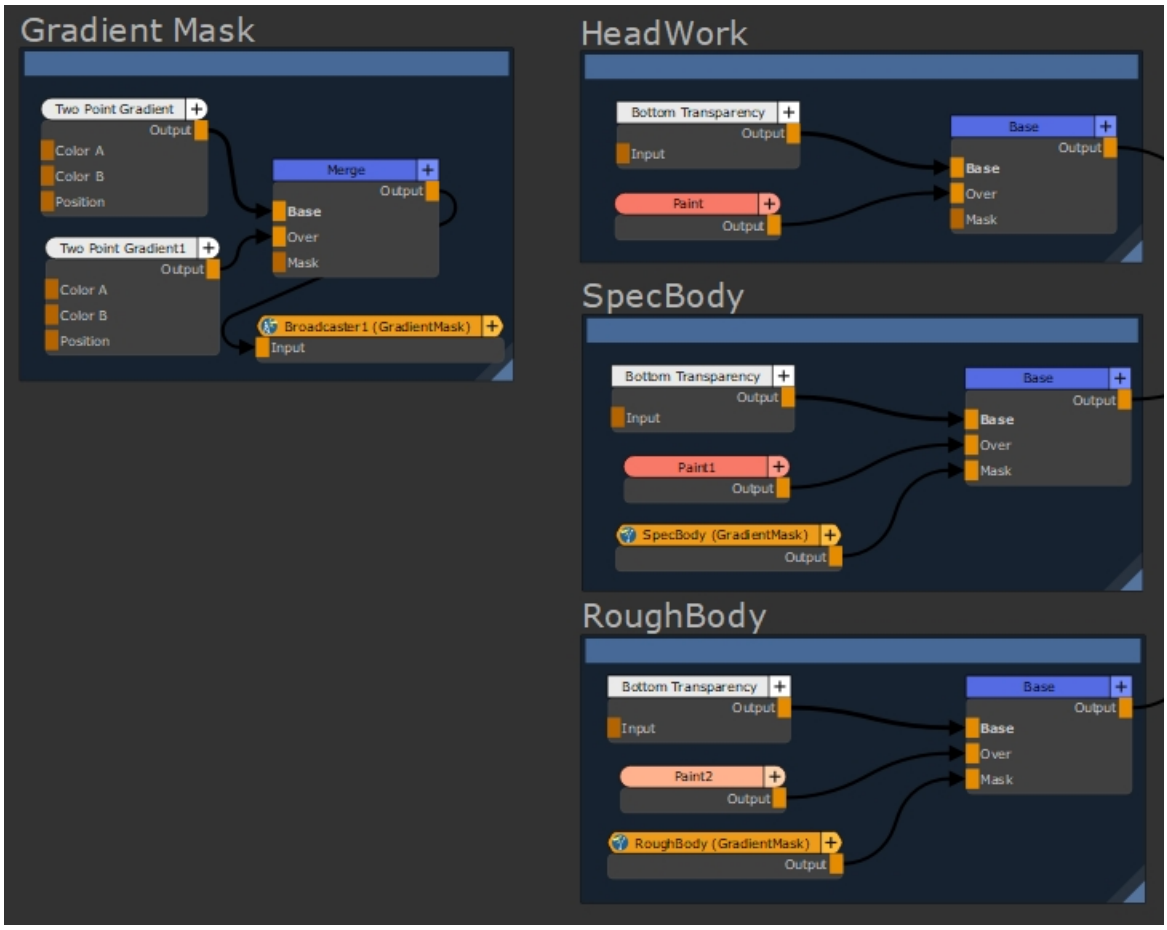
You can see in the node graph when broadcasters and receivers are working correctly because the blue icon on the left-hand side of the nodes changes to indicate a signal.

A disabled broadcast pair



An enabled broadcast pair

8. Connect the receivers to the **Mask** inputs of the required nodes to complete the clean-up.

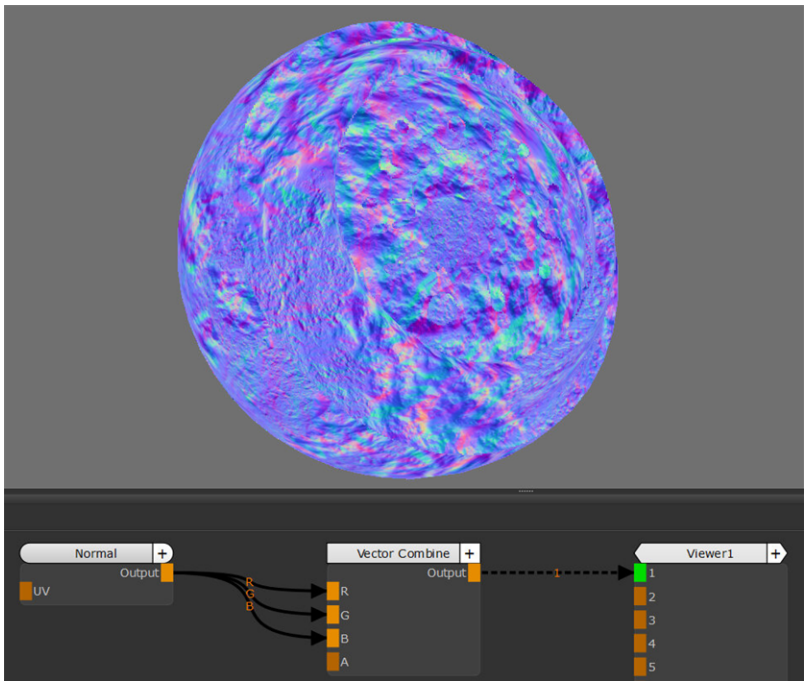


Vector Combine Node

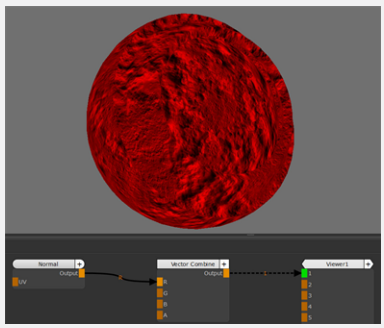
Access: **Nodes** > **Misc** > **Vector Combine**

The Vector Combine node combines components of four input vectors into a single output vector. This node has been designed to be very flexible and allows you to have complete control over the individual components of a vector. The Vector Combine node can be used to extract or make adjustments to the R, G, B and A components separately and then reconstruct them to output a new vector.

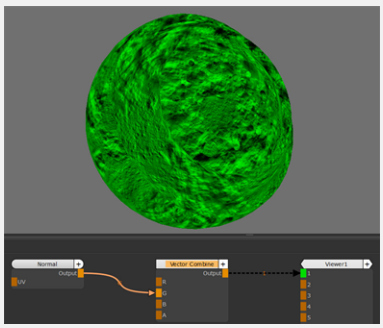
Vector Combine is often used after a Vector Split node to stitch a vector back together after it has been separated.



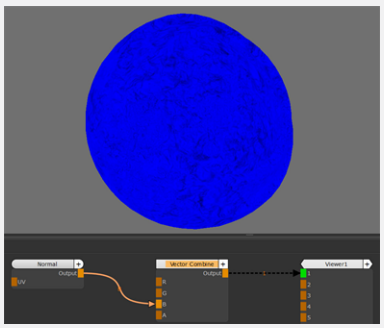
R, G and B input taking the respective components from the vector input resulting in an unchanged vector



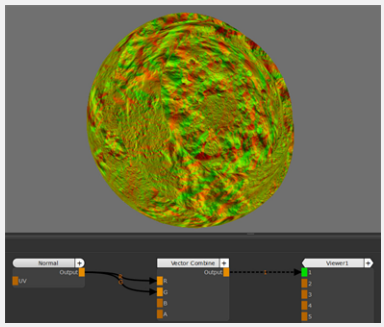
R input taking the red component from the input vector



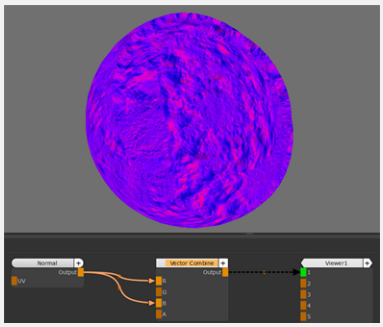
G input taking the green component from the input vector



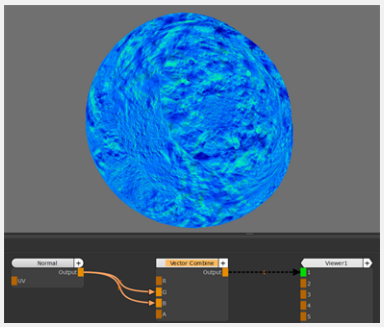
B input taking the blue component from the input vector



R and **G** inputs taking the



R and **B** inputs taking the



G and **B** inputs taking the

respective components from the input vector	respective components from the input vector	respective components from the input vector

Vector Combine Node Inputs

R	The vector to use as the input for the red component. If nothing is input, this will default to the red component (the first field) of the **Default Inputs** attribute in the **Node Properties**.
G	The vector to use as the input for the green component. If nothing is input, this will default to the green component (the second field) of the **Default Inputs** attribute in the **Node Properties**.
B	The vector to use as the input for the blue component. If nothing is input, this will default to the blue component (the third field) of the **Default Inputs** attribute in the **Node Properties**.
A	The vector to use as the input for the alpha component. If nothing is input, this will default to the alpha component (the forth field) of the **Default Inputs** attribute in the **Node Properties**.

Vector Combine Node Outputs

Output	This outputs a vector comprised of the chosen components of the input vectors.

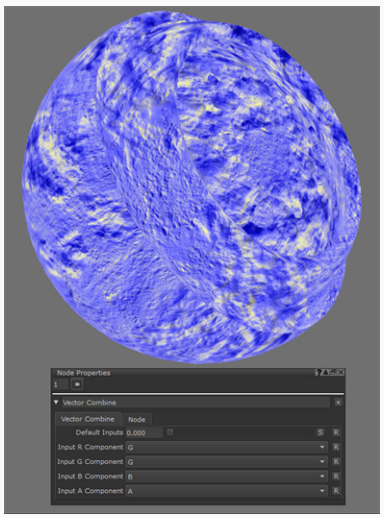
Vector Combine Node Properties

Default Inputs *text field, slider*	This defines the default component values to use if the node has nothing connected to their respective input ports. **Note:** Click the button to open the text fields for the individual vector components to use as the **Default Inputs**.

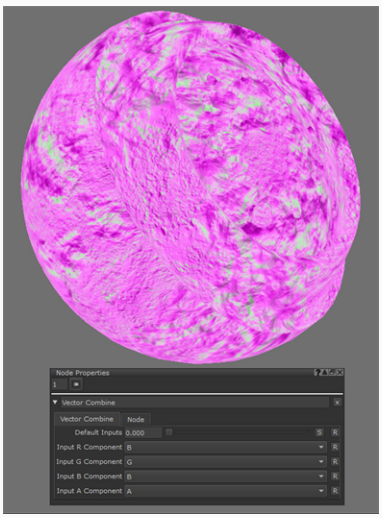
Input R Component

dropdown

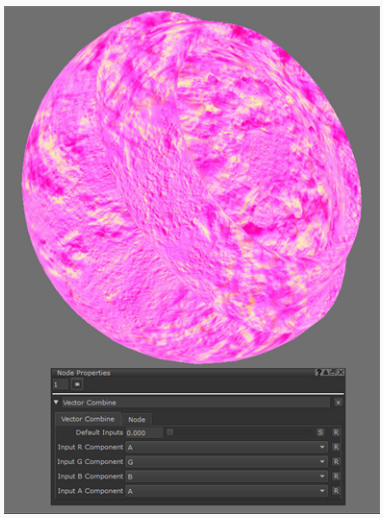
Select the component of the input **R** to use as the red component for the output vector.



G input used as **R** Component



B input used as **R** Component

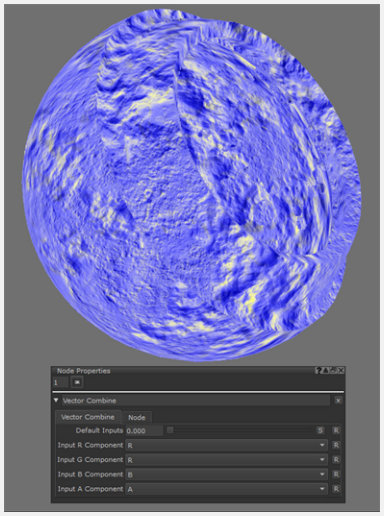


A input used as **R** Component

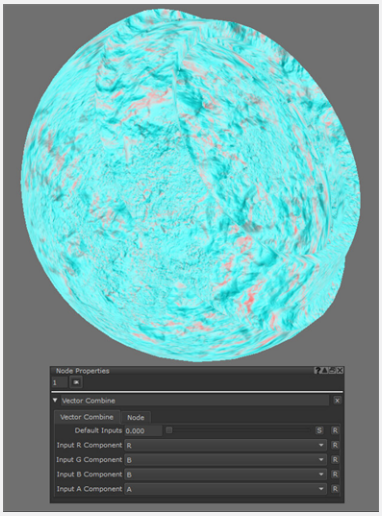
Input G Component

dropdown

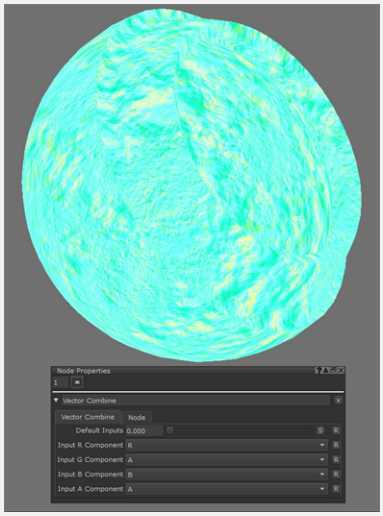
Select the component of the input **G** to use as the green component for the output vector.



R input used as **G** Component



B input used as **G** Component

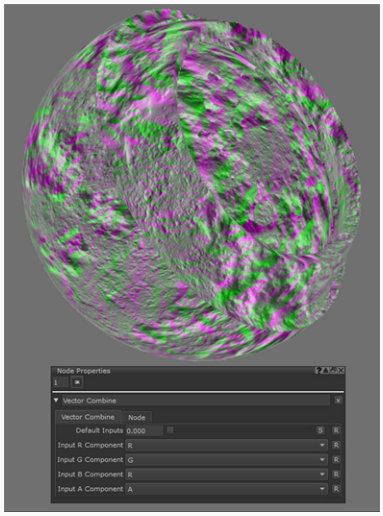
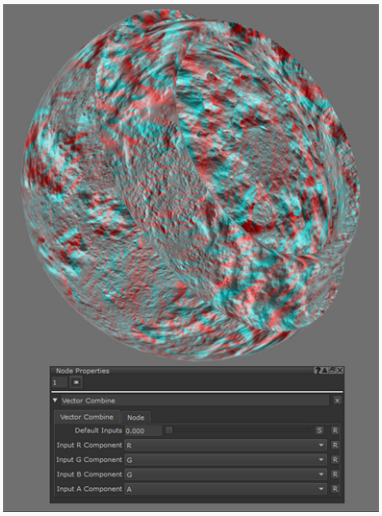
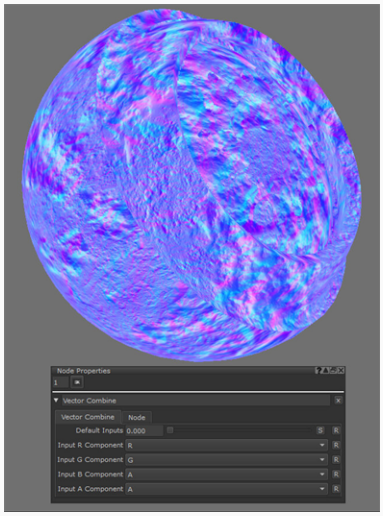
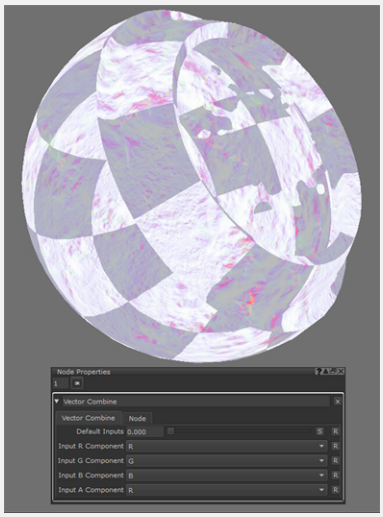
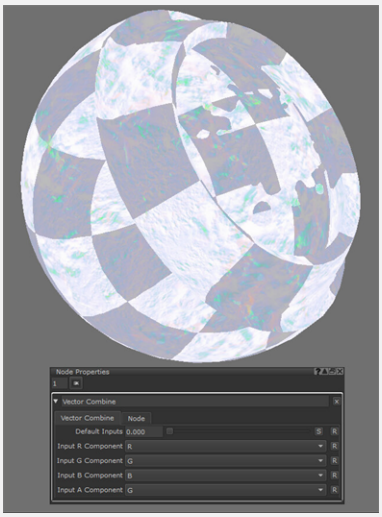
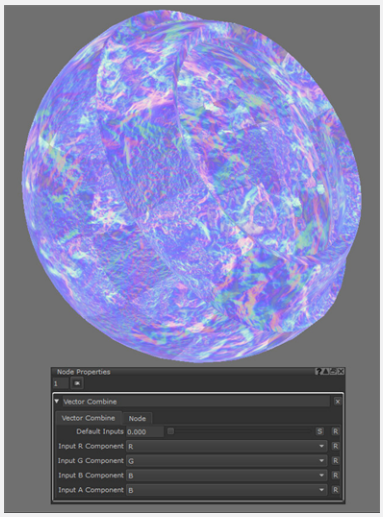


A input used as **G** Component

Input B Component

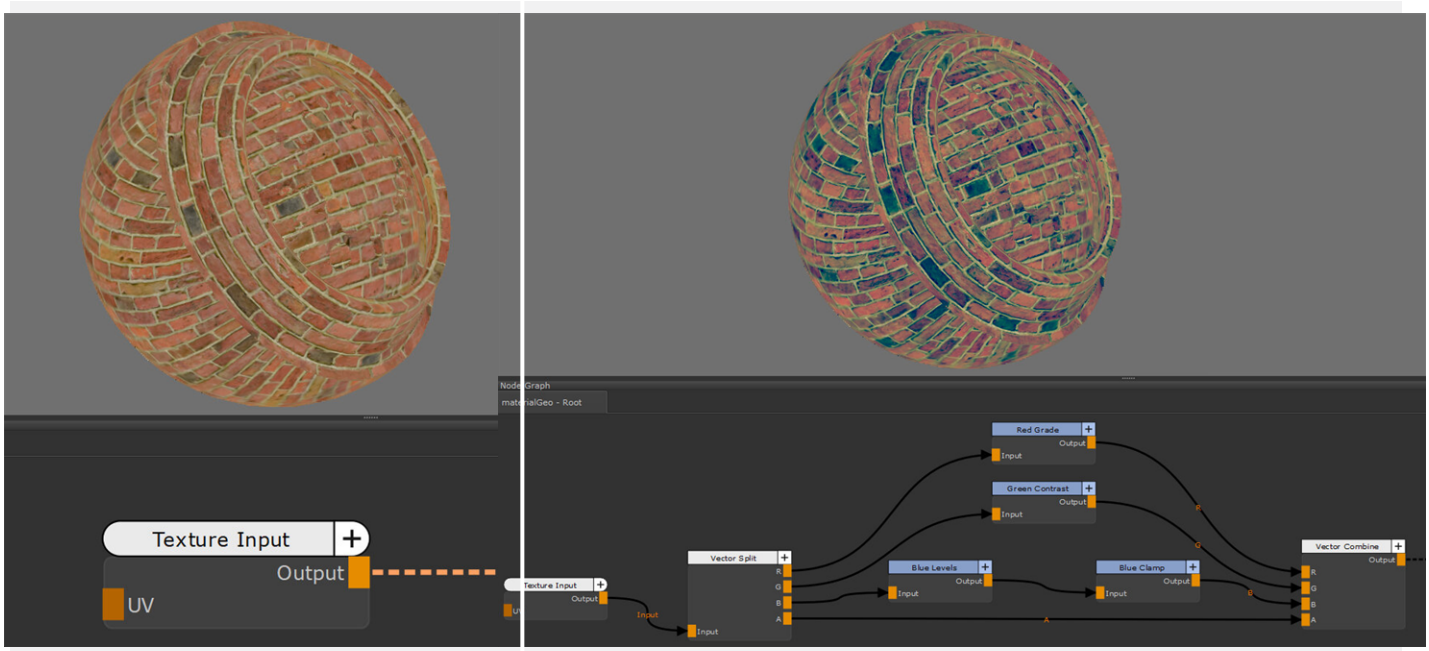
dropdown

Select the component of the input **B** to use as the blue component for the output vector.

	 **R** input used as **B** Component  **G** input used as **B** Component  **A** input used as **B** Component
Input A Component *dropdown*	Select the component of the input **A** to use as the alpha component for the output vector.  **R** input used as **A** Component  **G** input used as **A** Component  **B** input used as **A** Component

Workflow Example

Vector Combine is closely linked to the Vector Split node and the two are usually used together. One common workflow is to use Vector Split to separate out individual components, make adjustments to one or more, and then feed the modified component back using Vector Combine.



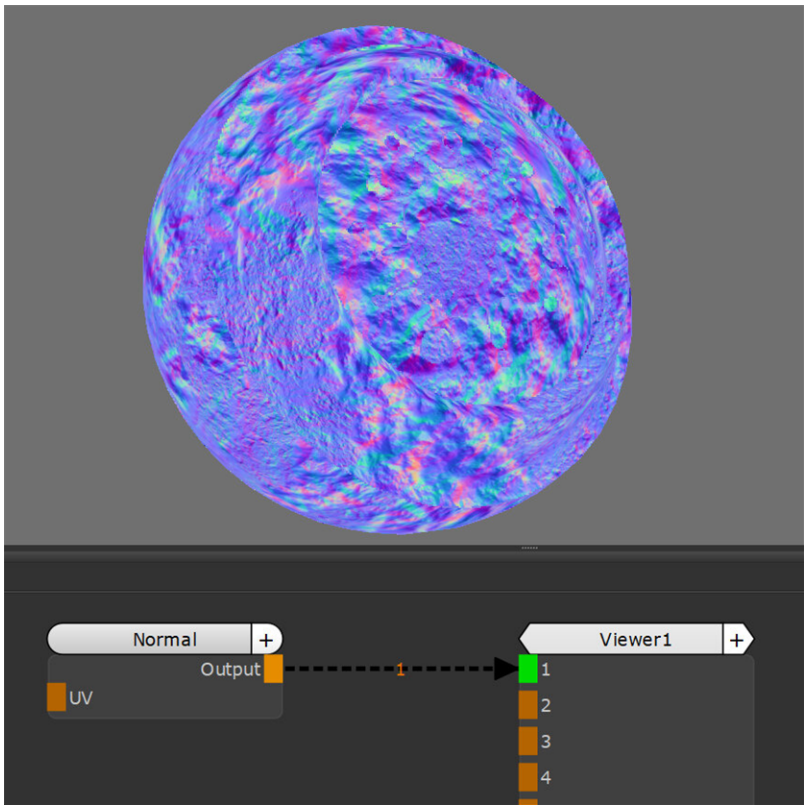
Texture Input | Texture Output after Adjustments

This use of the Vector Split and Vector Combine nodes can be especially useful when working with textures for video games. These textures are often packed which means multiple channels, such as **Roughness**, AO and **Specular**, are stored in a single channel in an image. The Vector Split node allows you to separate these components to use in Mari and the Vector Combine node can then be used to put them back together to export to a game engine.

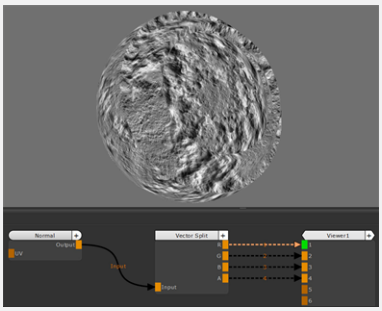
Vector Split Node

Access: **Nodes** > **Misc** > **Vector Split**

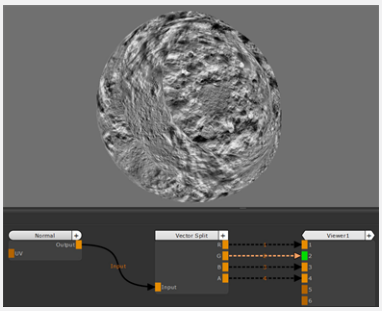
The Vector Split node splits apart a vector into its individual components (R, G, B, A). This node can be useful if you want to isolate a certain color from a vector in order to make adjustments to it separately or to extract it entirely from and image or map.



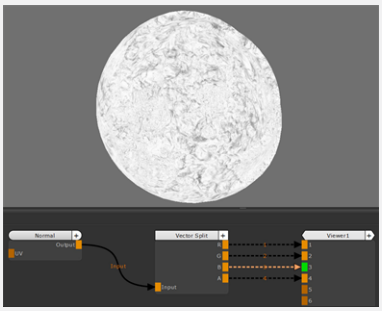
Example Vector input



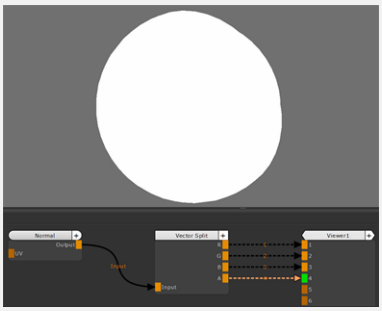
R Output



G Output



B Output



A Output

Vector Split Node Inputs

Input	This input takes the vector to be split apart. If nothing is plugged in, this defaults to the value in the **Default** attribute in the **Node Properties**.

Vector Split Node Outputs

R	This will output the red component of the **Input** vector in the format (R,R,R,1).
G	This will output the green component of the **Input** vector in the format (G,G,G,1).
B	This will output the blue component of the **Input** vector in the format (B,B,B,1).
A	This will output the alpha component of the **Input** vector in the format (A,A,A,1).

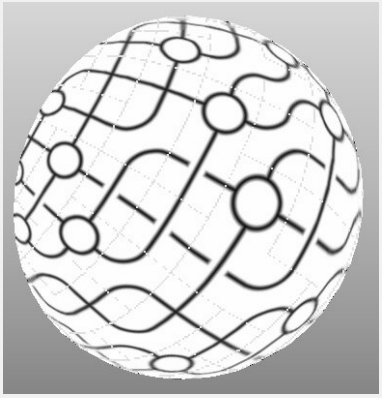
Vector Split Node Properties

Default *color select*	This defines the default vector to split apart if the node has nothing connected to its **Input** port.

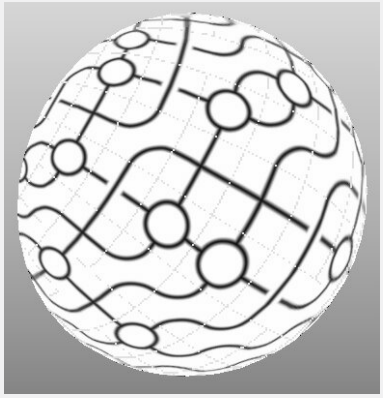
Atlas Random Node

Access: **Nodes** > **Procedural** > **Pattern**

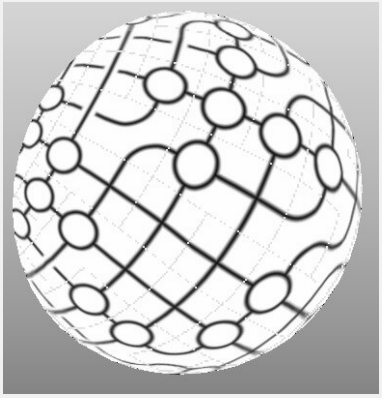
The Atlas Random node takes an atlas texture and selects a random image (or tile) within the atlas texture to use for each patch on the object.



Atlas Random node used with an unsymmetrical tiled image, with a **Horizontal Tiles** and **Vertical Tiles** value of 2, and **Random Seeds** W value of 20.



Atlas Random node used with an unsymmetrical tiled image, with a **Horizontal Tiles** and **Vertical Tiles** value of 2, and **Random Seeds** W value of 10.



Atlas Random node used with an unsymmetrical tiled image, with a **Horizontal Tiles** and **Vertical Tiles** value of 2, and **Random Seeds** W value of 1.

Atlas textures are single images containing a collection of smaller, uniform or non-uniform dimensional images. These textures are predominately used in game development, since atlas textures are useful for saving memory and texture fetching by using a single image that can be used multiple times, instead of caching multiple images. The images (or tiles) within the atlas texture can be commonly identified in two ways:

- Multiple sub-maps are packed into a single bitmap.
- Different maps packed into each color channel.

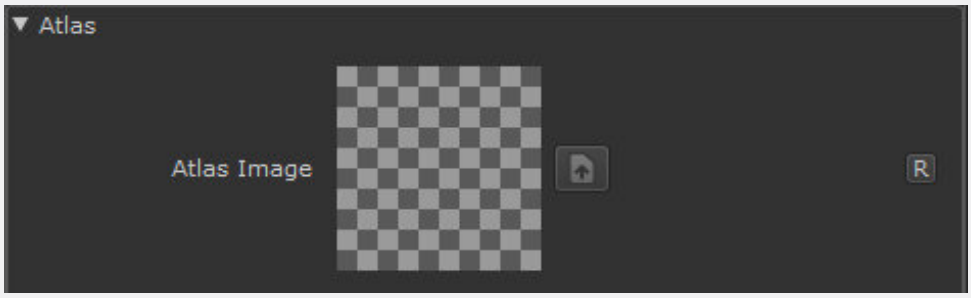
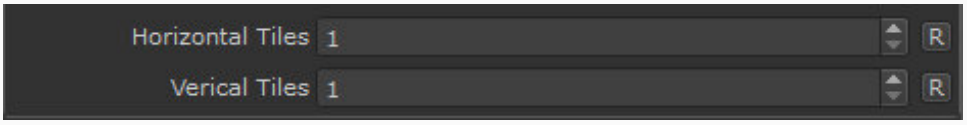
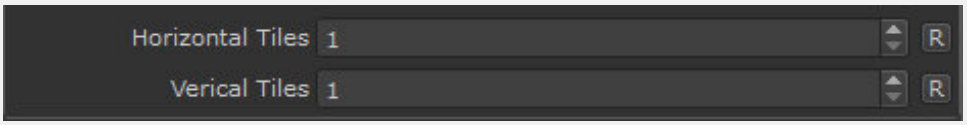
The Atlas Random node is very useful for creating random patterns or displacements when you have an asymmetric tiled image.

Atlas Random Node Inputs

UV	The UV coordinate to use for selecting the tile. Defaults to standard surface UVs.

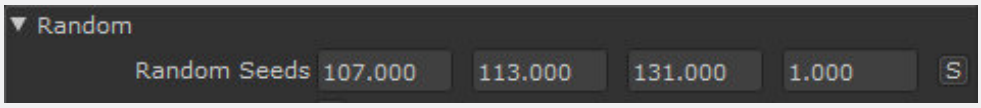
Atlas Random Node Properties

Atlas

Atlas Image *button*	Opens a file dialog for you to select an image from disk to be used as the atlas image for the node. 
Horizontal Tiles *arrows*	Select how many tiles the atlas contains horizontally. 
Vertical Tiles *arrows*	Select how many tiles the atlas contains vertically. 

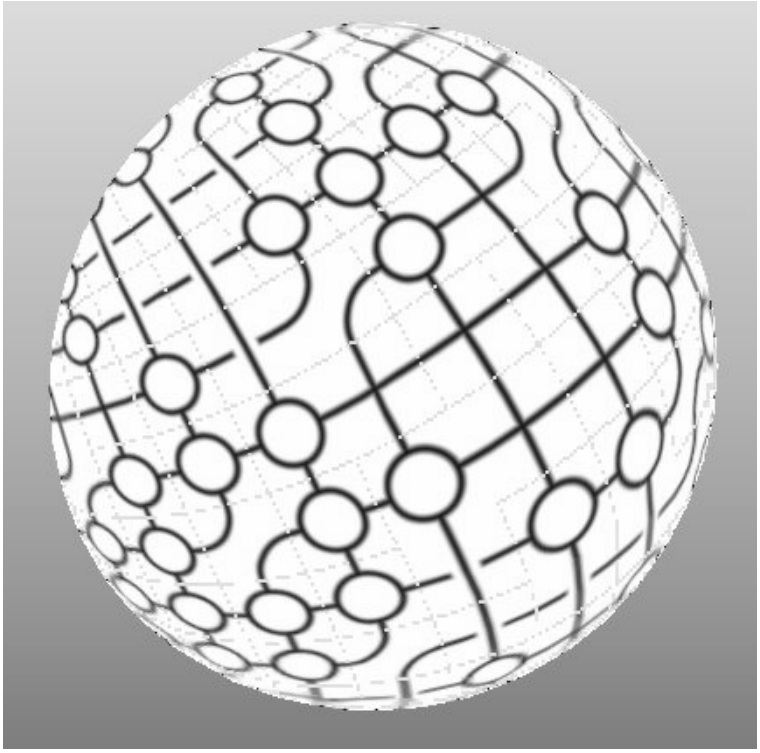
Random

Random Seeds *text field, slider*	Drives a seed for the various random properties. Each value represents a different property: • X - Mirror U. • Y - Mirror V.

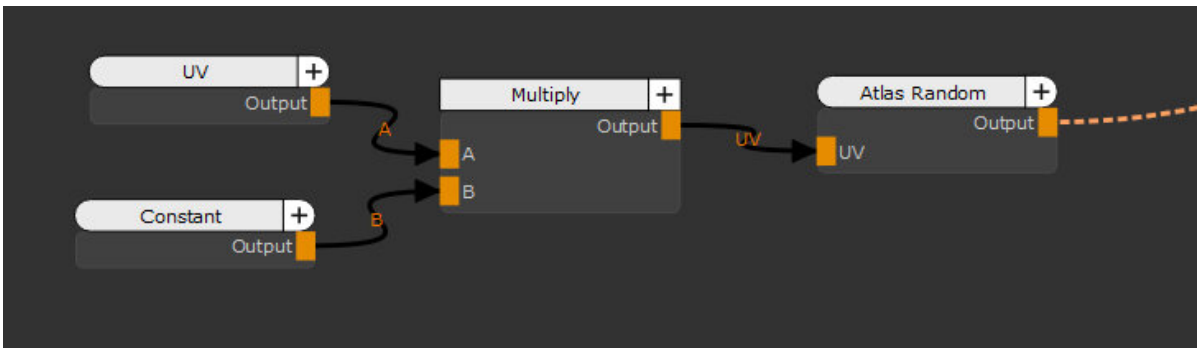
	• Z - Seed for random rotations. • W - Seed for tile choice. **Tip:** Click the **S** button to expand the X, Y, Z, W property inputs.  When the properties are not expanded, the slider drives the value into the X, Y, Z properties uniformly, and sets the W value to 1.
Random Horizontal Mirror *checkbox*	When turned on, this property randomly flips tiles horizontally.
Random Vertical Mirror *checkbox*	When turned on, this property randomly flips tiles vertically.
Random Rotations *checkbox*	When turned on, this property random rotates tiles in 90 degree increments.
Is Normal Map *checkbox*	If **Random Rotations** is enabled and the atlas image is an atlas of normal maps, this rotates the normal vectors to account for the random rotation when activated.

Atlas Random Node Workflow Example

The Atlas Random node can be used with tileable asymmetric textures to create random textures, like seen below. This could be useful for creating random gridlines, and circuit board patterns.



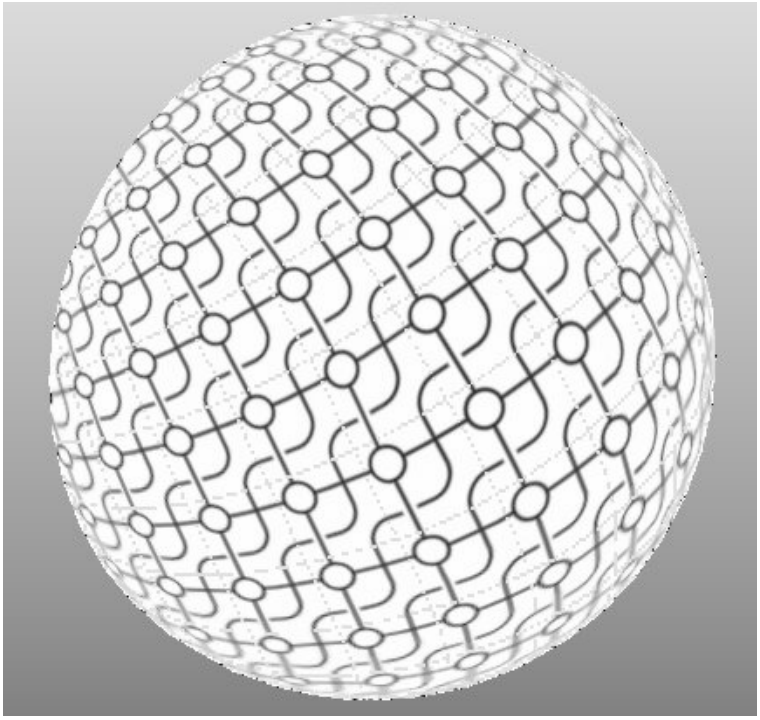
In order to achieve the effect seen here, setup your Node Graph using the following nodes.



The Node Graph setup used to create the randomized texture.

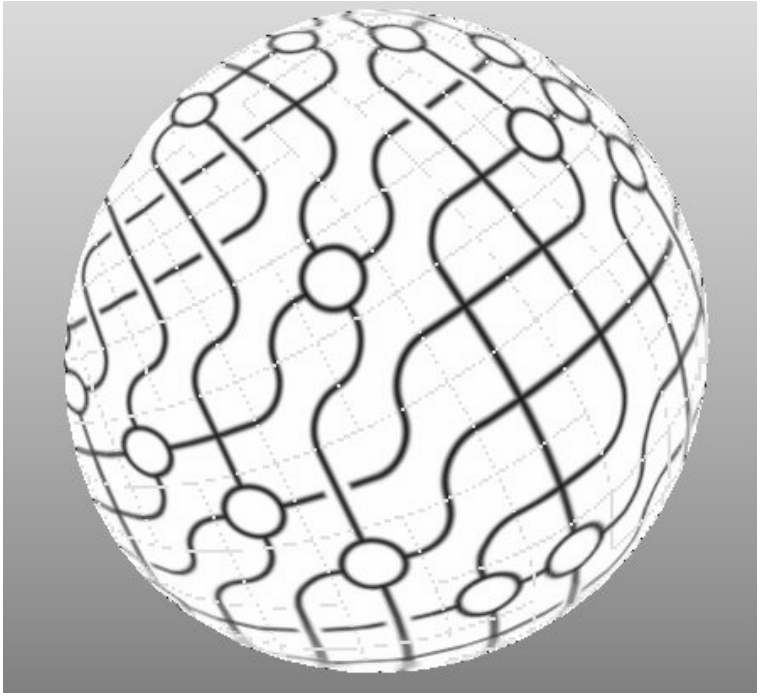
The Constant node multiplies with the UV node to set a scale for the UVs that are used by the Atlas Random node. Using this setup, with the atlas texture input into the **Atlas Texture**, creates the following effect.

In order to fix this UV scale issue, entering a value of 20 into the **Constant** property of the Constant node scales the UVs down to create the following effect.



Adjusting the **Random Seeds** property on the Atlas Random node does nothing to this pattern currently. In order to get an effect from the **Random Seeds** property, you need to increase the **Horizontal Tiles** and

Vertical Tiles values on the Atlas Random node. Then, using the **W** value on the **Random Seeds** property you can input a value for a random tile to be selected on each patch.

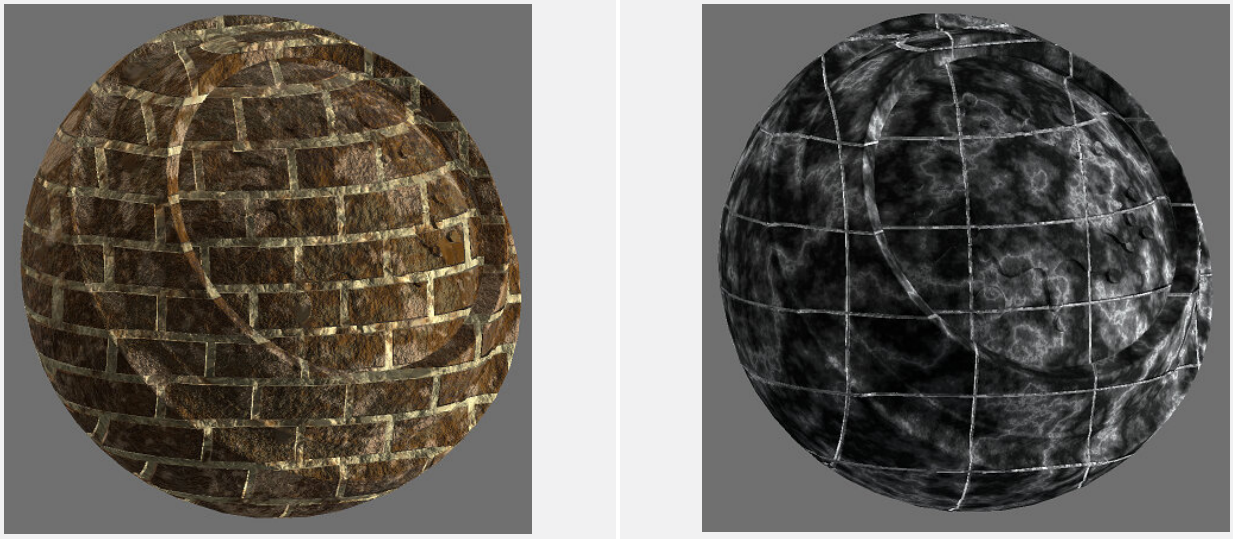


The results using a **Horizontal Tiles** and **Vertical Tiles** value of 2, and a **Random Seeds W** value of 5.0.

Bricks Node

Access: **Nodes** > **Procedural** > **Pattern** > **Bricks**

The Bricks node generates a pattern of bricks or tiles. These patterns can then be used as a height map, mask or both to create procedurally driven textures and materials.



Bricks node examples with varying parameters

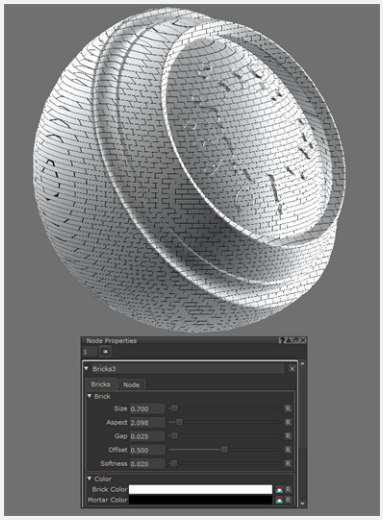
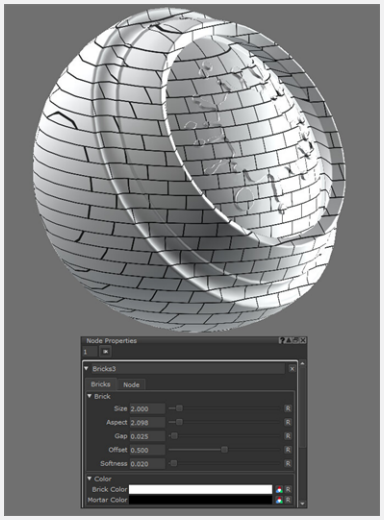
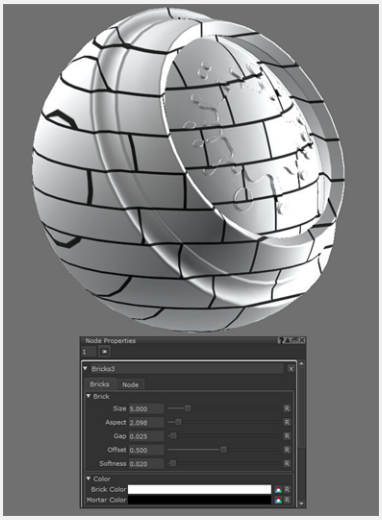
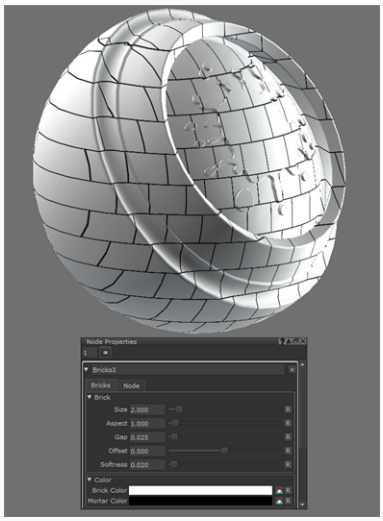
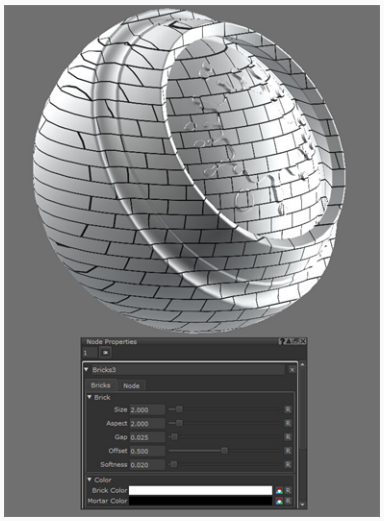
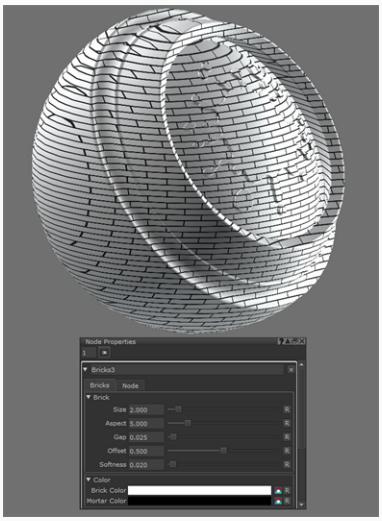
Bricks Node Inputs

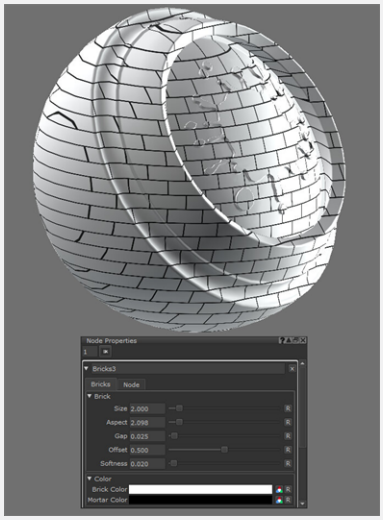
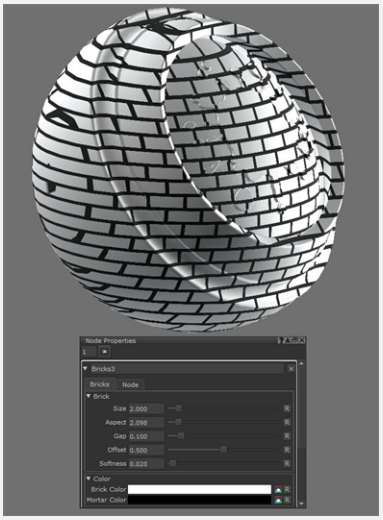
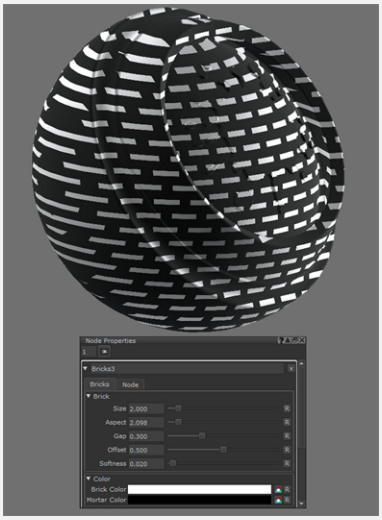
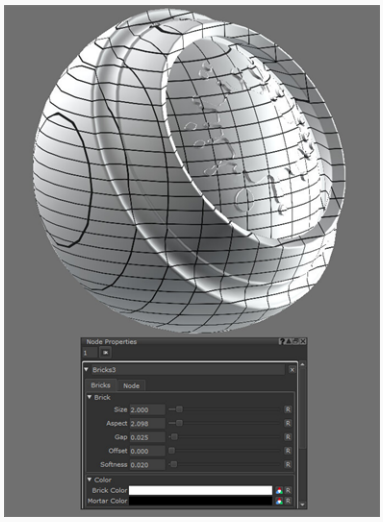
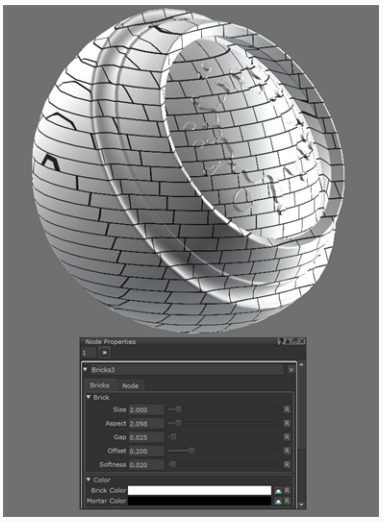
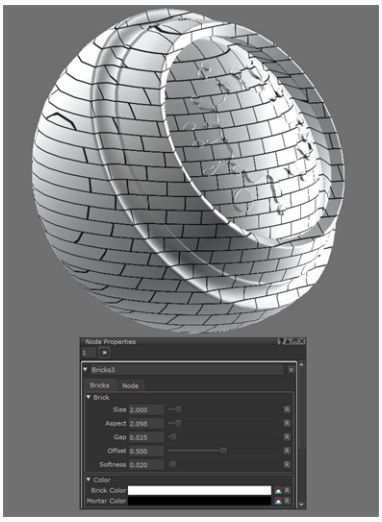
Input	Description
Position	The position to generate the brick pattern from. If nothing is input, this defaults to surface world position.

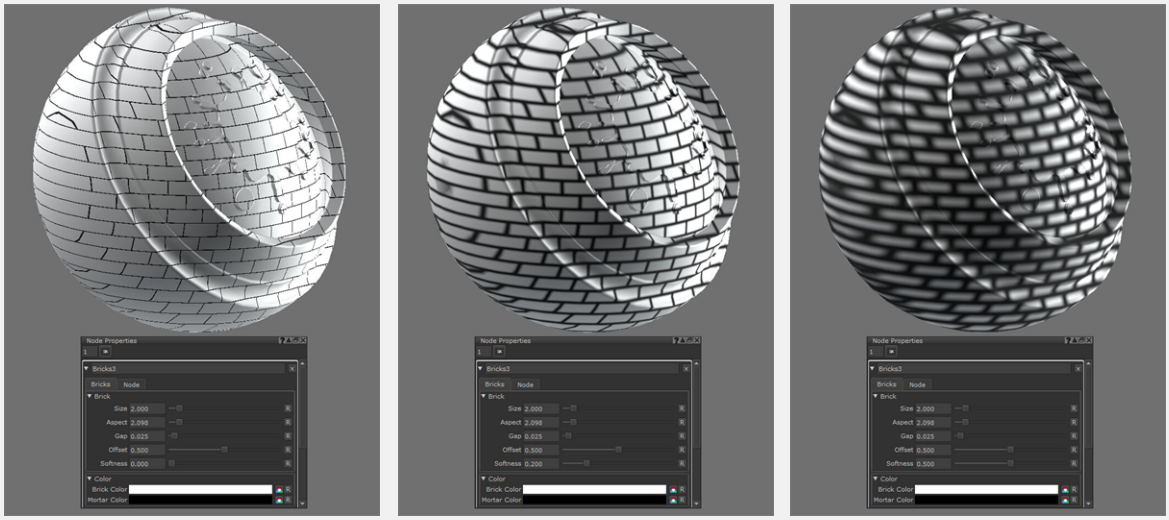
Bricks Node Outputs

Output	Description
Output	This outputs the defined **Brick Color** where there are bricks, **Mortar Color** where there are gaps and a blend of the two colors based on the **Softness**.
Brick UV/Mask	This outputs RGBA data where: **RG** = The UV coordinates for each brick. **B** = White where there are bricks, black where there are gaps, and a blend based on the Softness. **A** = 1.0

Bricks Node Properties

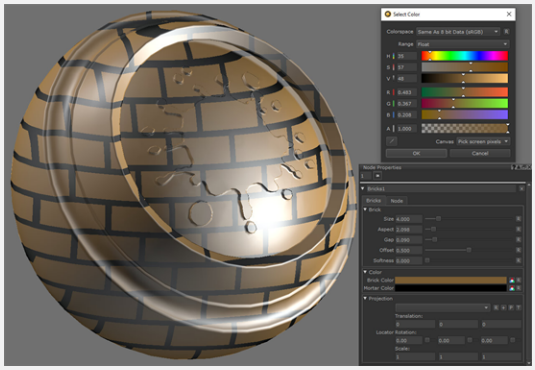
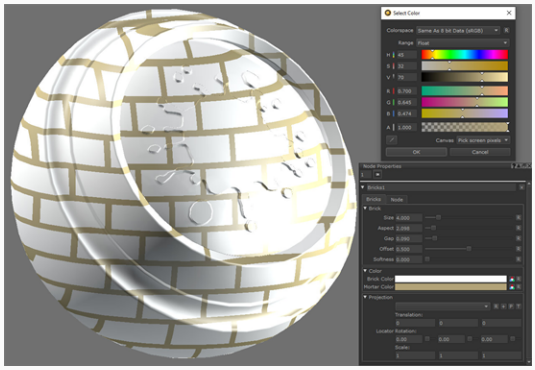
Size *text field, slider*	This value defines the scale of the bricks.    **Size**: 0.7 **Size**: 2 **Size**: 5
Aspect *text field, slider*	This defines the aspect ratio of the bricks. When this value is changed, the brick height is adjusted to match aspect ratio.    **Aspect**: 1 **Aspect**: 2 **Aspect**: 5
Gap *text field, slider*	This defines the size of the gap (mortar) between the bricks.

	   **Gap**: 0.025 **Gap**: 0.1 **Gap**: 0.3
Offset *text field, slider*	This defines the horizontal offset, in brick widths, of each row of bricks.    Offset: 0 Offset: 0.2 Offset: 0.5
Softness *text field, slider*	This value defines how soft the edges between the bricks and the gap (mortar) is.



Softness: 0 | **Softness**: 0.2 | **Softness**: 0.5

Color

Brick Color *text field, slider*	The sets the color of the brick shapes. 
Mortar Color *text field, slider*	This sets the color of the gap (mortar) between the bricks. 

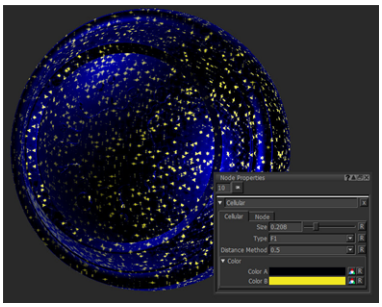
Projection

Projector *dropdown list*	This allows you to select a projector to align and scale the projected pattern. The list displays all the Objects, Locators, Cameras and Lights in the scene which can be used as a projector.
+ *button*	Add a new locator and set it as the Projector.
P *button*	Make the projector locator current and switch to the Transform Object Tool.
T *button*	Toggle the visibility of the transform fields.
Translation *text field*	Used to translate the projection in the x, y and z axis. **Note:** The transform fields (**Translation**, **Locator Rotation**, **Scale**) lock if you have something selected in the field. The transform field values mirrors those of whatever you have selected.
Locator Rotation *text field*	Used to rotate the projection around the x, y and z axis.
Scale *text field*	Used to scale the projection in the x, y and z axis.

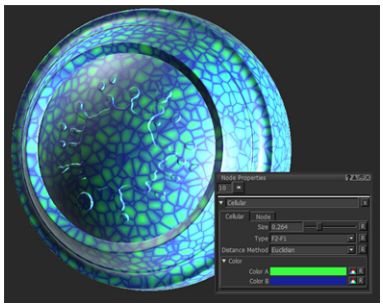
Cellular Node

Access: **Nodes** > **Procedural** > **Noise** > **Cellular**

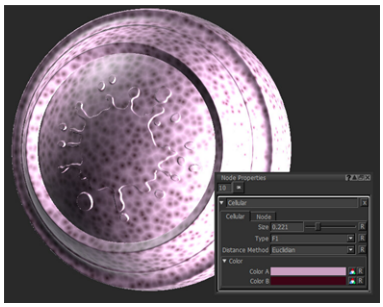
The Cellular node is a type of noise node that generates points randomly to simulate natural textures such as sparkles and gemstones, as well as organic textures such as human cells.



Sparkles



Gemstones



Human Cells

The Cellular node relies on 2D or 3D position information (set of numbers) on the canvas. Its **Position** input can be connected to any position nodes and you can adjust the position in various ways.

Tip: For example, you can use the UV and Position nodes to switch between 2D and 3D positions. You can also stretch the position using the Scale node or shift it using a Vector node.

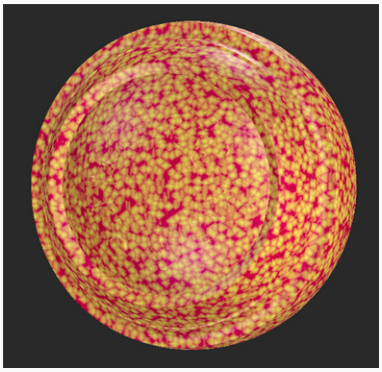
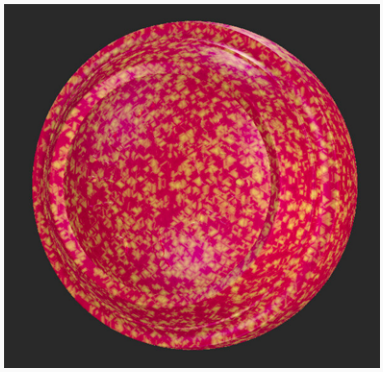
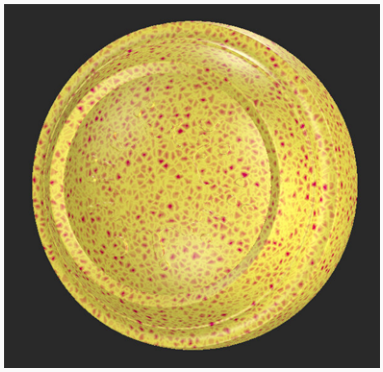
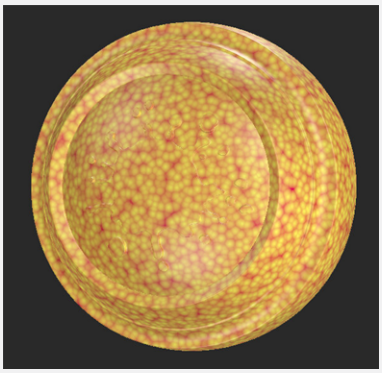
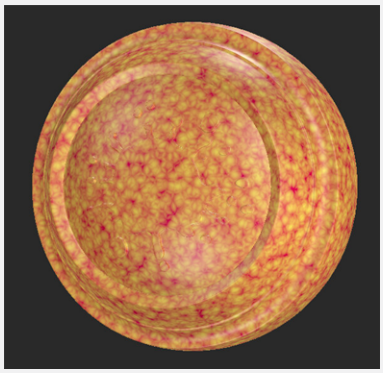
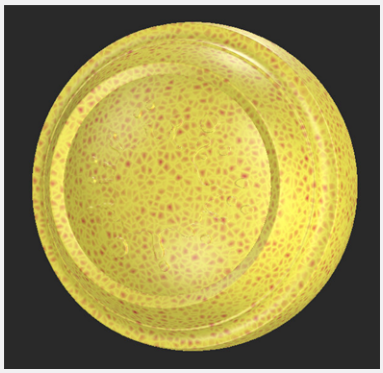
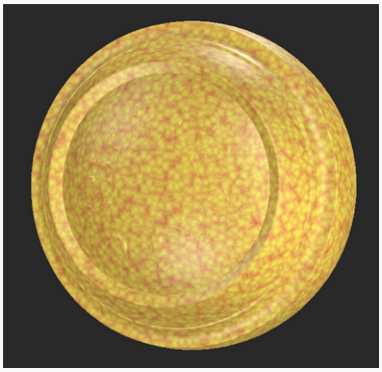
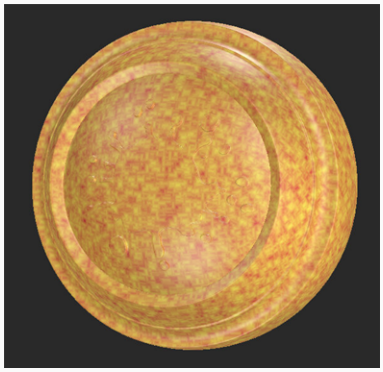
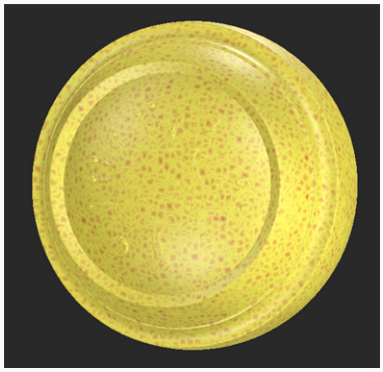
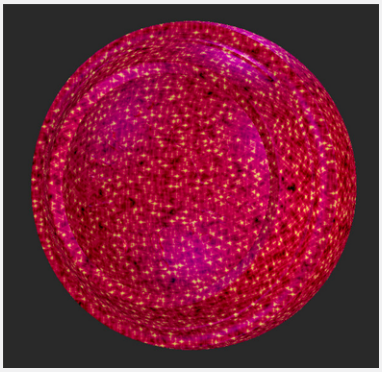
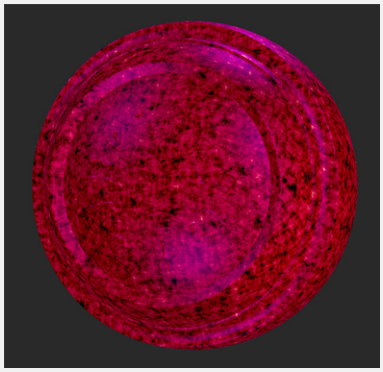
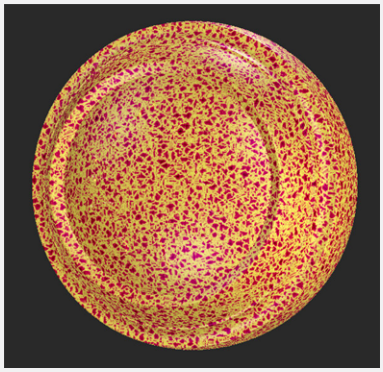
Cellular Node Inputs

Position - Any position data can be connected to the **Position** input, such as the UV and Position nodes. You can use any type of node to modify the mapping of the pattern. For example, you can use the Scale or Vector node in conjunction with the Position or UV node to stretch or offset the position (data) from the Position or UV node.

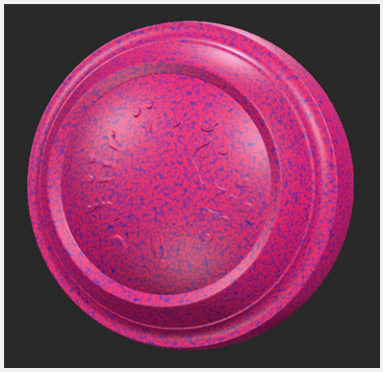
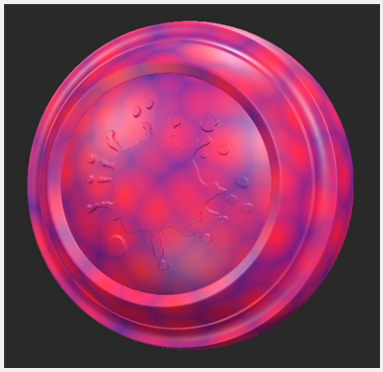
Cellular Node Properties

The table below shows the type and distance method on the x and y axes.

	F1	F2	F2-F1

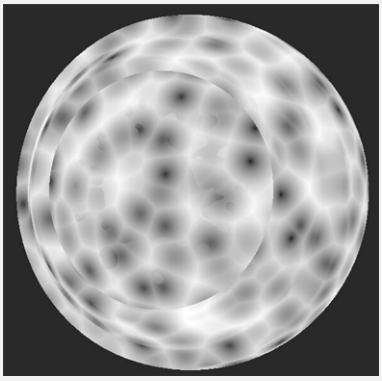
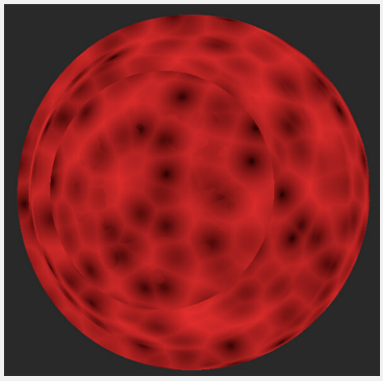
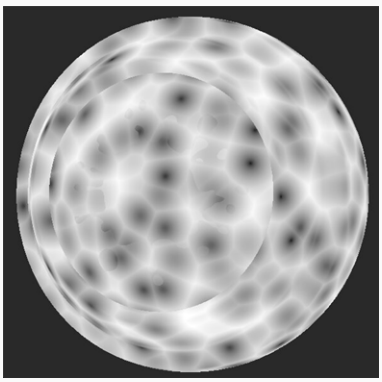
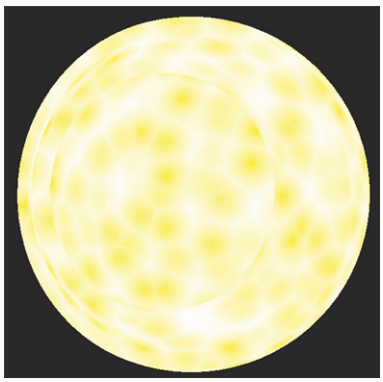
Manhattan			
Euclidian			
Chebyshev			
0.5			

The Cellular noise node generates noise by computing distance to "feature points" scattered in the coordinate space.

Size *floating point control*	Changes the size of the points driving the pattern.  **Size** set to **0.147**  **Size** set to **0.375**
Type *dropdown*	Changes how it computes the distance to render the noise: • **F1** - computes the distance to the closest point. • **F2** - computes the distance to the second closest point. • **F2-F1** - computes the difference between the distances to the second closest point and the closest point.
Distance Method *dropdown*	Changes the **Distance Method** used to render the noise: • **Manhattan** - computes distance by taking the sum of distances in Cartesian coordinates. • **Euclidian** - computes distance by the distance between two points in Euclidian space. • **Chebyshev** - computes distance by taking the longest distance in one dimension of coordinate space. • **0.5** - computes the distance by taking the p-norm where p is 0.5.

Color

Color A *swatch*	Changes the background color.

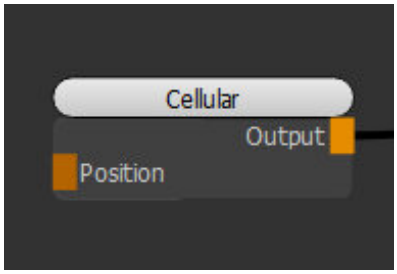
	 Default: white	 Red
Color B *swatch*	Changes the color around the points.  Default: black	 Yellow

Cellular: Node Graph Workflow Examples

Example - Shifting the Mapping Position of the Cellular Node.

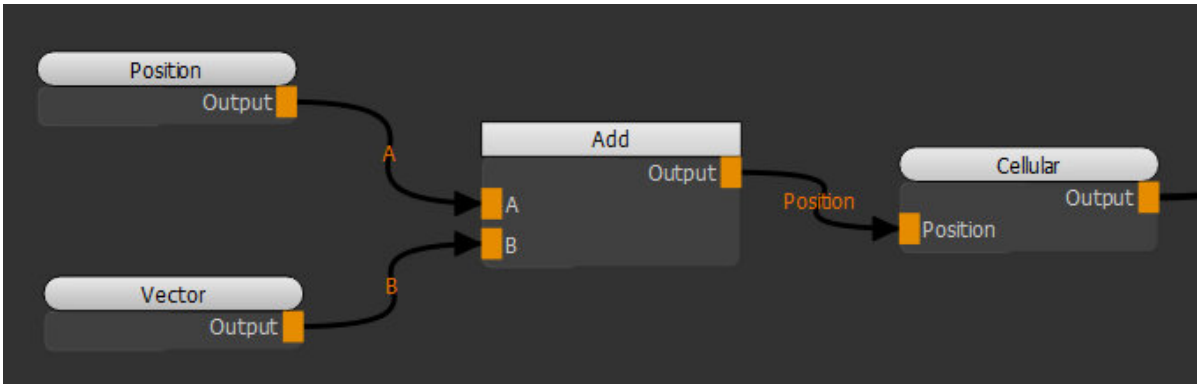
In the example below, let's add a Vector node to the baseline position to shift the noise.

1. Create a Cellular node.



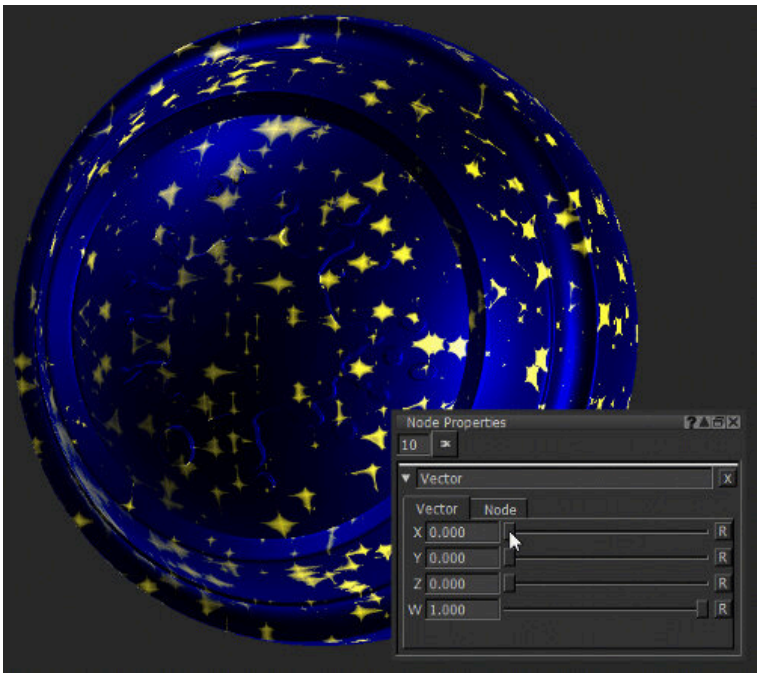
Adding a Cellular node to the scene.

2. Create an Add node and connect it to the Cellular node. The Add node lets you use two nodes in conjunction to modify the output node, in this example, the Cellular node.
   Create a Vector node and a Position node and connect them respectively to the **A** and **B** inputs of the Add node.



Adding a Position and Vector node to shift the position of the Cellular noise.

Notice how moving the X slider of the Vector node moves the noise on the X axis as we are changing the position information through the Position node.

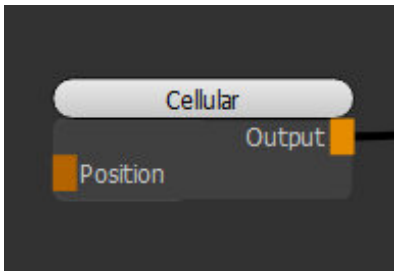


Adjusting the position with the Vector node to shift the noise position.

Example - Changing the Texture of the Cellular Noise

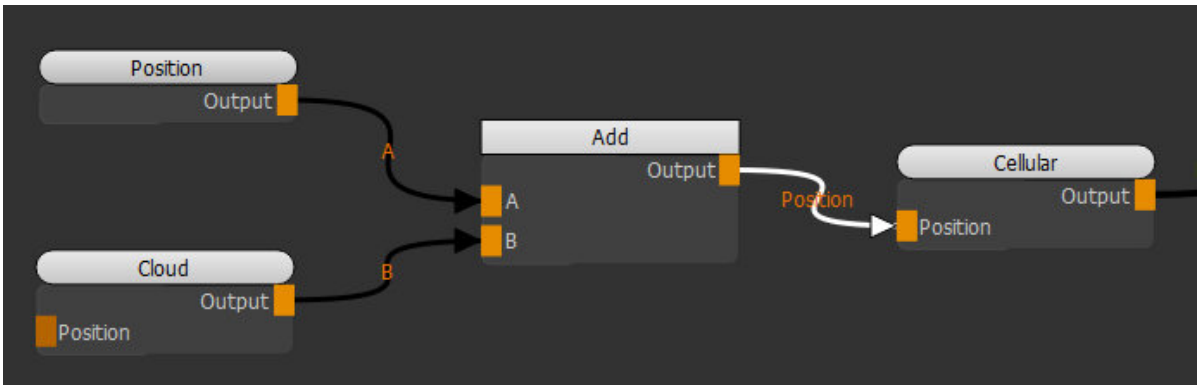
In the example below, let's add a Cloud node to the baseline position to change the texture of the Cellular noise.

1. Create a Cellular node.



Adding a Cellular node to the scene.

2. Create an Add node and connect it to the Cellular node.
   Create a Cloud node and a Position node and connect them respectively to the **A** and **B** inputs of the Add node.



Adding a Position and Cloud node to distort the Cellular noise.

Notice how adjusting the Cloud node's foreground color (Color A) distorts the Cellular noise.

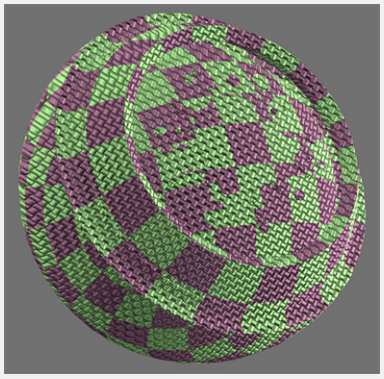
Checkerboard Node

Access: **Nodes** > **Procedural** > **Pattern** > **Checkerboard**

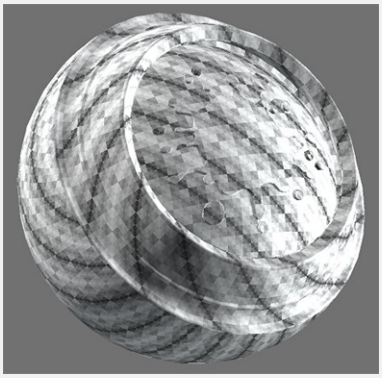
The Checkerboard node generates a checkerboard pattern in UV space. This node is primarily intended to be used as a way of checking the UV scale of a given object. This is useful for determining, and ultimately avoiding, uneven scaling across your textured object.

The checkerboards can also be used in the same way as any other procedurally generate pattern, as a mask or height map.

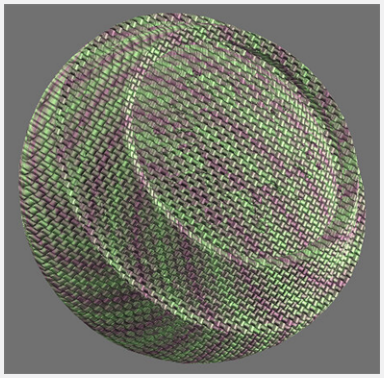
Checkerboard used as Base Color



Checkerboard use as a Mask



Checkerboard with Increased Octaves used as Base Color



Checkerboard with Increased Octaves used as a Mask

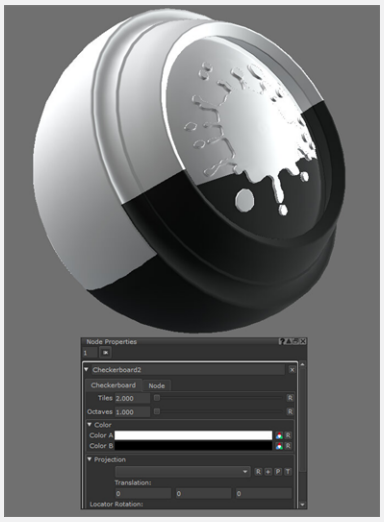
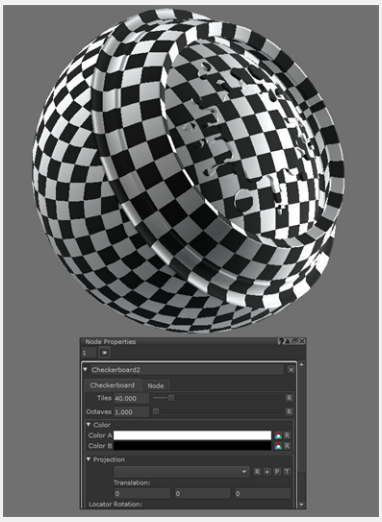
Checkerboard Node Inputs

UV	The UV position at which to generate the checkerboard pattern.

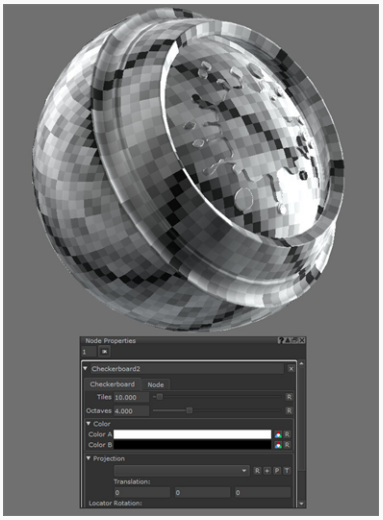
Checkerboard Node Outputs

Output	The output checkerboard pattern.

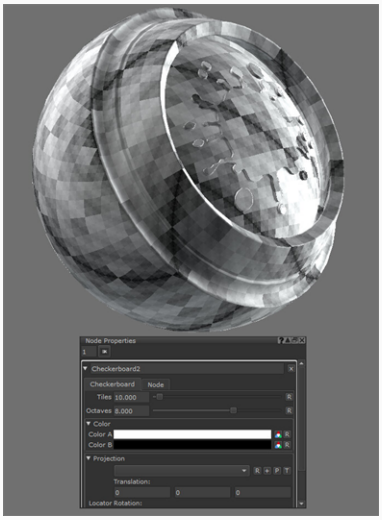
Checkerboard Node Properties

Tiles *text field, slider*	This defines the number of squares that should span one patch.  **Tiles**: 2 **Tiles**: 10  **Tiles**: 40
Octaves *text field, slider*	This defines the number of checkerboards to draw. Each successive iteration will have double the tiles of the one before, and will be slightly more transparent.

Octaves: 2

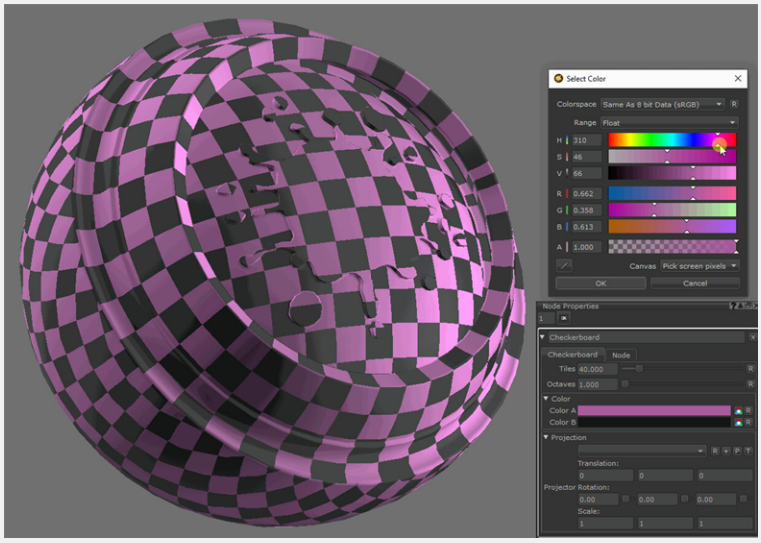


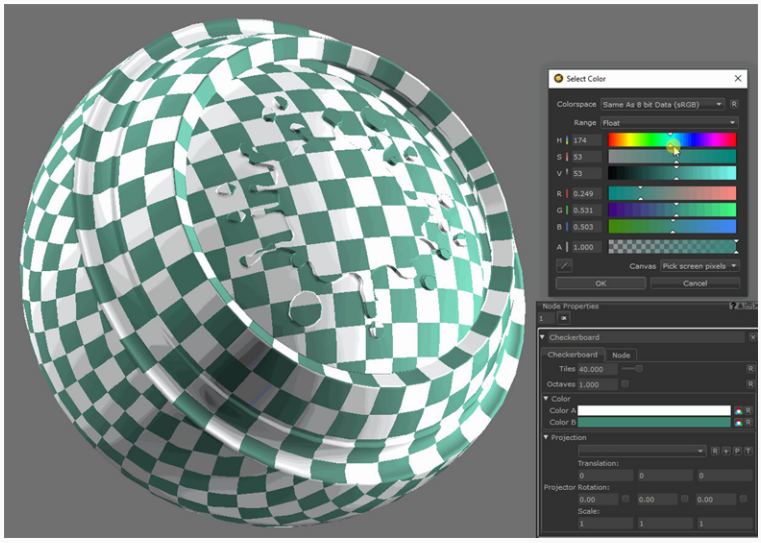
Octaves: 4



Octaves: 8

Color

Color A *Color Select*	This defines what color to use for the checkerboard background. 
Color B *Color Select*	This defines what color to use for the checkerboard foreground.



Projection

Projector *dropdown list*	This allows you to select a projector to align and scale the projected checkerboard. The list displays all the Objects, Locators, Cameras and Lights in the scene which can be used as a projector.
+ *button*	Add a new locator and set it as the projector.
P *button*	Make the projector locator current and switch to the Transform Object Tool.
T button	Toggle the visibility of the transform fields.
Translation *text field*	Used to translate the projection in the x, y and z axis. **Note:** The transform fields (**Translation**, **Locator Rotation**, **Scale**) will lock if you have something selected in the field. The transform field values will mirror those of whatever you have selected.

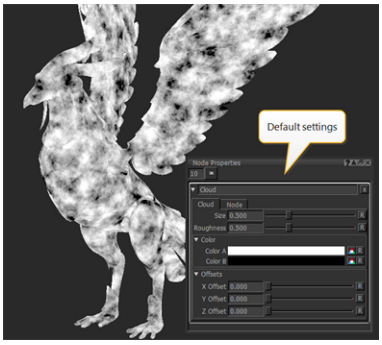
Projector Rotation *text field*	Used to rotate the projection around the x, y and z axis.
Scale *text field*	Used to scale the projection in the x, y and z axis.

Cloud Node

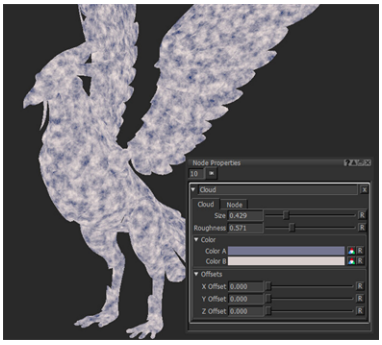
Access: **Nodes** > **Procedural** > **Fractal** > **Cloud**

The Cloud node is a type of noise node that generates random patterns to simulate cloud-like textures. It is commonly used as a mask to control the amount of blending between two nodes.

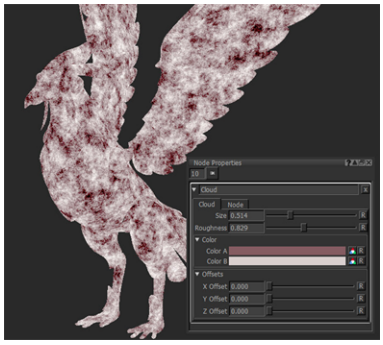
> **Tip:** For example, with a gold and rust textures, you might want to use the Cloud node as a mask to blend both textures, making the gold texture less perfect and therefore more realistic by adding some little imperfections with the rust texture.



Cloud node's default settings



*Cloud node's **Size** decreased, **Roughness** increased, **Color A** blue and **Color B** gray*



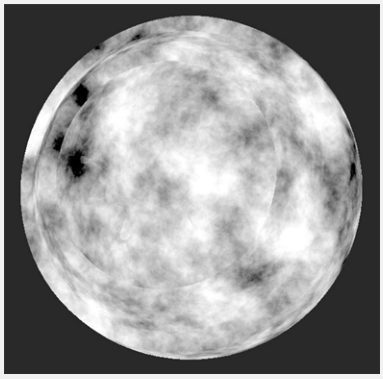
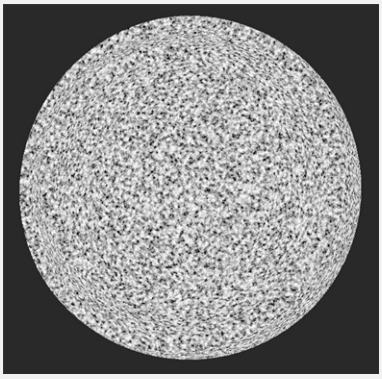
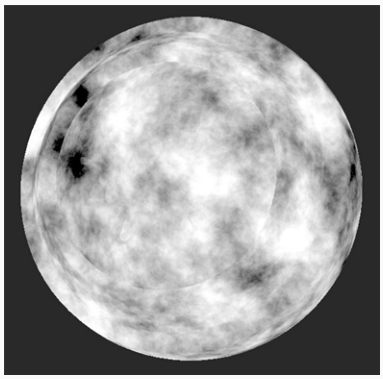
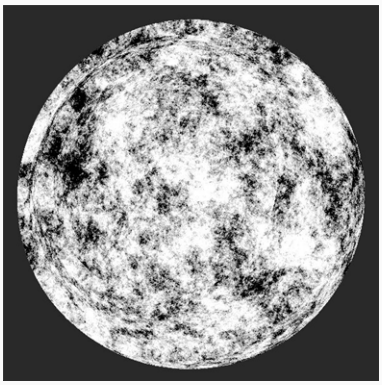
*Cloud node's **Size** increased, **Roughness** increased, **Color A** red and **Color B** gray*

The Cloud node relies on 2D or 3D position information (set of numbers) on the object. Its **Position** input can be connected to any position nodes and you can adjust the position in various ways.

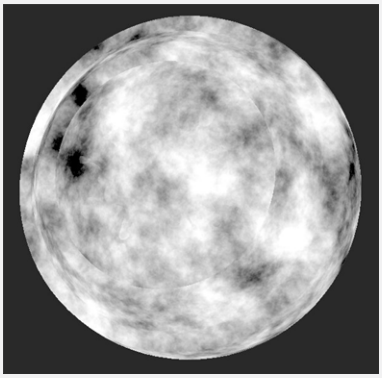
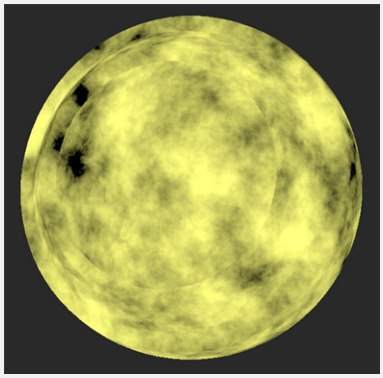
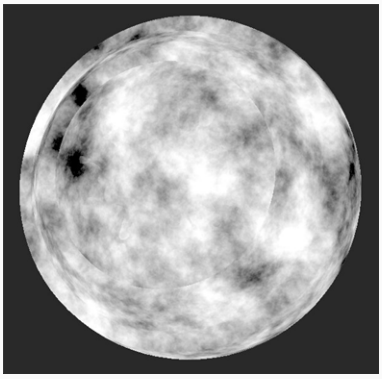
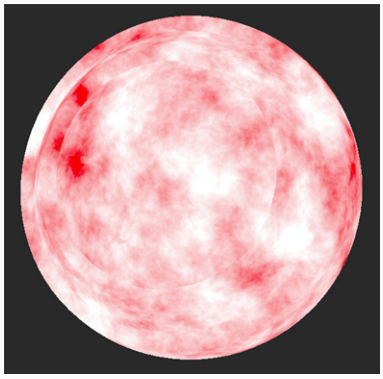
Cloud Node Inputs

Position - Any position data can be connected to the **Position** input, such as the UV and Position nodes. You can use any type of node to modify the mapping of the pattern. For example, you can use the Scale or Vector node in conjunction with the Position or UV node to stretch or offset the position (data) from the Position or UV node.

Cloud Node Properties

Size *floating point control*	Changes the size of the noise.  Default: 0.500  0.190
Roughness *floating point control*	Changes the roughness appearance of the noise, from very smooth at low values to very rough.  Default: 0.500  0.900

Color

Color A *swatch*	Changes the background color.  Default: white  Yellow
Color B *swatch*	Changes the color around the cloud pattern.  Default: black  Red

Offsets

X Offset *floating point control*	Changes the position of the noise on the x axis.
Y Offset *floating point control*	Changes the position of the noise on the y axis.
Z Offset *floating point control*	Changes the position of the noise on the z axis.

Note: For more information on the **Node** tab in the **Node Properties** palette, See Node tab.

Cloud: Node Graph Workflow Examples

Example - Obtaining a Blended Texture Using a Cloud node

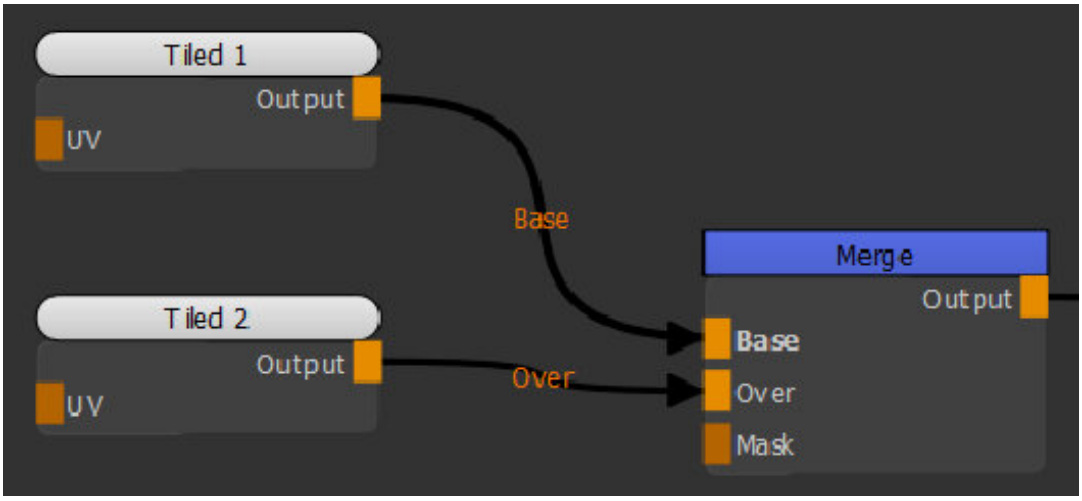
In this example, let's use the Cloud node as a mask on top of two different textures (Tiled 1 and Tiled 2) to obtain a nice blended texture.

These are the visual outputs of Tiled 1 and Tiled 2.

Texture 1 is hidden under Texture 2.

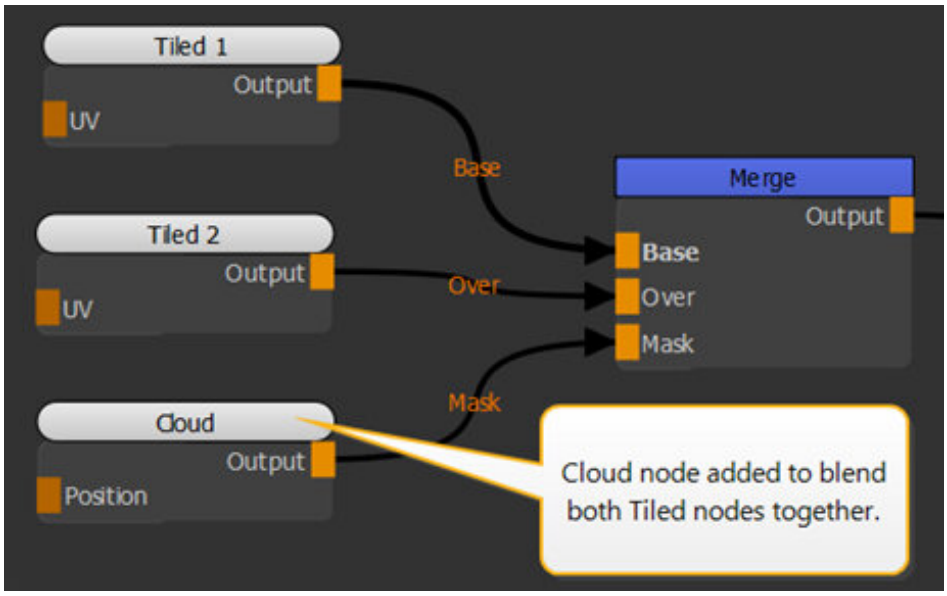
Texture 2 is visible.

1. Connect two Tiled nodes to the Merge node.
   The Tiled 1 node is connected to the **Base** input and the Tiled 2 node is connected to the **Over** input of the Merge node.



Adding two Tiled nodes to the Merge node to achieve a layering operation.

2. Connect a Cloudnode to the **Mask** input of the Merge node.



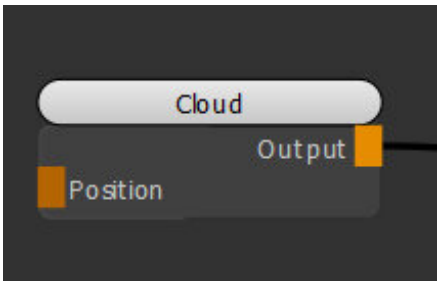
Adding a Cloud node the scene.

Notice how Texture 1 and Texture 2 are both made visible due to the use of a Cloud node applied as a mask.

Example - Switching Mapping from 3D space to UV space

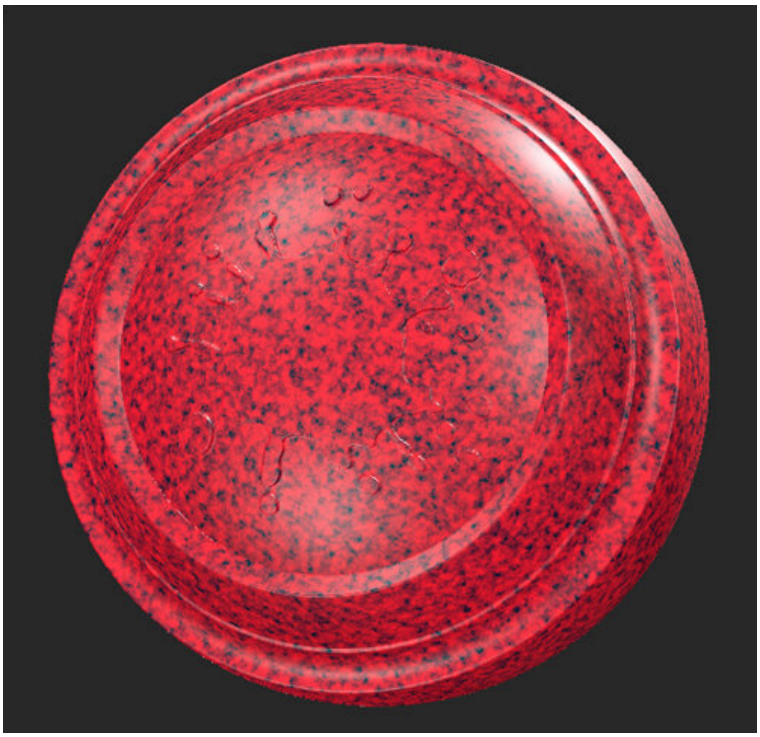
In this example, let's use a UV node to adjust the Cloud noise from 3D space to UV space.

1. Connect a Cloud node.



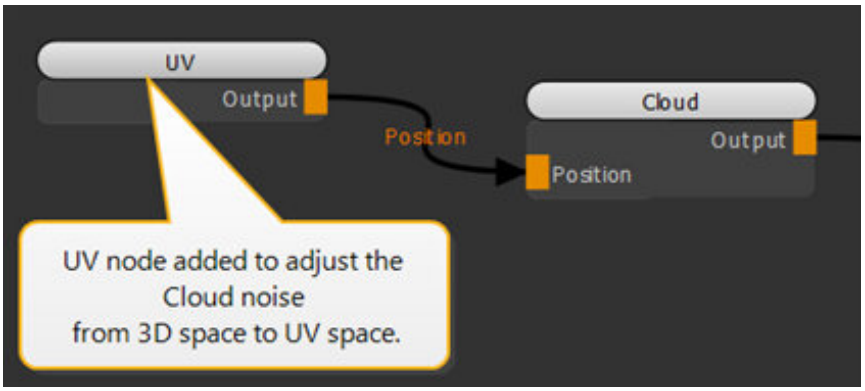
Adding a Cloud node to the scene.

The Cloud node is represented in 3D space. Notice that this method doesn't contain UV seams.



Cloud node represented in 3D space.

2. Create a UV node and connect it to the **Position** input of the Cloud node.

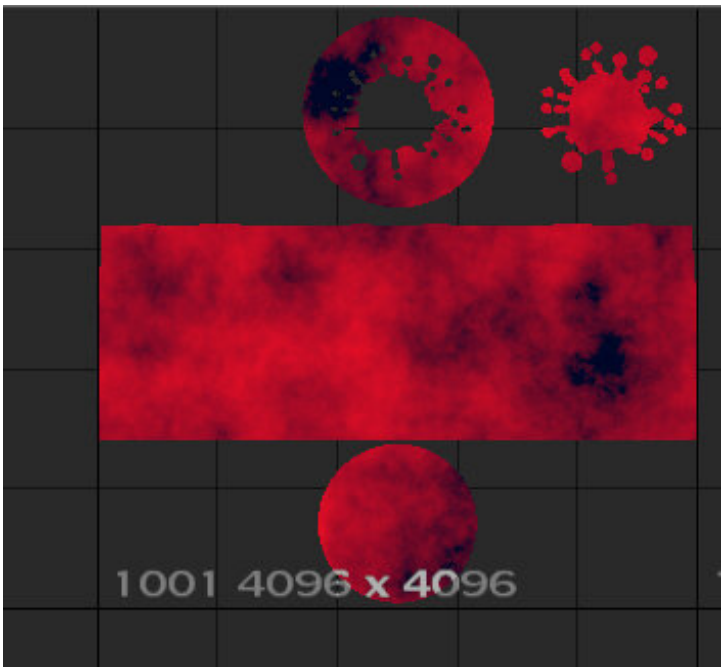


Adding a UV node to the scene by plugging it to the Cloud node.

The Cloud noise is now represented in 2D space.
Notice the UV seam on the right-hand side of the model, which shows the border of a patch.

Cloud node represented in 2D space.

The following image is the UV view of the model to give you a better idea of the noise layout.

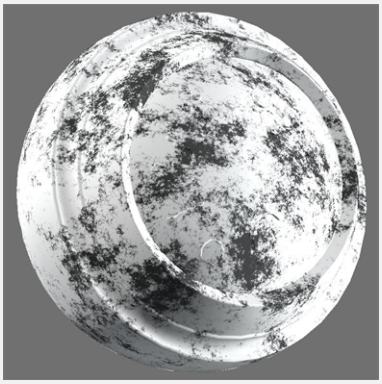


UV view of the model.

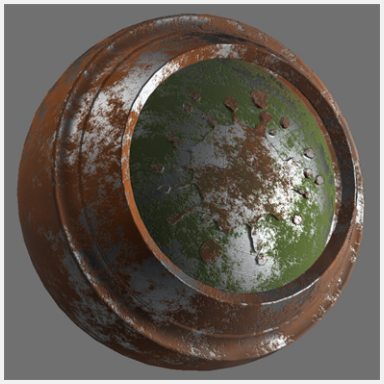
fBm Node

Access: **Nodes** > **Procedural** > **Fractal** > **fBm**

The fBm (Fractal Brownian Motion) node generates a procedural, fractal pattern. This can be used in the same way as any other procedural node to make masks or height maps to create various textures and effects.



fBm Generated and Used as Base Color



fBm Used as a Mask and Height Map

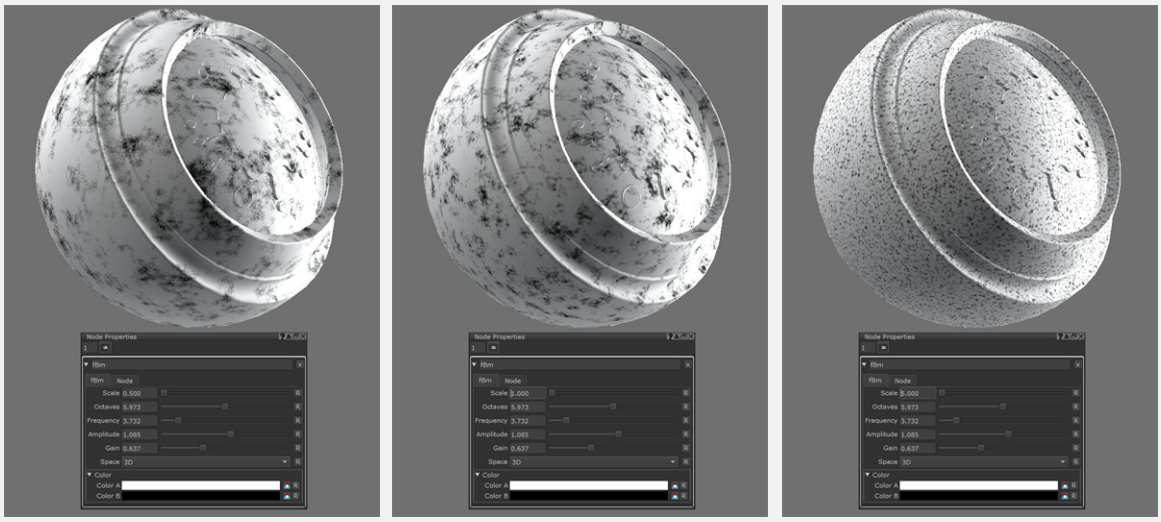
fBm Node Inputs

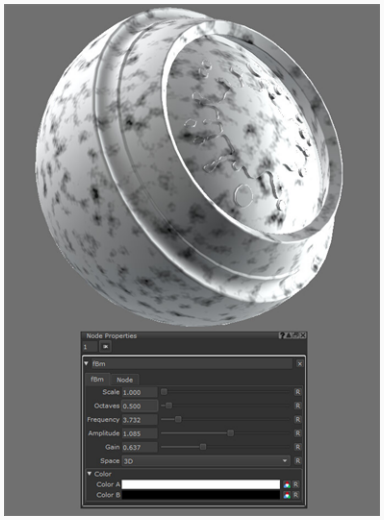
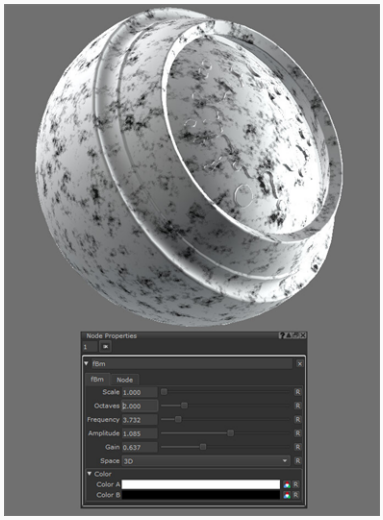
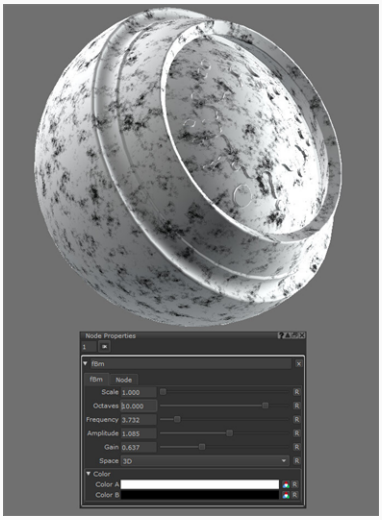
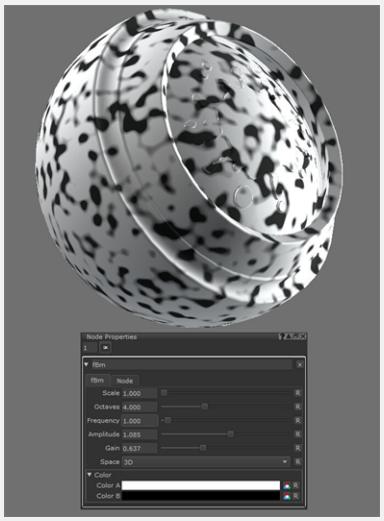
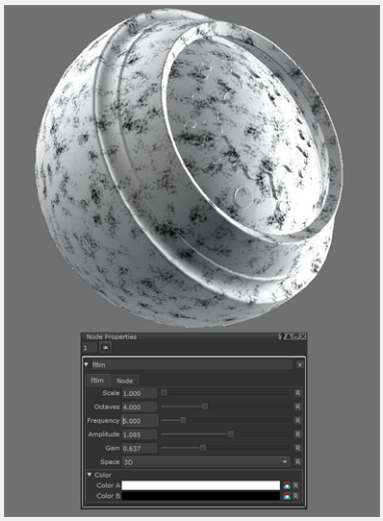
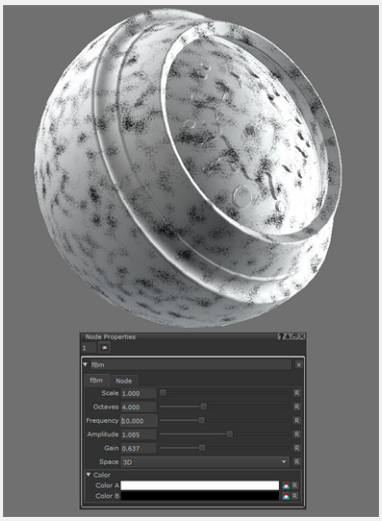
Input	Description
Position	The position for the fBm to be generated at.

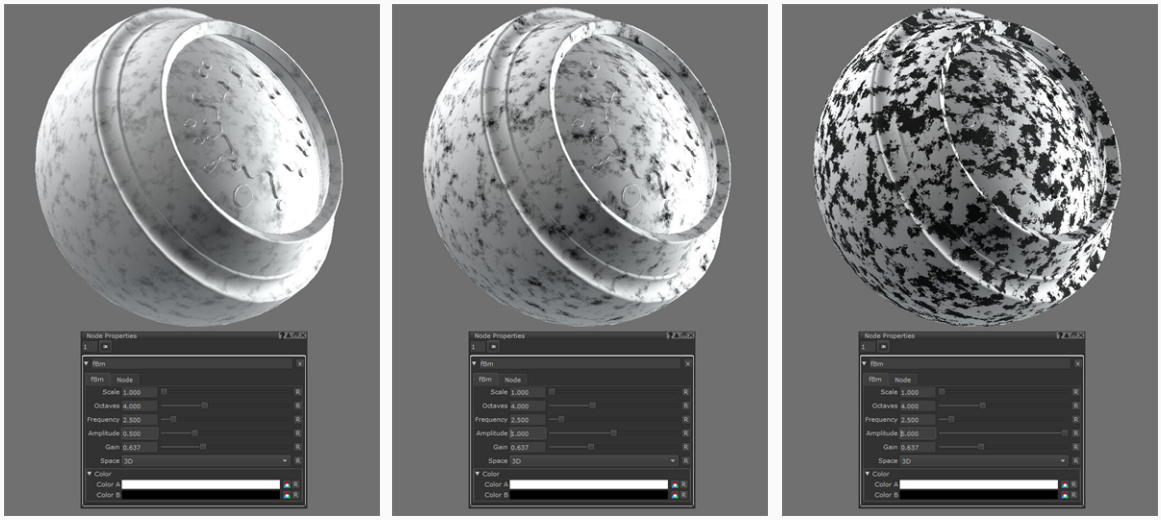
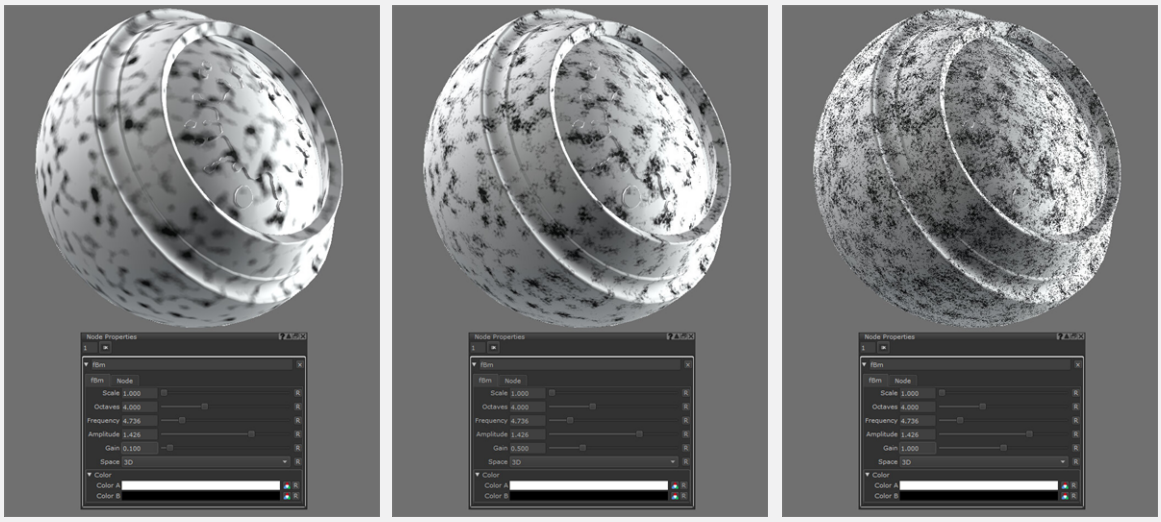
fBm Node Outputs

Output	Description
Output	This outputs the fBm effect.

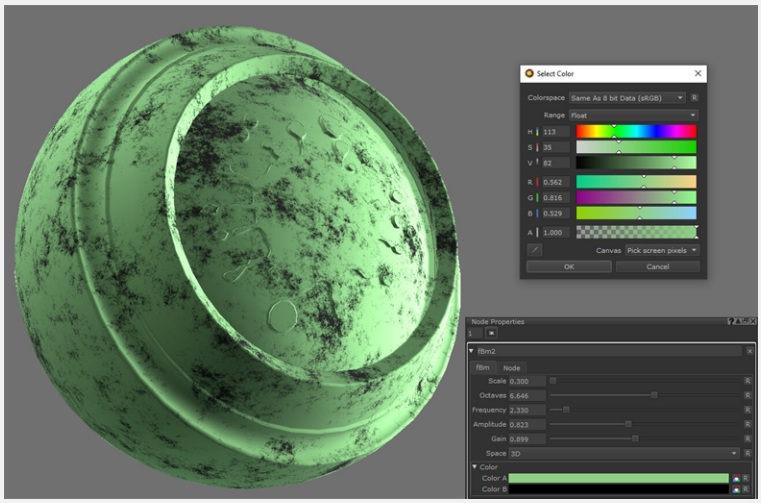
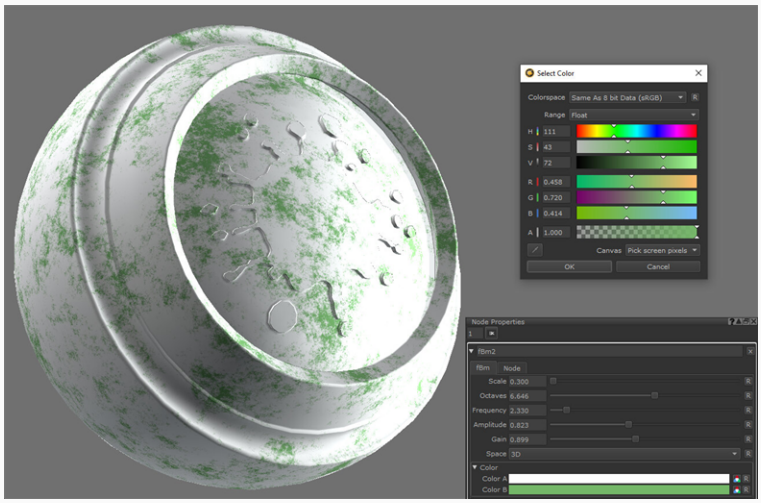
fBm Node Properties

Scale *text field, slider*	This defines the overall scale of the effect.  Scale: 0.5 Scale: 1 Scale: 5
Octaves *text field,*	This controls the number of iterations of the effect.

slider	 Octaves: 0.5  Octaves: 2  Octaves: 10
Frequency *text field, slider*	This defines the turbulence of the resulting effect. A higher value gives a more turbulent effect.  Frequency: 1  Frequency: 5  Frequency: 10
Amplitude *text field, slider*	Low values give a soft gradient. High values push the noise to the extremes.

	 Amplitude: 0.5 Amplitude: 1 Amplitude: 5
Gain *text field, slider*	The Amplitude value is multiplied by this value for each iteration.  Gain: 0.1 Gain: 0.5 Gain: 1
Space *dropdown list*	This specifies the space the effect should be applied in, 3D world space or 2D UV space.

Color

Color A *color select*	This defines the base color. 
Color B *color select*	This defines the color of the fBm pattern. 

Hex Tile node

Access: **Nodes** > **Procedural** > **Pattern** > **Hex Tile Node**

The Hex Tile node randomly distributes textures across the UV plane, resulting in a seamless organic look that has no tiling repetition.

This node can be used with both tiled and non-tiled assets, which means even if you have a single non-tiling image, you can create an infinite, natural surface without any visible seams. This node is useful for texturing rock or sand.

Tiled node inputs

- **UV** — The UV plane you are distributing the texture over.
- **Image** —The image you want to project onto the UVs.

Hex Tile node properties

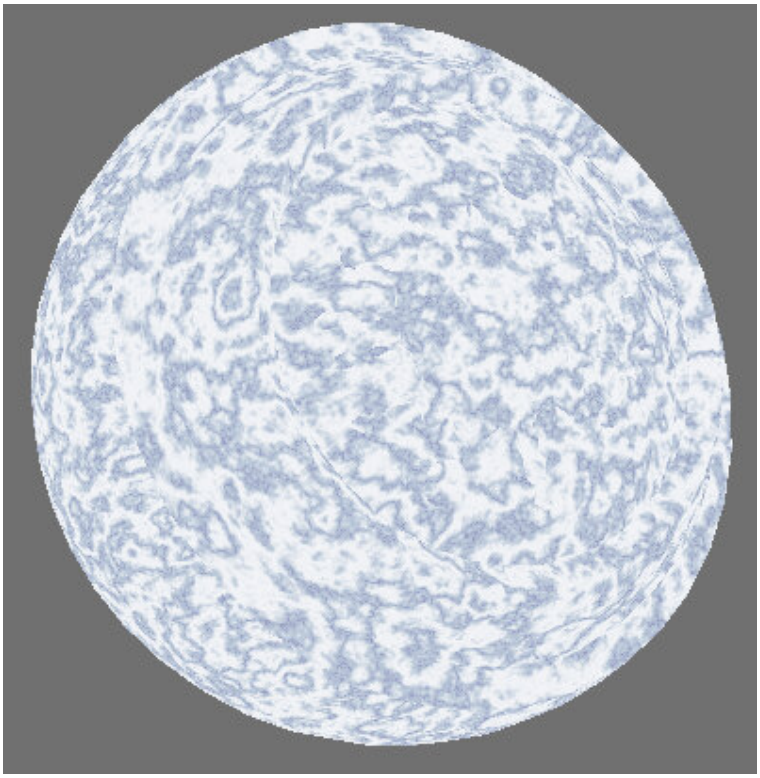
Image	*Image selection*	The input image you want to project onto your object.
Image Does Not Tile	*Checkbox*	Check this if you are not using a tiling image. When this option is selected, Mari prevents sampling from crossing input image boundaries.
Sample Size	*Slider, textbox*	The area of input image to sample for each hex tile.
Normal Map		
Image Is Normal Map	*Checkbox*	If the input image is a normal map, it requires some extra handling. Check this option to enable normal map rotation.
Green Channel	*Dropdown*	Whether the normal map is treated as having a Y+ or Y- green channel.
Blending		
Bleed	*Slider, textbox*	Defines the range of bleeding between hexagons.
Contrast	*Slider, textbox*	Defines the contrast of blending between hexagons.
Rotation		
Rotation	*Slider, textbox*	Initial rotation for the sample of each hex tile

Rotation Variation	*Ranged slider*	Random rotation control on top of the initial rotation

Marble Node

Access: **Nodes** > **Procedural** > **Fractal**

The Marble node is a procedural node which creates a marble-like effect on a surface.



Marble node displayed on an object in Mari.

Marble Node Inputs

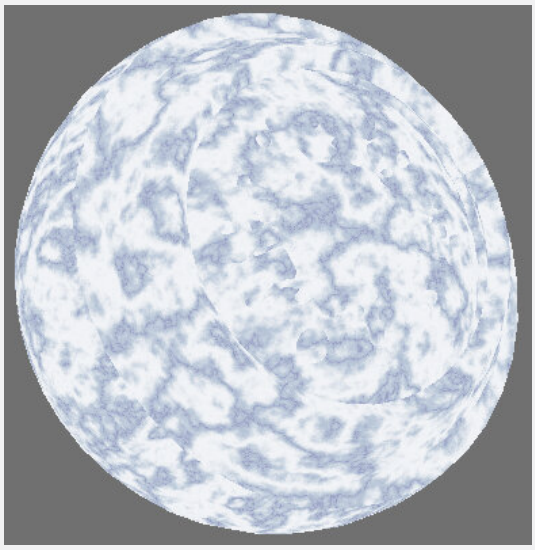
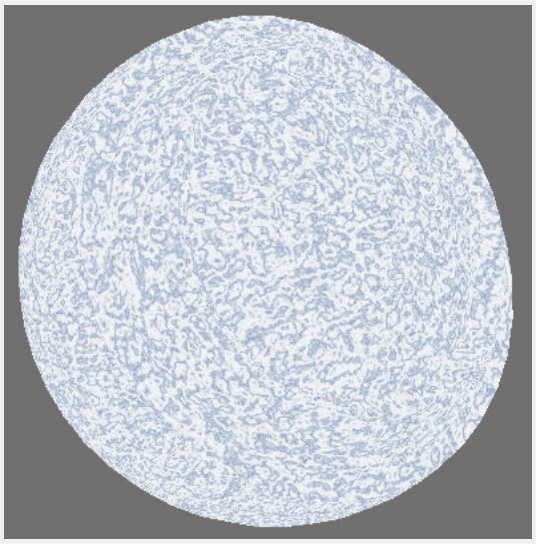
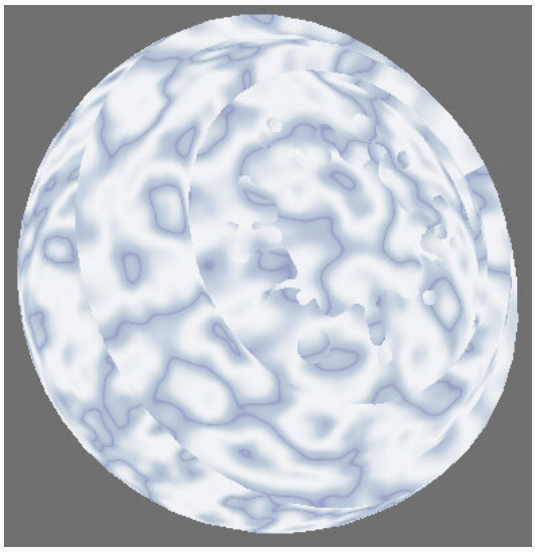
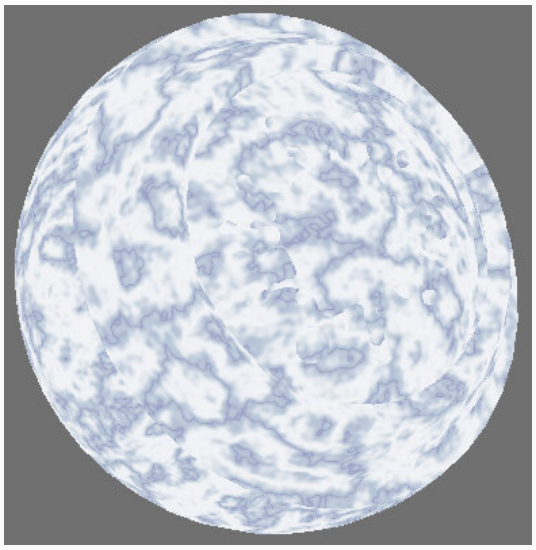
Position: Samples the marble effect at the input position. Defaults to standard surface position when **Space** property is set to **3D** and standard surface UV when **Space** property is set to **UV**.

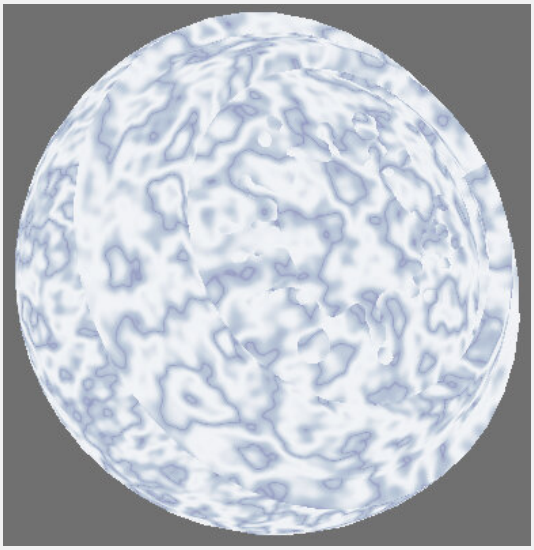
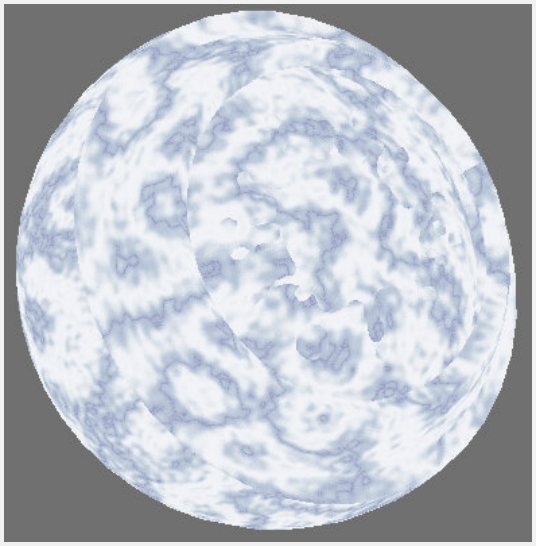
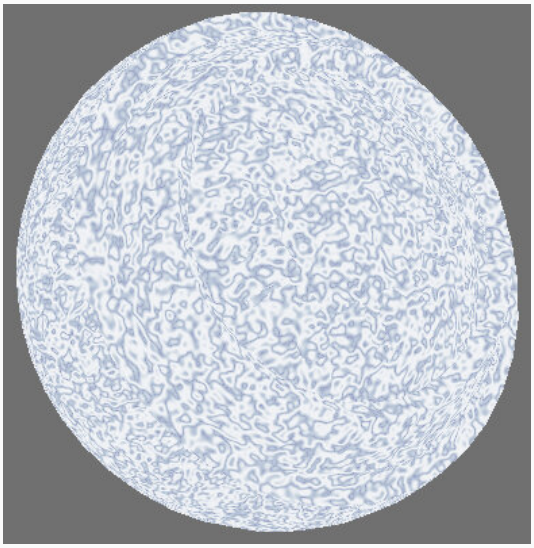
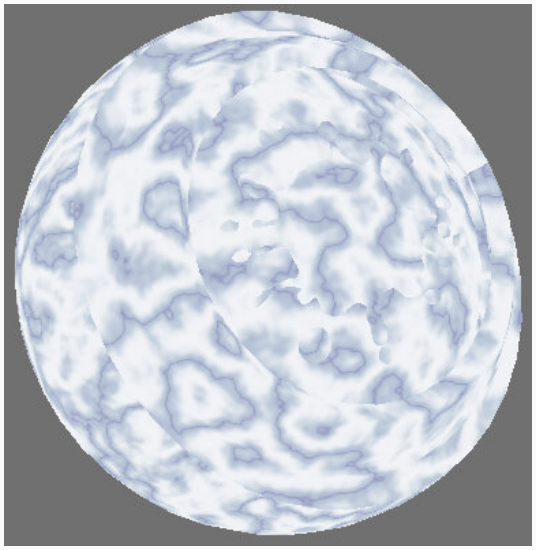
Veins: Uses the input value as the veins.

Bleed: Uses the input value for between the veins and background.

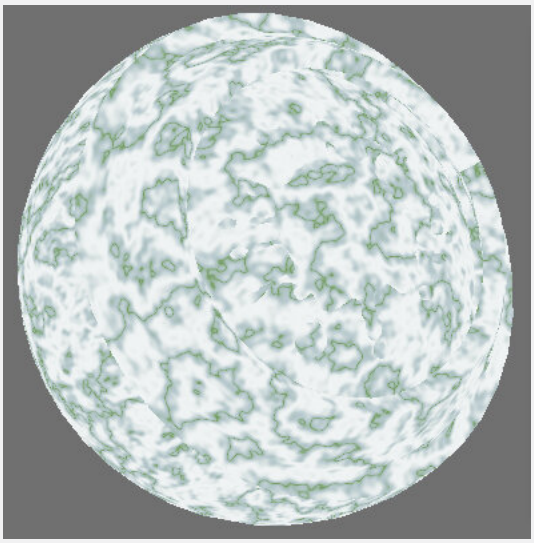
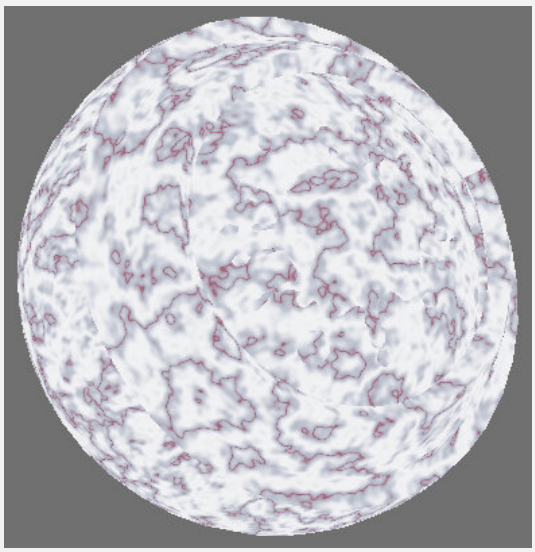
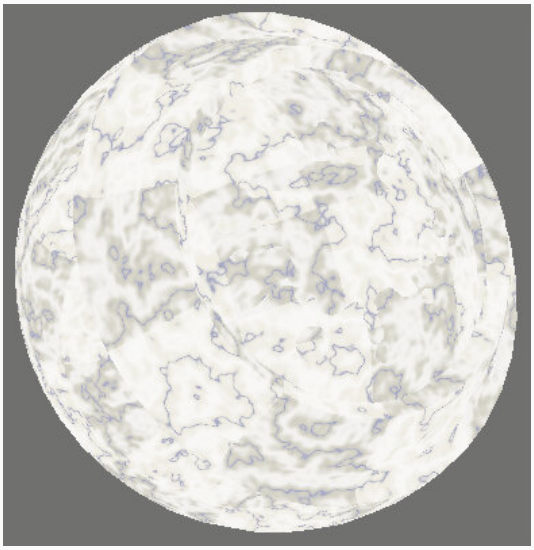
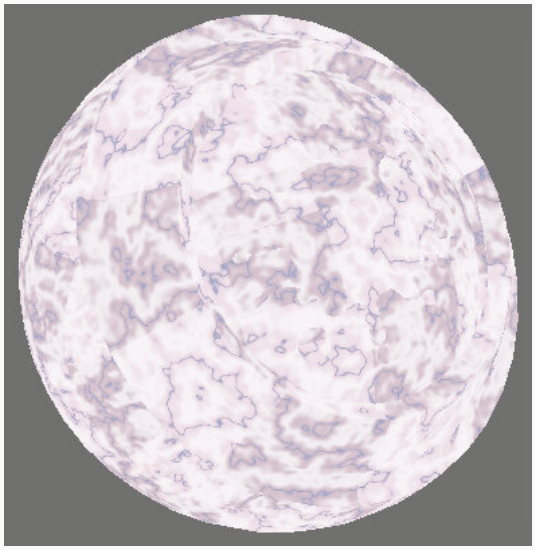
Background: Uses the input value as the background.

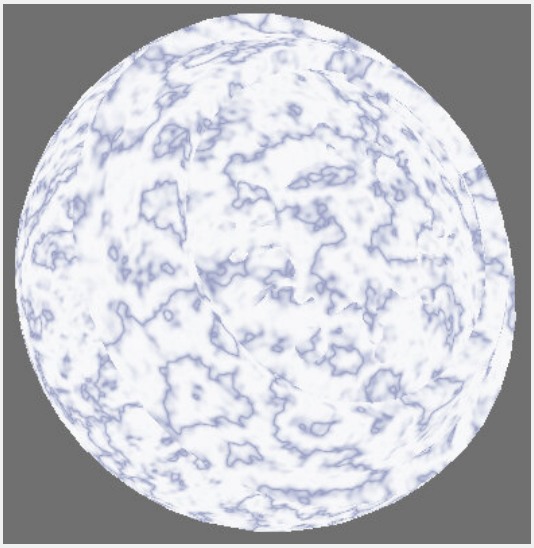
Marble Node Properties

Scale *text field, slider*	Controls the overall scale of the effect.  **Scale** value set to 14.  **Scale** value set to 73.
Octaves *text field, slider*	Controls the density of marble effect lines on the surface. Higher values result in more densely packed lines, lower values result in sparser lines.  **Octaves** value set to 0.5.  **Octaves** value set to 1.9.

Frequency *text field, slider*	Controls the crinkling of marble effect lines on the surface. Higher values result in more crinkled lines.  **Frequency** value set to 2.3.  **Frequency** value set to 7.1.
Increment *text field, slider*	Controls the density of marble effect crinkled lines on the surface. Lower values result in more densely crinkled lines.  **Increment** value set to 0.1.  **Increment** value set to 1.1.
Space *dropdown*	You can specify whether the marble effect should use **3D** space, or **UV** space coordinates.

Color

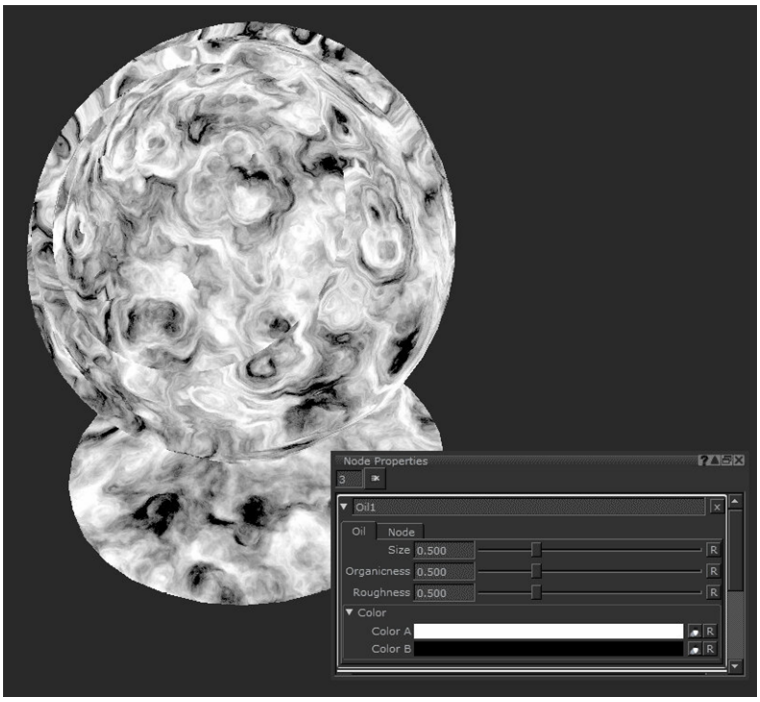
Veins *swatch*	Specify the color to be used for the marble veins.  **Veins** color set to green.  **Veins** color set to purple.
Bleed *swatch*	Specify the color to be used between the veins and background.  **Bleed** color set to yellow.  **Bleed** color set to purple.
Bleed Amount	Specify how much of the **Bleed** color should appear. A lower value shows less bleed, a higher value shows more bleed.

text field, slider	 **Bleed Amount** value set to 0. **Bleed Amount** value set to 1.
Background *swatch*	Specify the default color to be used as the background of the marble effect.

Oil Node

Access: **Nodes** > **Procedural** > **Misc** > **Oil**

The Oil node is a procedural noise node that creates an oil slick style pattern. By changing its inputs you can produce a wide variety of results that you can use to create a range of procedural style textures inside of Mari. It is great for masking, but also emulating more organic style textures that other procedurals aren't quite as successful as matching, such as moss, stone and soil.



Oil node displayed on an object in Mari.

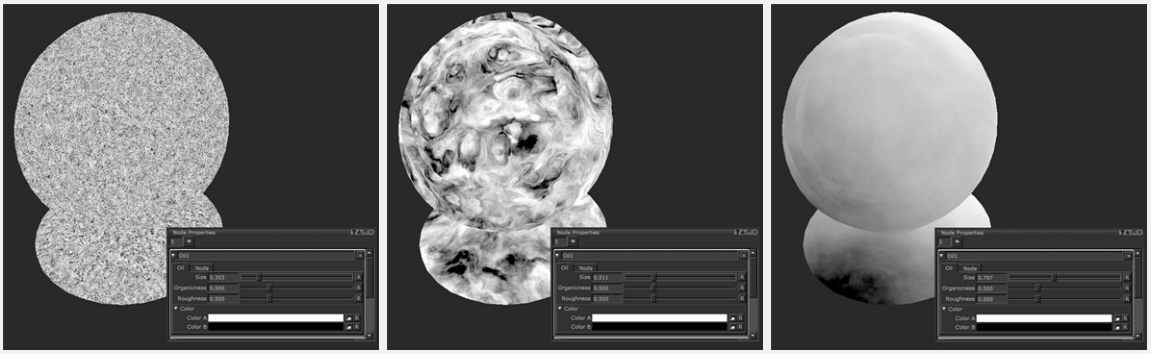
The Oil node is generated using 3D space, but you can manipulate this by plugging a node into its position input.

Oil Node Inputs

Position - The Oil node works in 3D space, but by adding a node to the **Position** input you can manipulate how the Oil noise is applied. For example, you can use the UV node to use a UV space rather than 3D space, or the Position node to get world space position data.

Oil Node Properties

Size *text field, slider*	Changes the pattern frequency of the noise.

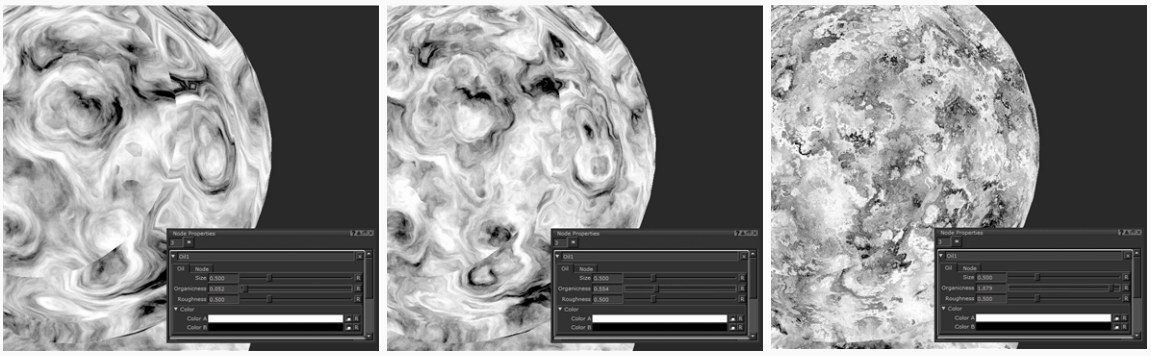


Size of 0.200, 0.500 (default), and 0.700 are shown in these images.

Organicness

text field, slider

Changes the overall flow of the noise, from smoothed and rounded groups to smaller and more fractal-like shapes.

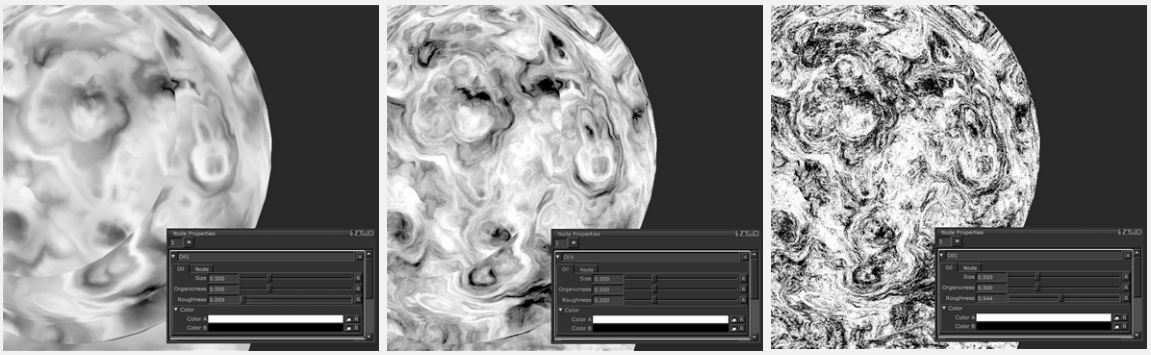


Organicness of 0, 0.500 (default), and 1.8 shown in these images.

Roughness

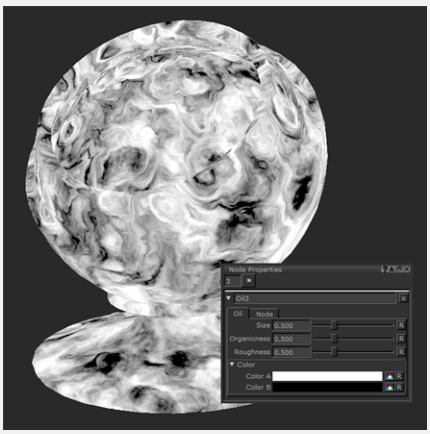
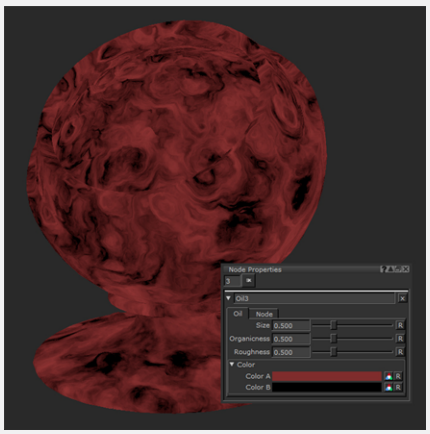
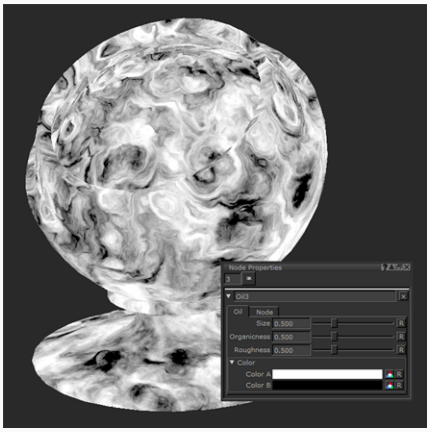
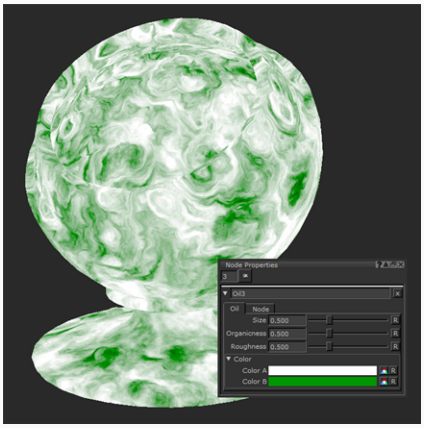
text field, slider

Changes how rough the noise appears, from very smooth and blurred to very rough and sharp.



Roughness of 0, 0.500 (default) and 0.9 are shown in these images.

Color

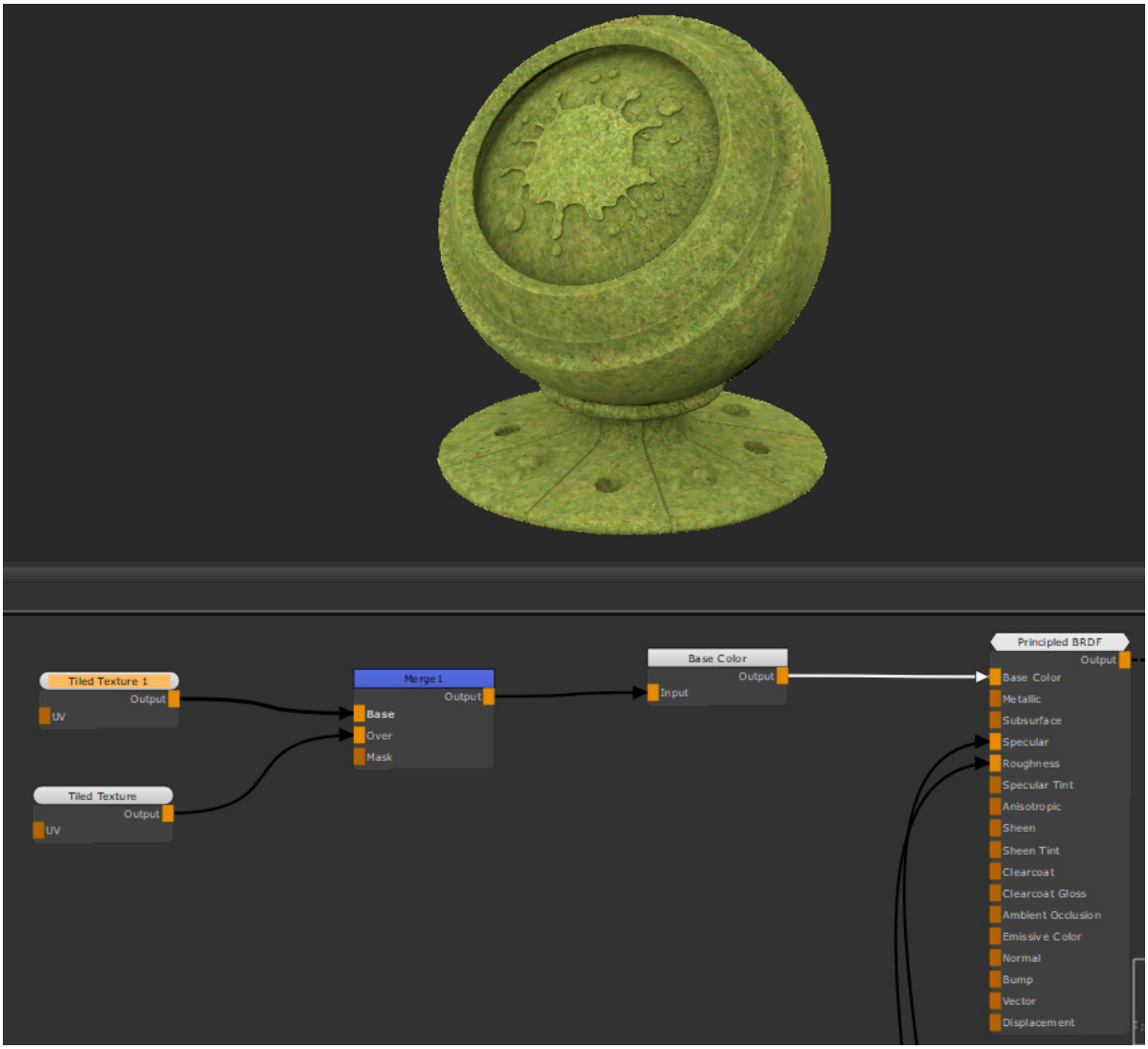
Color A	Replaces any white (value 1, full alpha) values.
Swatch	 Default: **Color A** set to white  **Color A** set to red
Color B	Replaces any black (value 0) values.
Swatch	 Default: **Color B** set to black  **Color B** set to green

Oil Node Workflow Example

Oil Node Masking

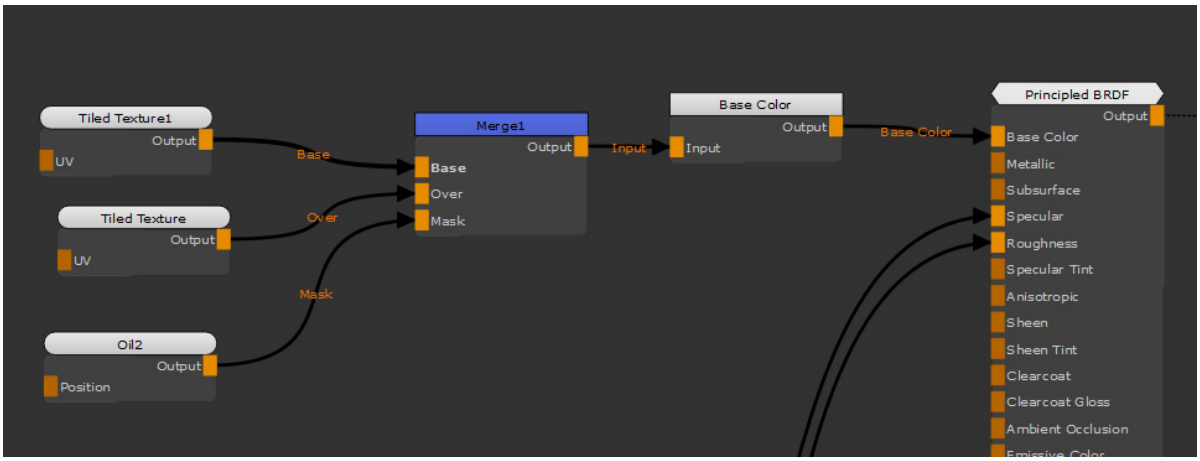
The organic look of the Oil noise node is not often something you can easily achieve with other procedural nodes, and with only three sliders you can get varying results.

Let's look at combining two textures. Here we have two tiled textures, a dirt (**Base** input) and a grass texture (**Over** input), plugged into a Principled BRDF Shader. We want the Tiled (Grass) to merge over the Tiled (Dirt), but without a mask there is no breakup so you only get Tiled (Grass):



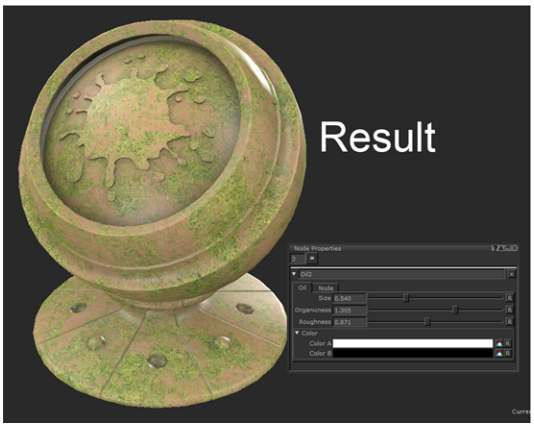
The **Over** texture in our Merge Node (the grass) has no mask so it completely covers the base tiled texture.

Like most noise nodes, the Oil being black and white makes it perfect to use as a mask. Adding an Oil node breaks up the grass texture which is currently everywhere, allowing the dirt to show through in the black areas of the mask.



Here we have two Tiled textures, dirt and grass, being merged together and being used as the **Base Color** of this **Principled BRDF** shader.

By adjusting the sliders it makes a great base for masking dirt or mud in your textures. Adjusting the **Organicness** property of the noise works well for these sort of things, but could be manipulated to make less obvious examples: a flaking paint mask, color for a lizard's scales, and many others.



Adjusting the Oil node properties allows you to get the desired result that matches the scale of your scene and how crisp and rough you want the mask to be.

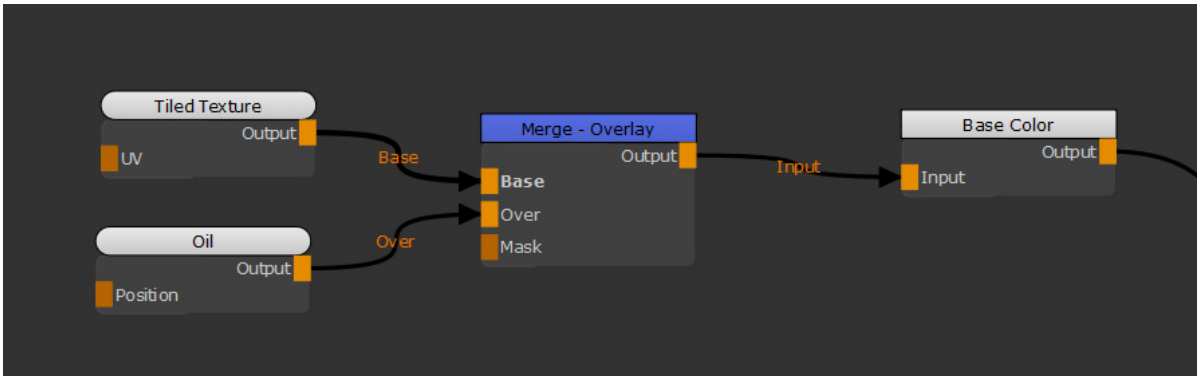
Oil Node Procedural Variation

Another good use of the Oil node is setting up some quick procedural ground textures, or adding details to plain textures.

Here we have just a Tiled texture.

Using an Oil node you can add further variation to the plain color it currently has:

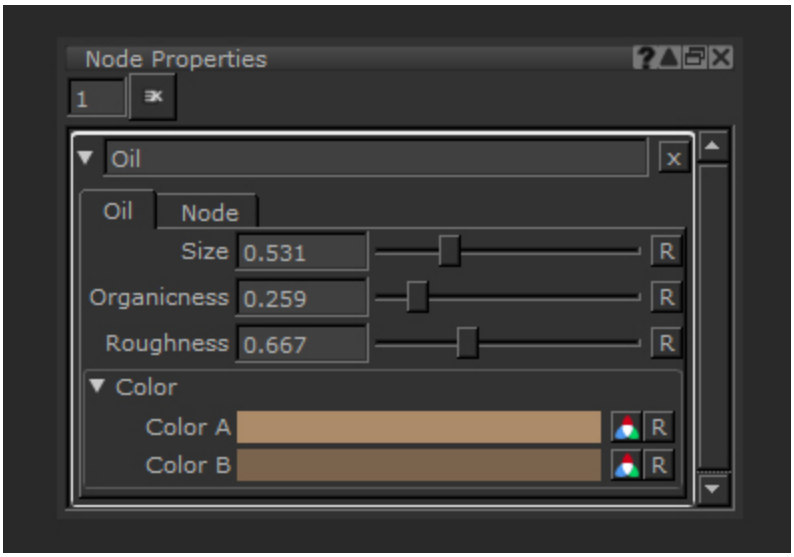
1. Add an Oil node.
2. Add a Merge node and set the **Mode** to **Contrast** > **Overlay**.
3. Plug the Oil node into the **Merge - Overlay** node **Over** input.



A plain texture with an Oil node merged on top, set to blend mode **Overlay**.

By changing the **Roughness** to a higher value, you can get a similar look to certain stones and marbles quickly.

Changing the **Color** properties of the node, or merging it with pre-existing images and projections you can produce variance and breakup. You can also create a mask to only have the oil effect in certain places for even more variance.



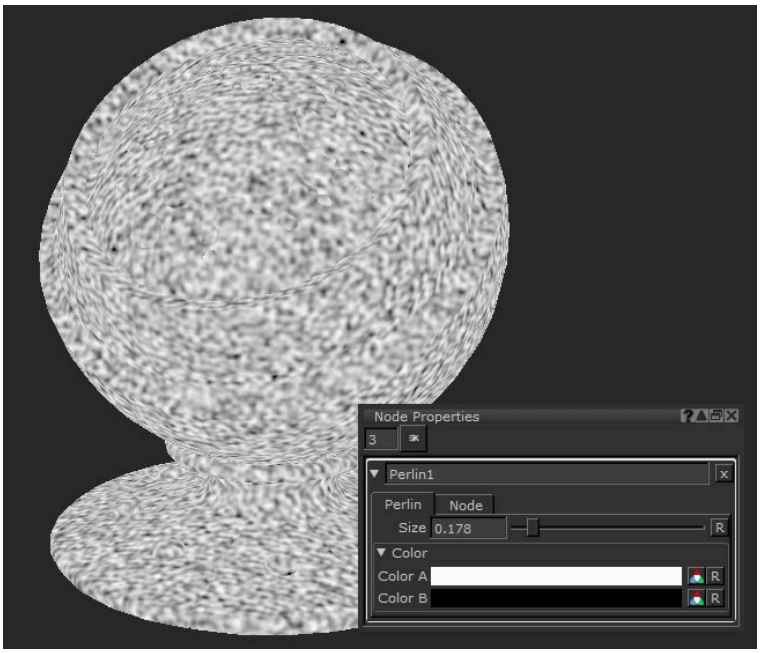
This is the final result after adjusting the values of the Oil node.

You can quickly get texture variation by combing the Oil node on top of your tiled textures.

Perlin Node

Access: **Nodes** > **Procedural** > **Noise** > **Perlin**

The Perlin node is a procedural noise node that creates a basic two color soft and random texture. The Perlin node is great for masking or adding variety when paired with a Merge node.



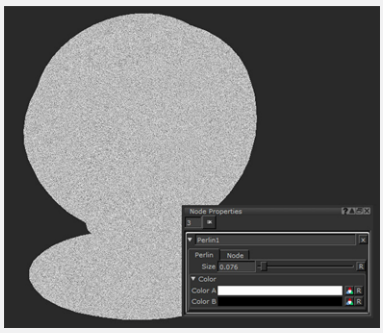
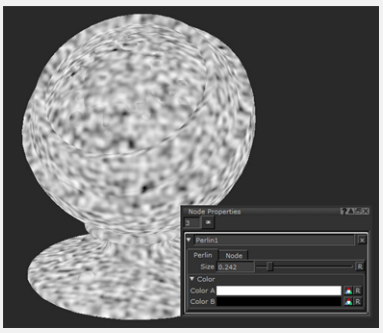
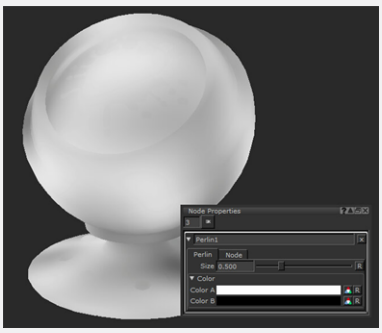
Perlin node displayed on an object in Mari.

The Perlin node generates noise in 3D space, but you can manipulate this by entering an input into its **Position** input.

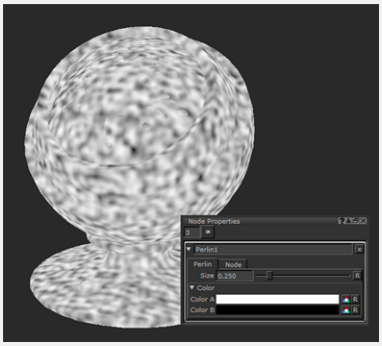
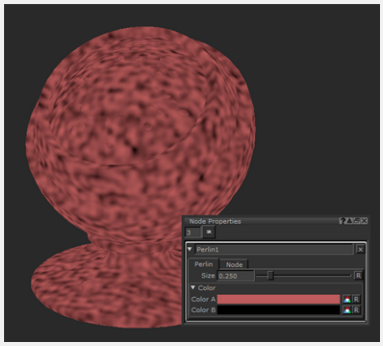
Perlin Node Inputs

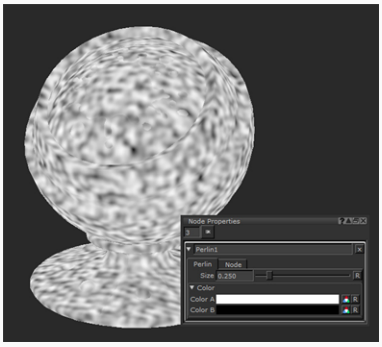
Position - The node works in the 3D space, but by adding a node to the **Position** input you can manipulate how the Perlin noise is applied. Some nodes can be used to manipulate the Perlin noise. For example, the UV node allows you to use an object's UV space rather than 3D space. You could also use the Position node.

Perlin Node Properties

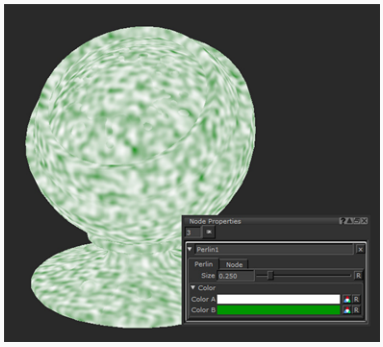
Size *text field, slider* Default: 0.500	Changes the pattern frequency of the noise.  Perlin **Size** value at 0.  Perlin **Size** value at 0.2.  Perlin **Size** value at 0.5.

Color

Color A *swatch*	Changes the first color.  Default : **Color A** set to white.  **Color A** set to red.
Color B *swatch*	Changes the first color.



Default : **Color B** set to white.



Color B set to green.

Perlin: Node Graph Workflow Example

Perlin Noise Mask Breakup

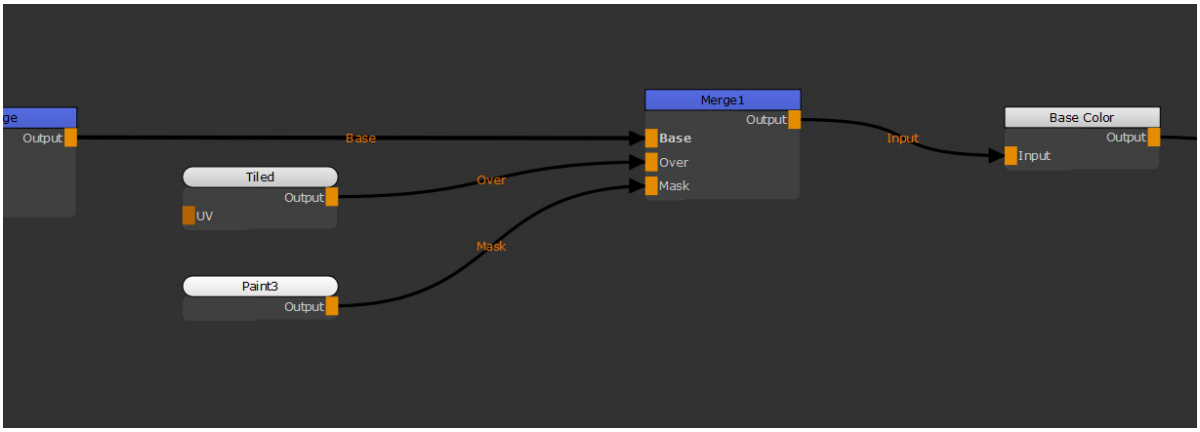
This example uses a Perlin node with an existing mask to add a dirt effect. The Mari noise nodes are great ways to create quick masks, or add breakup to pre-existing masks.

This is the existing setup: some simple block colours shown through the flat **Base Color** and the Shaded view through a BRDF shader.

Adding Dirt to the Asset

1. Add a Merge node with default settings.

2. Add another texture, which in this example is a Tiled node with a sandy texture plugged into the **Over** input
3. Add a mask to define where the dirt will go.
4. Feed the Merge node into the **Base Color** input.

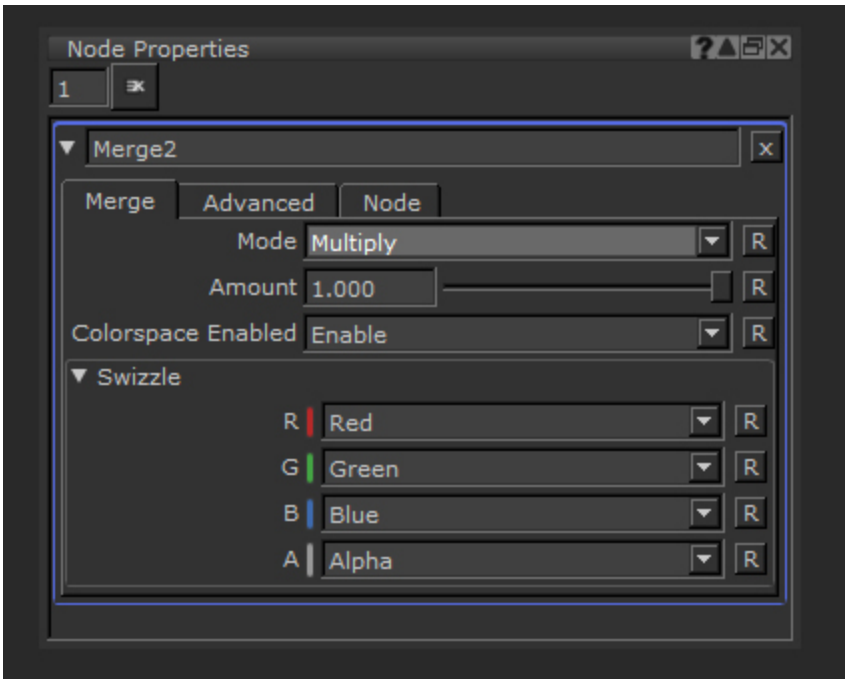


Our initial Node Graph setup for merging in a dirt texture.

The result of feeding the Merge node into the **Base Color**.

Adding Color Breakup Without Hand Painting

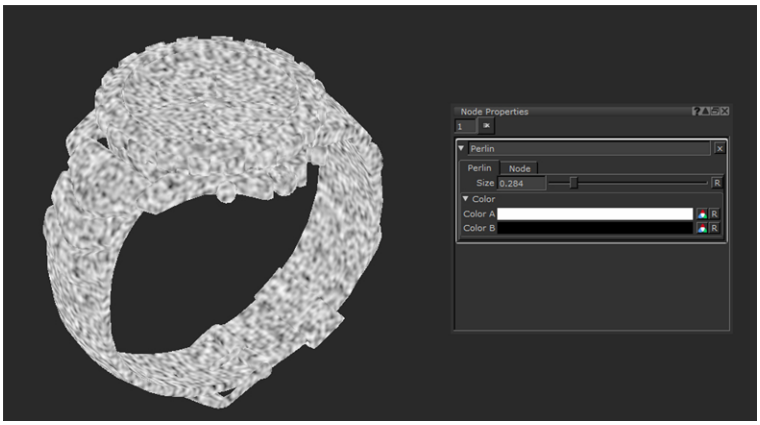
You can multiply a noise node into the node tree of your mask to add some dirt to your asset without hand painting. By using a Merge node you can add extra detail to your textures by using a mask input.



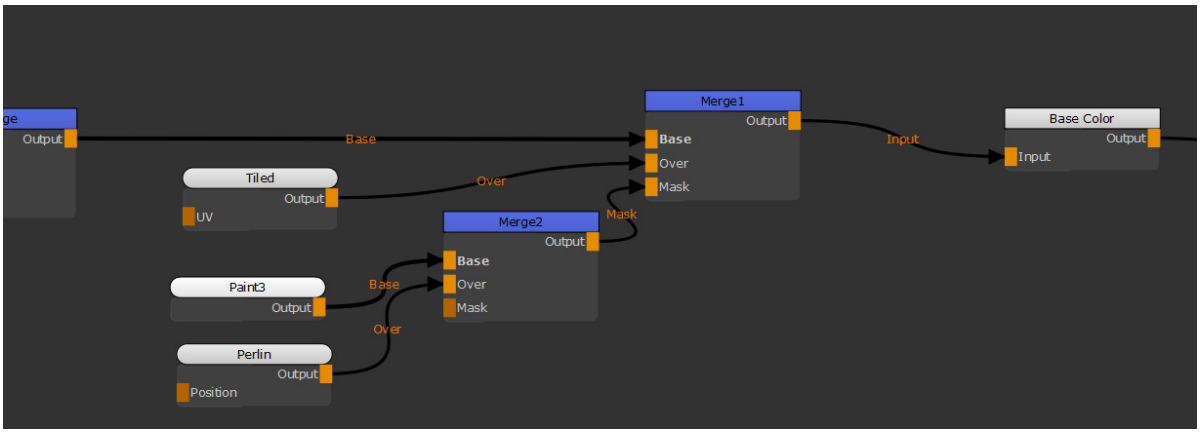
The Merge node properties.

Using the **Multiply** mode takes the dark values of the Perlin node and blends those values into your dirt mask. You can adjust the blend **Amount** property to change how much the blend is multiplied by.

The multiplication erodes the node that is added to the base input to break it up and make it look dirtier.



Viewing the Perlin node on the model.



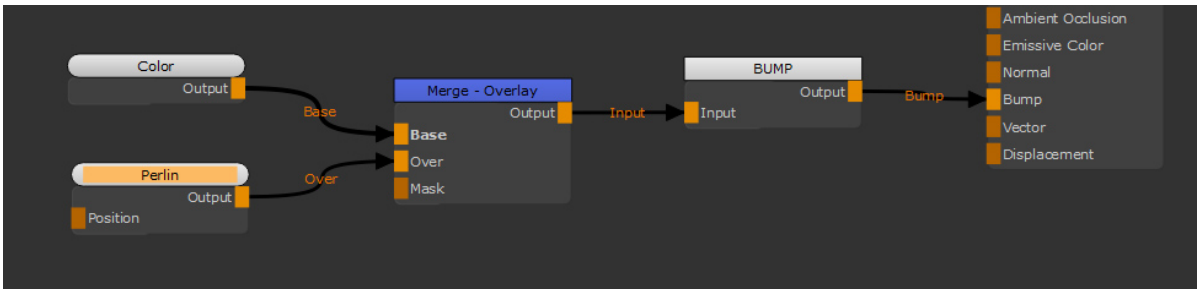
Using a second Merge node on the mask to add Perlin breakup.

Tip: You can also experiment with the other blend modes, like **Overlay** or **Add** to get different effects for different situations.

The results of applying Perlin.

Perlin Noise Channel Breakup

Another good use of the Perlin node and other noise nodes is applying break up to different texture maps. If you have a very flat looking colored metal, or a bump map that needs a bit of undulation, using a Perlin noise overlay can add breakup in lots of different ways.



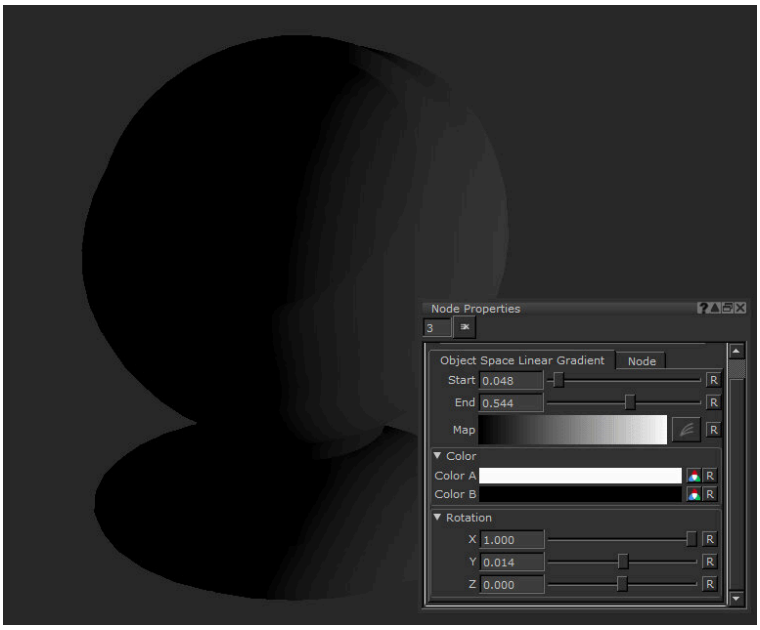
Here we have a flat color with a Perlin node being merged on top with the **Overlay** mode. This feeds into the **Bump** Channel node which leads into a Principled BRDF Shader.

You can plug a Perlin node into the **Over** input on the Merge node, which allows the overlay's dark and light values to show through on what you already have. This allows you to add some variety in the bump of the shader.

Making sure the **Size** is relative to the object is important to sell the scale. Perlin and other noise can be used to quickly build up a procedural node tree inside Mari without having to import images to use or hand paint.

Object Space Linear Gradient Node

Access: **Nodes** > **Procedural** > **Pattern** > **Object Space Linear Gradient**



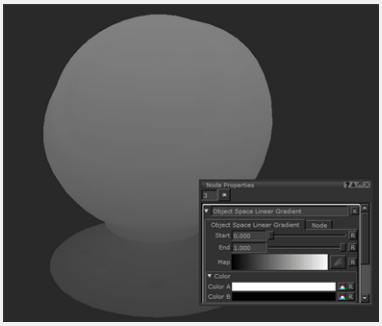
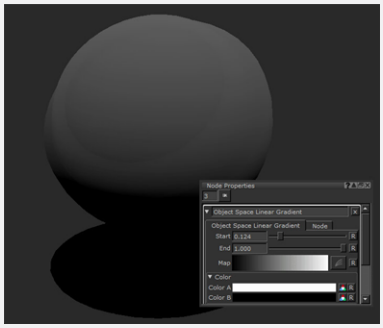
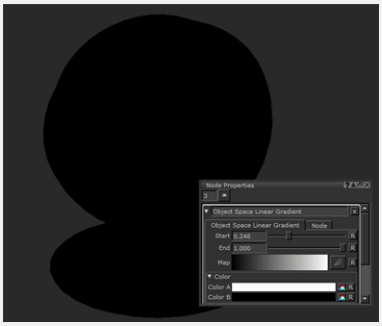
A rotation applied to a Object Space Linear Gradient node on the geometry.

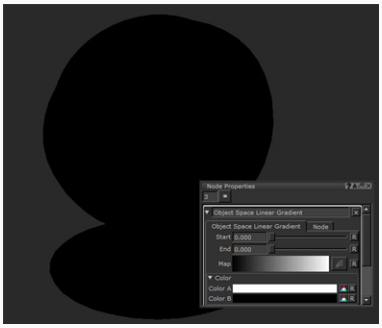
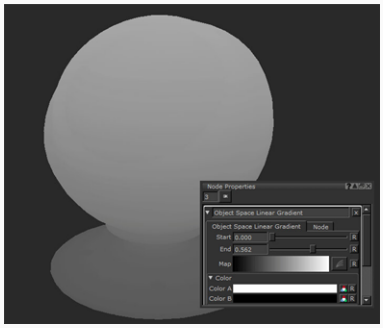
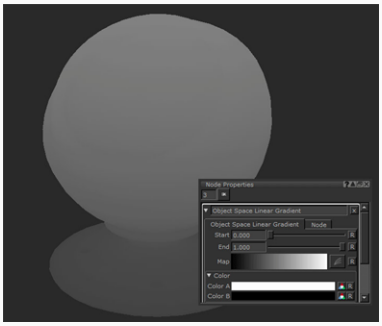
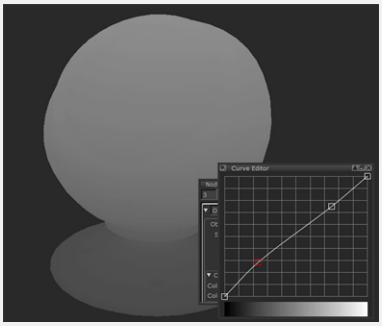
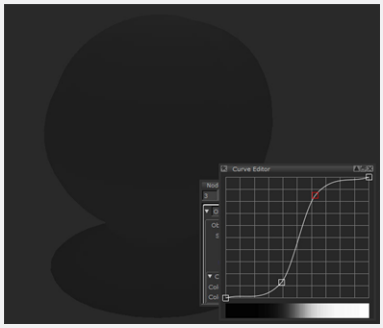
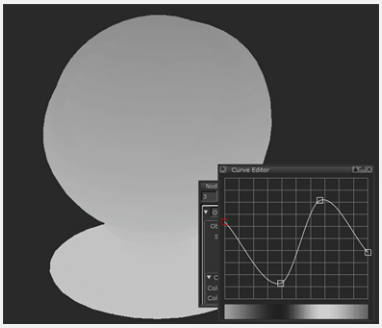
The Object Space Linear Gradient node is a great way to applied graduated colors across your objects geometry.

The node gives you a lot of options to change the result and its great for creating quick masks based on the world space of your object. By editing the properties you can make looser or tighter gradients and with the provided ramp options you can add extra levels of fidelity to the output.

The Object Space Linear Gradient node is a great way to mask off just the bottom of a mesh, for example for adding mud around the base, or for adding a color change over the mesh without having to try paint a smooth gradient yourself.

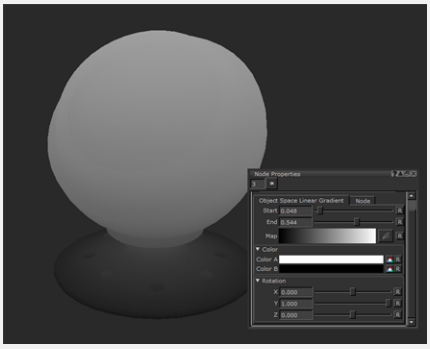
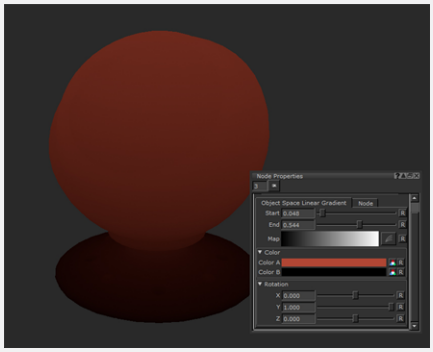
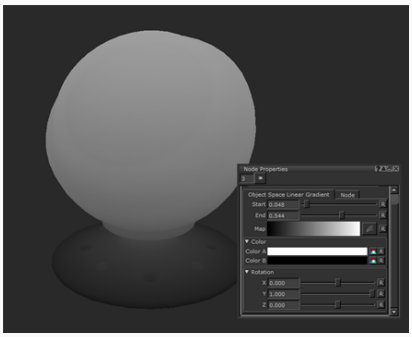
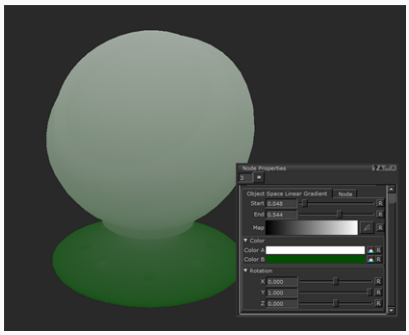
Object Space Linear Gradient Node Properties

Start *text field, slider*	Changes the world space location of the start of the gradient. A value of 0 is the edge of your mesh's dimensions. Increasing it will cause the start point to move further along your mesh. The slider will cap at 0 but you can type in negative numbers to push it further away, stretching out the gradation.  **Start** value set to 0.  **Start** value set to 0.1.  **Start** value set to 0.2. Default is 0.
End *text field, slider*	Changes the world space location of the end of the gradient. 1 is the edge of your mesh's dimensions. Decreasing it will cause the start point to move further along your mesh. The slider will max at 1 but you can type in larger numbers to push it further away, stretching out the gradation.

	 **End** value set to 0.  **End** value set to 0.5.  **End** value set to 1. Default is 1.
Map *Curve editor*	By clicking the curve editor icon you can change the curve of the gradient instead of it being a linear 0 to 1 ramp.    Default is a black to white linear gradient.

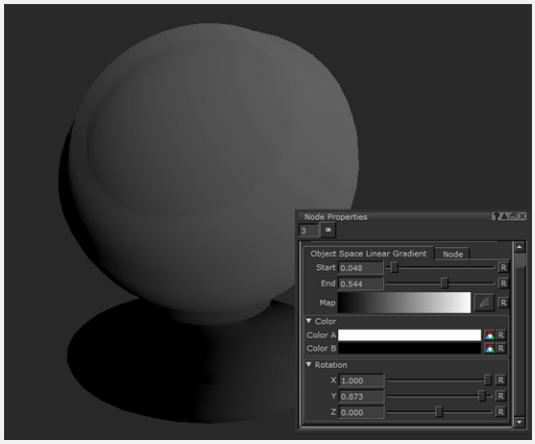
Color

Color A *Swatch*	Changes the start color of the gradient.

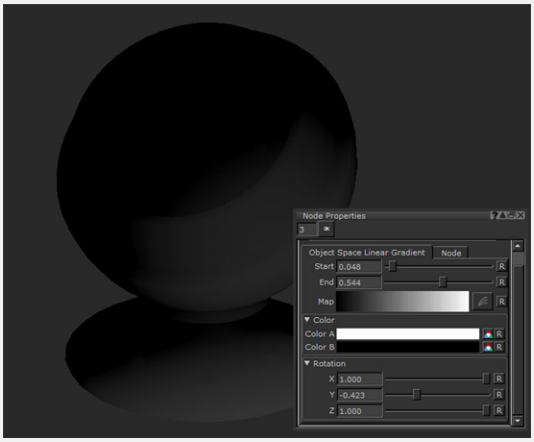
	 Default: **Color A** set to white.	 **Color A** set to red.
Color B *Swatch*	Changes the end color of the gradient.  Default: **Color B** set to black.	 **Color B** set to green.

Rotation

X, Y, Z *text field, slider*	Changes rotation of the ramp along the **X**, **Y** or **Z** axis. Manipulate these sliders to have the gradient go across a different direction in world space.



Rotation X value set to 1, **Rotation Y** value set to 0.8, **Rotation Z** value set to 0.



Rotation X value set to 1, **Rotation Y** value set to -0.4, **Rotation Z** value set to 1.

Default is 1, 0, 0.

Object Space Linear Gradient Node Workflow Example

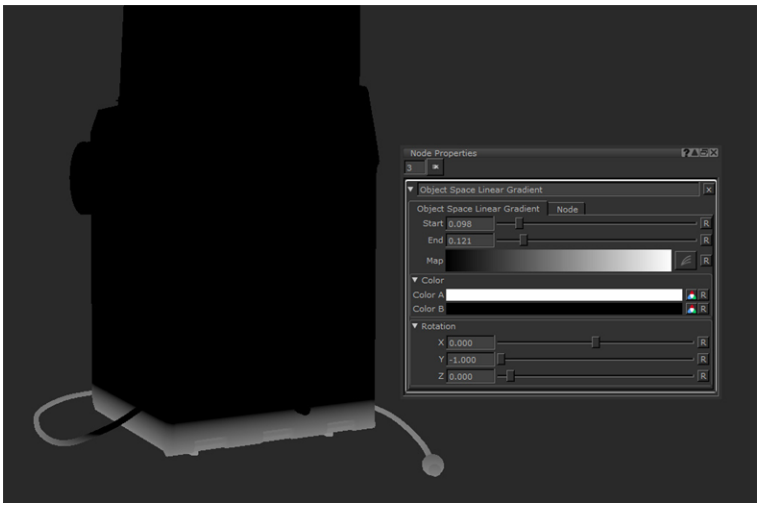
Masking Dirt with Object Space Linear Gradient

Using the Object Space Linear Gradient node we can easily create masks on the object that sit just at the bottom or top of the geo. This is a really quick and easy way to add mud or dirt to the bottom of objects. By combining it with other masks or procedurals you can get procedural dirt masks using just a few nodes.

Here we have a camera object that we are going to create a dirt mask for.

The object has been textured with materials, so we are going to use a dirt material from the **Shelf** palette as a new addition.

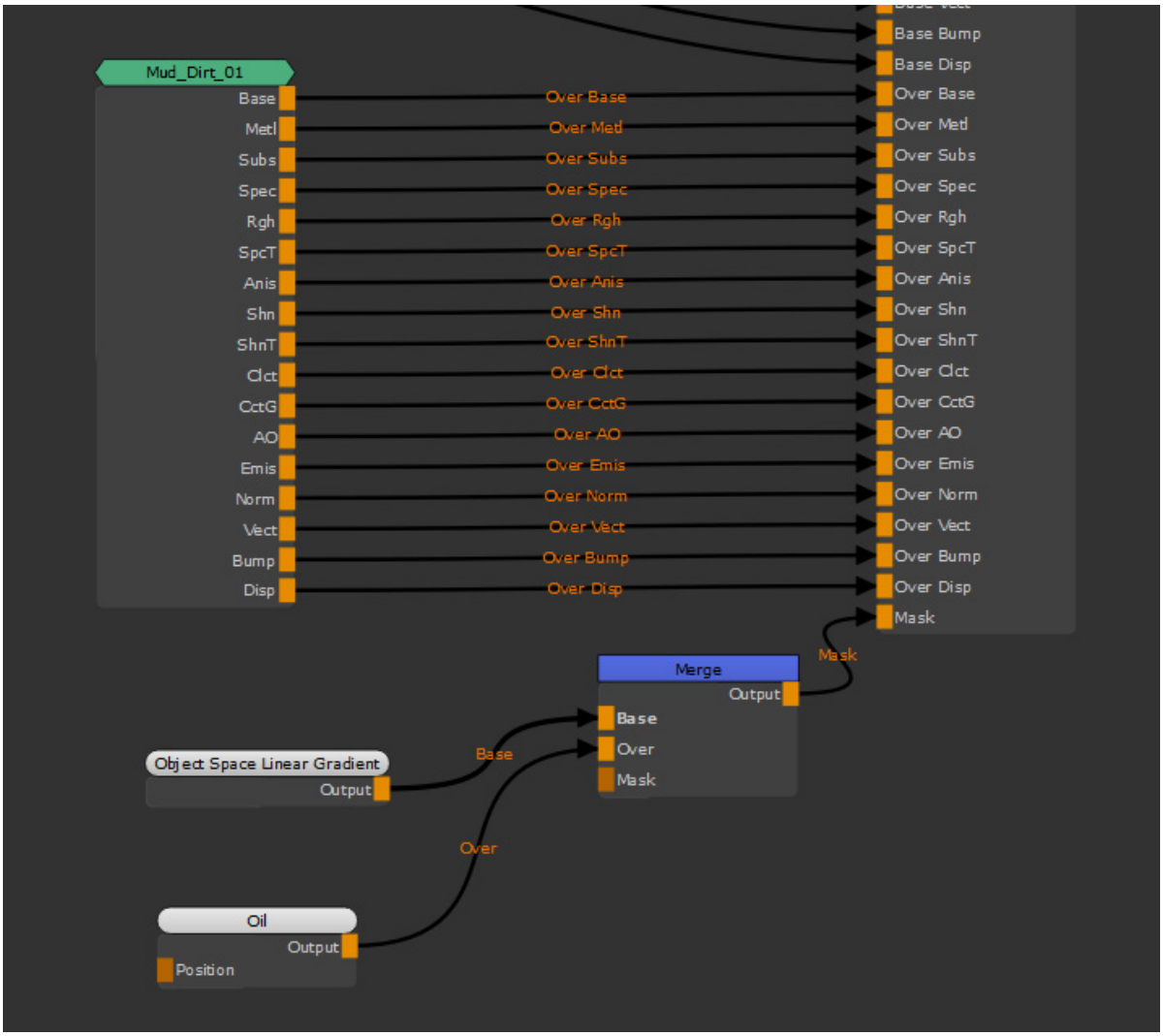
Tip: For more on Material nodes see here.



By dropping down an Object Space Linear Gradient node we can start off the mask construction. First thing is to get the **Rotation** correct. We want it to be along the height of the geometry which is the Y axis, so **X** needs to be 0 and **Y** needs to be -1. Then you can use the start and end sliders to change the tightness of it.

> **Tip:** The reason we use -1 instead of 1 in the **Rotation Y** field is because for a mask you want white to be where the dirt will be so by using -1 it will flip to gradient the correct way around.

Finally using a Merge node to multiply an Oil node on top of this mask allows us to add some organicness to this mask so it looks more realistic.

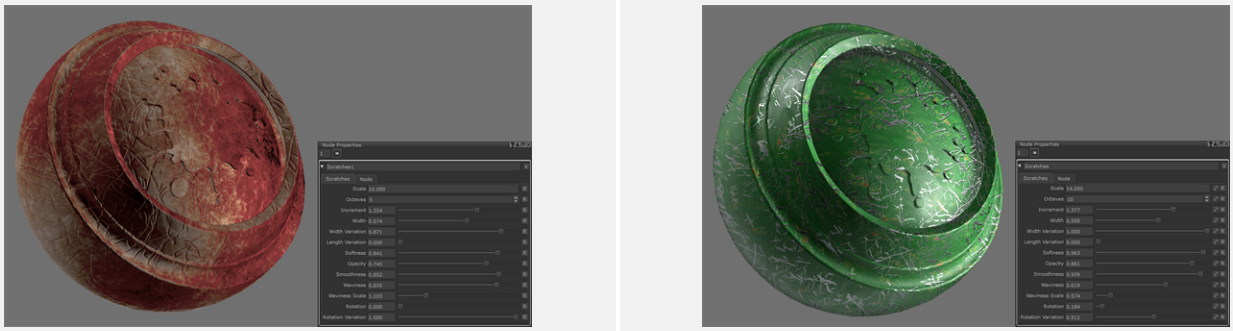

Mud_Dirt_01
Base
Metl
Subs
Spec
Rgh
SpcT
Anis
Shn
ShnT
Clct
CctG
AO
Emis
Norm
Vect
Bump
Disp
Over Base
Over Metl
Over Subs
Over Spec
Over Rgh
Over SpcT
Over Anis
Over Shn
Over ShnT
Over Clct
Over CctG
Over AO
Over Emis
Over Norm
Over Vect
Over Bump
Over Disp
Base Bump
Base Disp
Over Base
Over Metl
Over Subs
Over Spec
Over Rgh
Over SpcT
Over Anis
Over Shn
Over ShnT
Over Clct
Over CctG
Over AO
Over Emis
Over Norm
Over Vect
Over Bump
Over Disp
Mask
Mask
Merge
Output
Base
Over
Mask
Object Space Linear Gradient
Output
Base
Over
Oil
Output
Position

With only a couple of nodes, the Object Space Linear Gradient has helped create a very easy, quick and versatile dirt mask setup in our Node Graph.

Scratches Node

Access: **Nodes** > **Procedural** > **Pattern** > **Scratches**

The Scratches node generates iterations of scratch-like lines in UV space. This node can be used to replicate scratches, creases and general wear on the surface of an object. The scratches node can be used as a height map to indent the pattern, a mask to scratch through to a material underneath, or both.



Scratches used as height map and mask with varying parameters

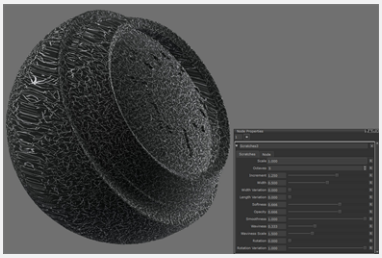
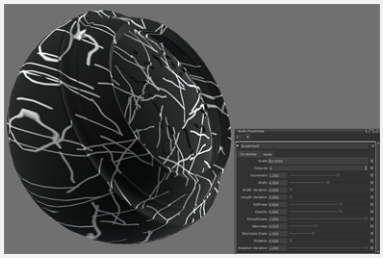
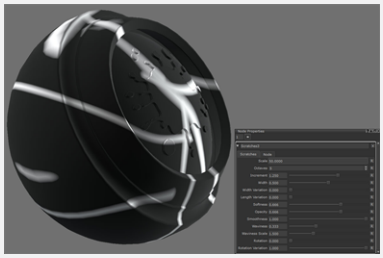
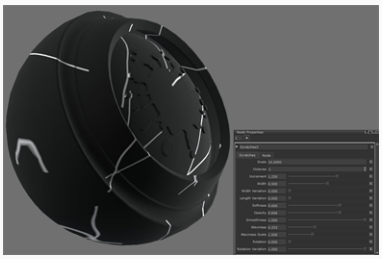
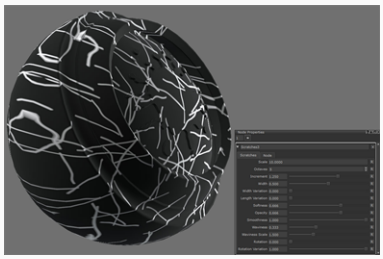
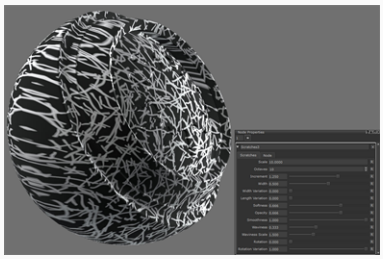
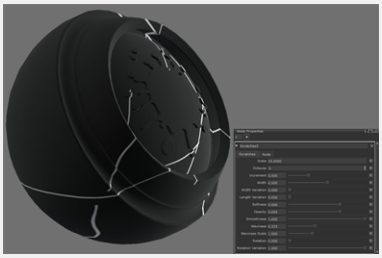
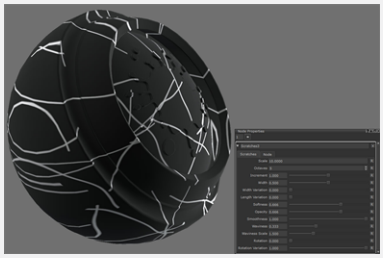
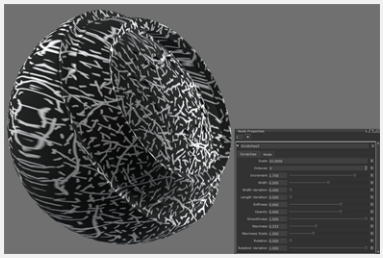
Scratches Node Inputs

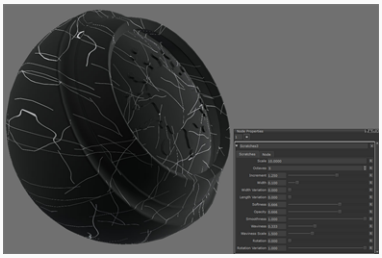
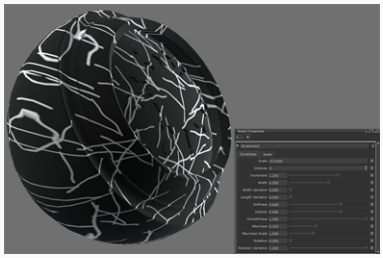
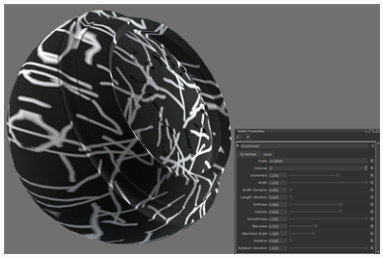
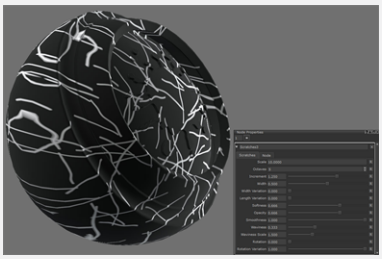
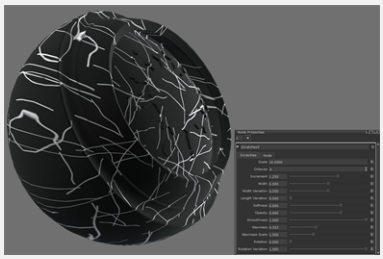
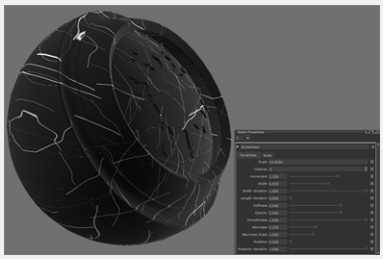
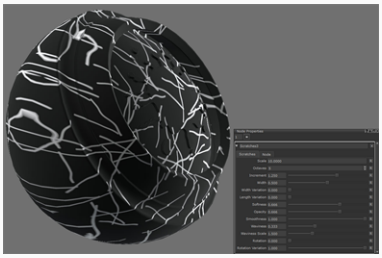
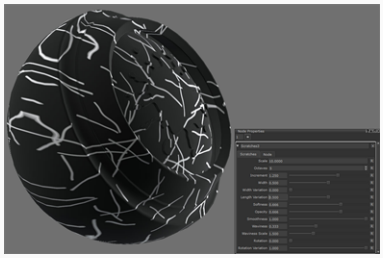
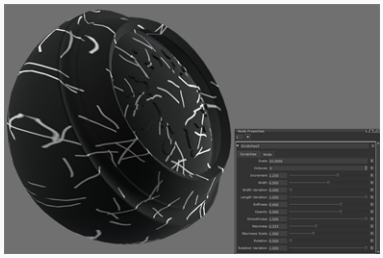
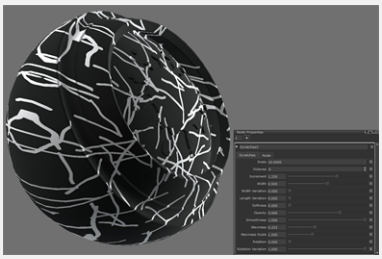
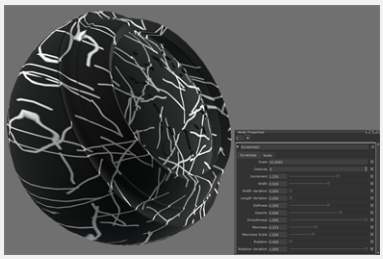
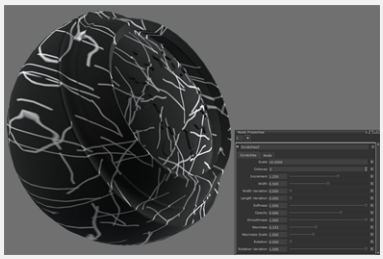
Position	The position for the scratches to be generated at. If nothing is input, this will default to world space.

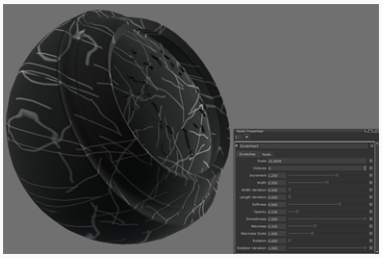
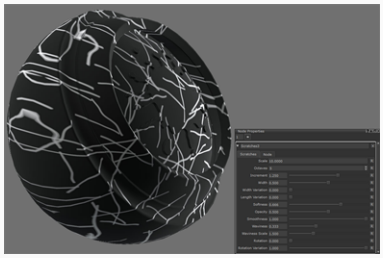
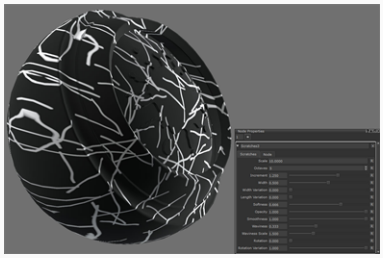
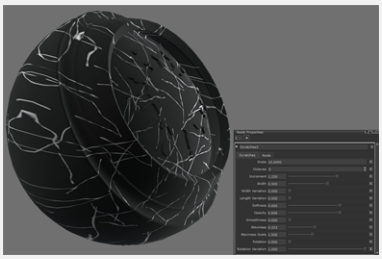
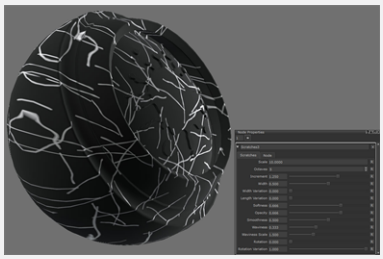
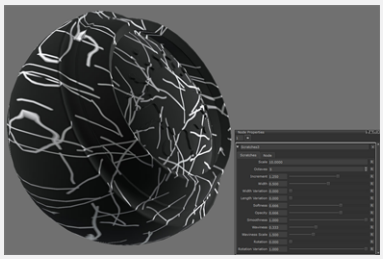
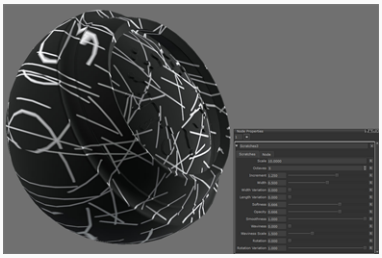
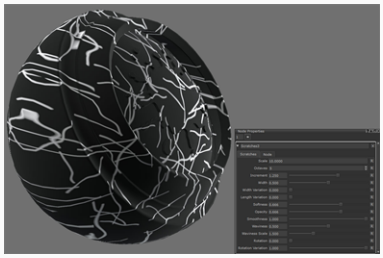
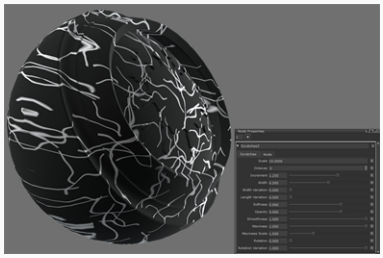
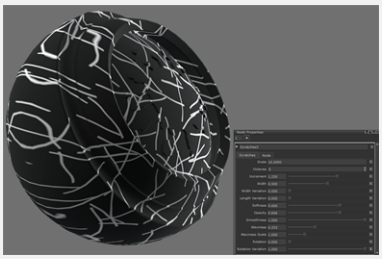
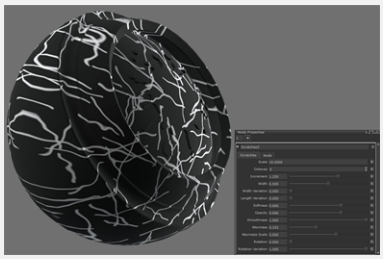
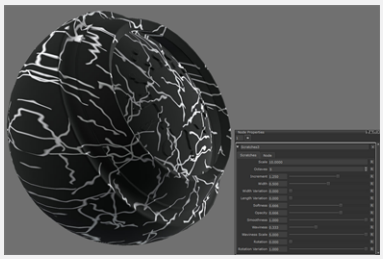
Scratches Node Outputs

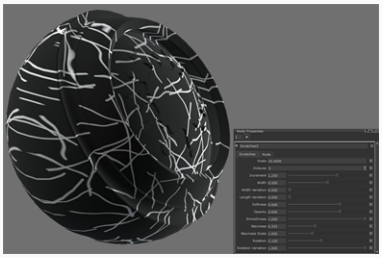
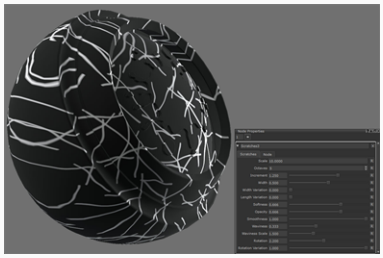
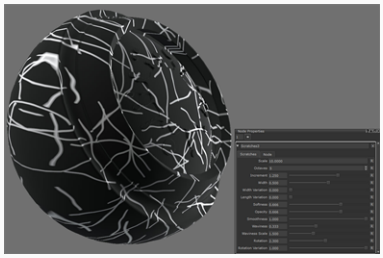
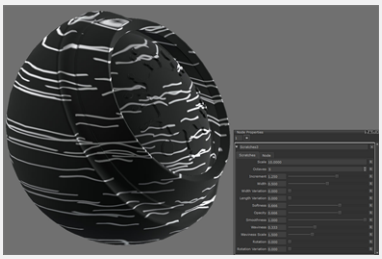
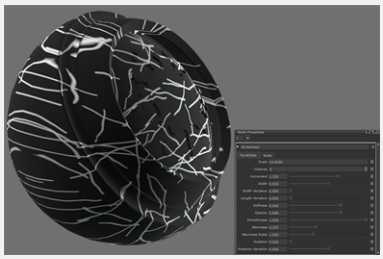
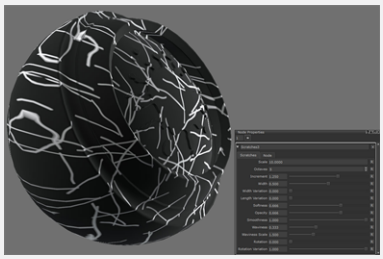
Output	Description
Output	This outputs the generated scratched pattern.

Scratches Node Properties

Scale *text field*	A lower scale means an increased density of smaller scratches, and a higher scale means fewer and larger scratches.  **Scale**: 1  **Scale**: 10  **Scale**: 50
Octaves *text field*	The scratches are applied in iterations rather than individual scratches. This value will determine the number of iterations to apply.  **Octaves**: 1  **Octaves**: 5  **Octaves**: 10
Increment *text field, slider*	The scale factor of each successive iteration.  **Increment**: 0.1  **Increment**: 1  **Increment**: 1.7
Width *text field, slider*	The width of the scratch lines.

	 **Width**: 0.1  **Width**: 0.5  **Width**: 1
Width Variation *text field, slider*	Higher values thin the scratch lines in a random fashion.  **Width Variation**: 0  **Width Variation**: 0.5  **Width Variation**: 1
Length Variation *text field, slider*	Higher values shorten the length of the scratch lines in a random fashion.  **Length Variation**: 0  **Length Variation**: 0.5  **Length Variation**: 1
Softness *text field, slider*	Controls the edge fade of the scratch lines. Lower values make the edges more crisp.  **Softness**: 0  **Softness**: 0.5  **Softness**: 1
Opacity	The opacity of the scratches.

text field, slider	 **Opacity**: 0.1  **Opacity**: 0.5  **Opacity**: 1
Smoothness *text field, slider*	Controls how smooth the edges of the scratch lines are by flattening jagged edges.  **Smoothness**: 0  **Smoothness**: 0.5  **Smoothness**: 1
Waviness *text field, slider*	Controls how jittery the path of the scratch lines appear by adding curves along the lines.  **Waviness**: 0  **Waviness**: 0.5  **Waviness**: 1
Waviness Scale *text field, slider*	Controls the scale of **Waviness**. Smaller values make the path of the scratch line jitter more frequently.  **Waviness Scale**: 1  **Waviness Scale**: 3  **Waviness Scale**: 5
Rotation	Controls the alignment of the scratch lines.

text field, slider	 **Rotation**: 2.1  **Rotation**: 2.2  **Rotation**: 2.3
Rotation Variation *text field, slider*	Controls how much each scratch line deviates from the **Rotation** value. Lower values bring all of the scratches into closer alignment.  **Rotation Variation**: 0  **Rotation Variation**: 0.5  **Rotation Variation**: 1

Squiggle Node

Access: **Nodes** > **Procedural** > **Noise** > **Squiggle**

The Squiggle node creating the bump on this matte plastic material of an old fashioned camera asset.

The Squigglenode is a procedural noise which applies a hand drawn scribbled looking noise through the object in 3d space. Adjusting the node and combining it with others allows you to get unique looking textures such as pebbles, scales or cells.

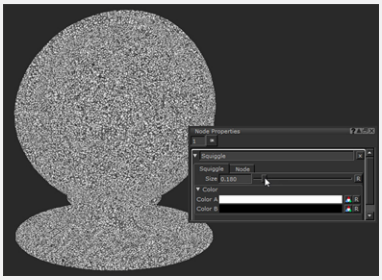
It is a useful node for merging in detail to textures or masks. Since it uses 3D space by default, when an object is updated, even its UVs, the noise still applies in a similar manner causing no repainting to be needed.

Squiggle Node Inputs

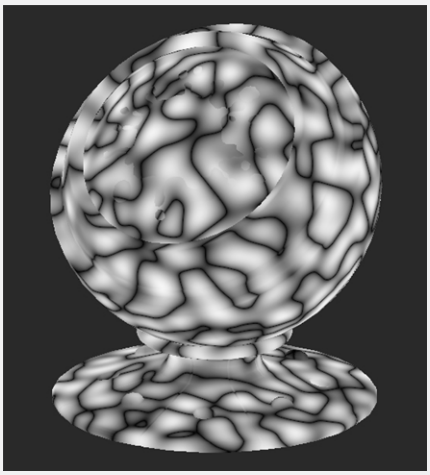
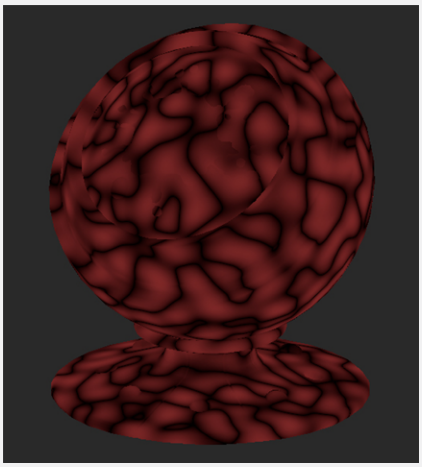
Position: The node works in a 3D space, but by adding a node to the **Position** input you can manipulate how the Squiggle noise is applied. Other nodes could be the UV node to use a UV space rather than 3D space, or the Position node.

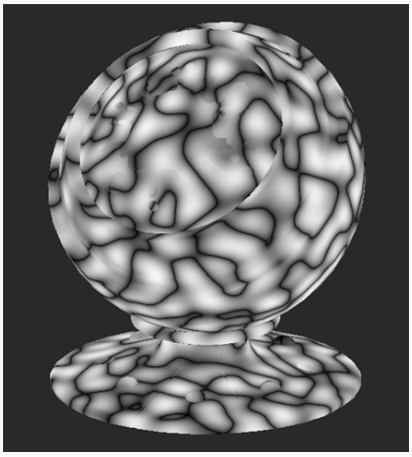
Squiggle Node Properties

Size *text field, slider*	Changes the pattern frequency of the noise.

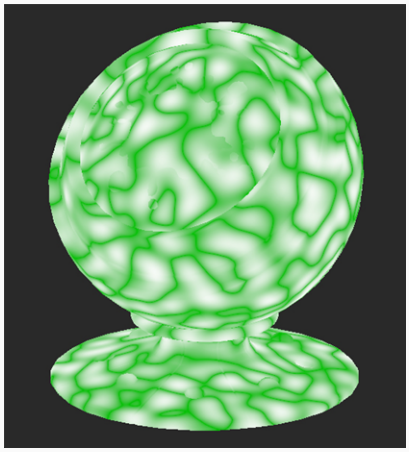
	 **Size** value set to 0.1. **Size** value set to 0.4. 0.5 is the default.

Color

Color A *swatch*	Changes the first color of the noise.  Default: **Color A** set to white.  **Color A** set to red.
Color B *swatch*	Changes the color on top of **Color A**.



Default: **Color B** set to black.



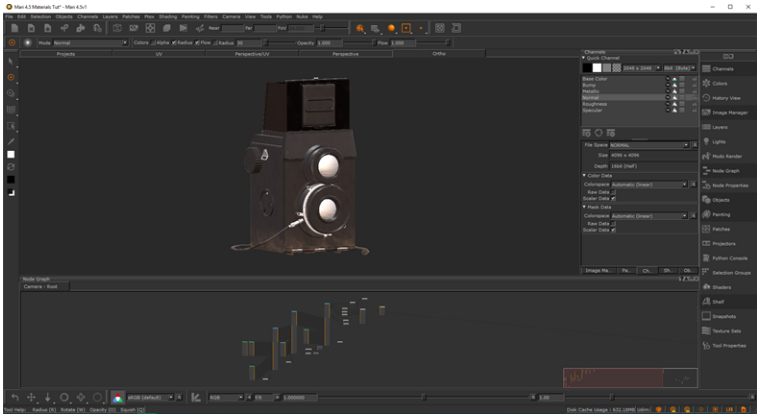
Color B set to green.

Squiggle Node Workflow Example

Creating a Bump with the Procedural Noise Squiggle Node

The procedural noises included in Mari each have a distinct look so it's good to take a proper look at your reference and work out how to recreate it. In this example we will take a look at using a procedural in a fairly advanced scene.

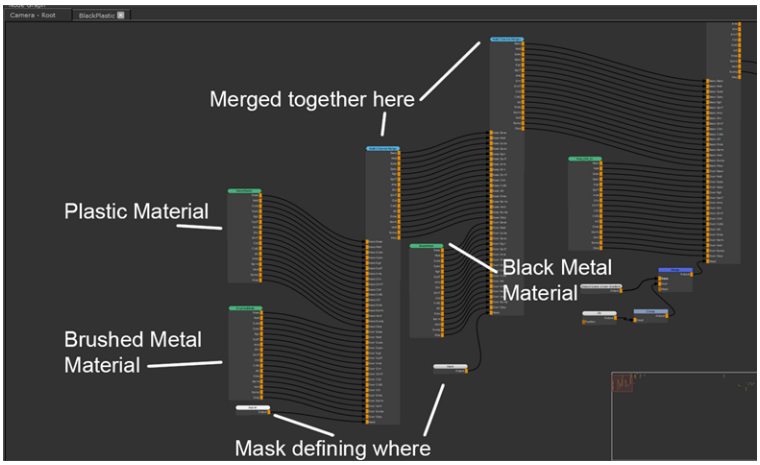
This near completed asset has been textured using the materials system. If you're unfamiliar with them then you can go to the Material Node documentation or watch this quick Foundry Youtube Video series.



An asset textured with materials.

This camera asset has a matte plastic Material node but the bump map from it isn't creating a bump that matches the reference. The material has a smooth finish and our reference has a distinct looking bump pattern.

While it may look overwhelming when zoomed out, it's a very simple project. The project is just a connection of Material nodes merged together using the Multi-Channel Merge node, utilizing masks that define where different materials should go.



Here is a breakdown of the nodes used for this texturing so far.

Material setups can look overwhelming when starting out but they are no different than normal texture methods.

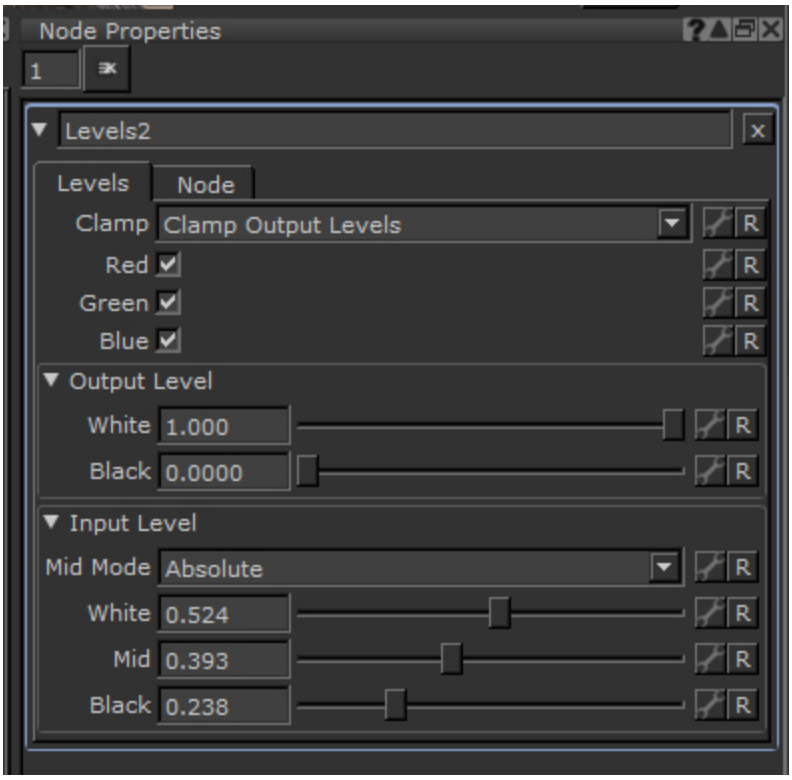
The Squiggle node is a great node to make a bump map that looks more like the real life pattern quickly. For assets that aren't going to be seen in detail on-screen, a procedural noise is a quick way to achieve your desired look without creating your own tiled texture which takes a lot longer and can be more time consuming to iterate and update.

Since it's a Material node it has everything set up inside so we need to jump into the node by pressing **Ctrl** + double-clicking on the Material node. This opens up the material tab in the Node Graph for us to enter.

Inside this Material node is a Perlin node for the bump, but in this instance a Squiggle node will be better. First, placing down a Squiggle node and plugging it in to the bump gives a good base. Lowering the **Size** of the Squiggle node also helps to better match the scale of the object.

Inside the Material node a Squiggle node has been hooked up to the **Bump** output.

The Squiggle has a range of values giving smooth rounded peaks on the bump. The reference has flattened peaks instead of rounded ones, so using a Levels node helps to add contrast and raise the values closer to white. We can then take advantage of the **Clamp** feature of the Levels node to flatten off the top and help better match the reference.



The settings used on the Levels nodes. The **Clamp** property has been used to flatten the top of the bump peaks.

The Squiggle node has quickly helped make the bump for this material, something that wouldn't have been as quick to achieve with another noise or method. The final material looks closer to our reference now than it did just using a Perlin node. Each procedural noise has their own benefits. The black metal material on the top of the camera has also had a bump added in the same way, but using the Sphere node.

The final result.

The beauty of procedurals is how versatile they are. By using other nodes in conjunction with the Squiggle node you can quickly get a variety of effects.

Tiled Node

Access: **Nodes** > **Procedural** > **Pattern** > **Tiled Node**

The Tiled node allows you to place down an image from your **Image Manger** and tile it across your UVs, repeating as many times as you define in the **Node Properties**.This can be used for many different tasks, from setting up masks, creating procedural texturing node trees, and breaking up plain looking textures with extra detail.

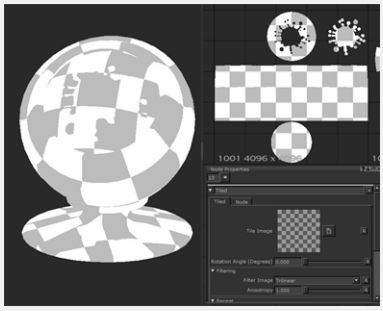
Tip: While the node doesn't require a texture that tiles, it is best to use one to avoid seams at the edges.

Tiled Node Inputs

UV - By default the Tiled node uses the UVs of the object.

Tip: However you can, for example, plug in a UV node with a Levels node to manipulate the way the Tiled node is placed across the surface of your object. You can also plug in other nodes, like noises, to get more experimental results.

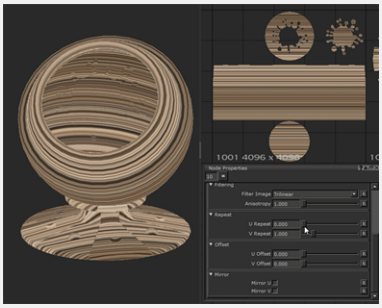
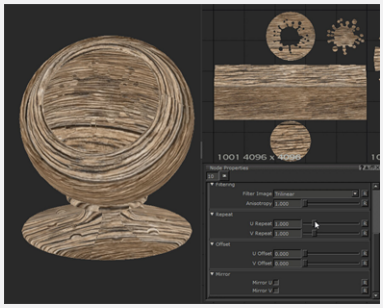
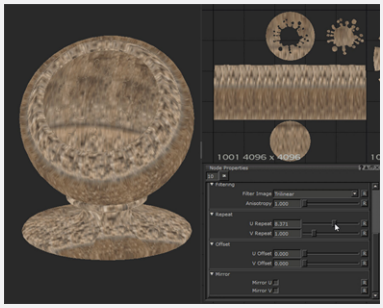
Tiled Node Properties

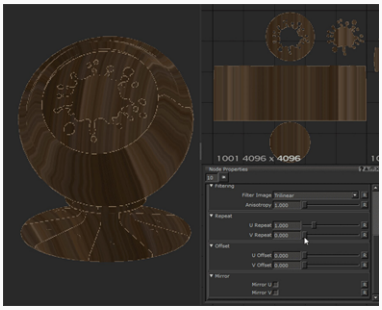
Tile Image *button*	Chooses the image to tile across your geometry's UVs. You can either use the button next to the image preview to find an image with the file browser or drag and drop an image from the **Image Manager**.  No **Tile Image** assigned. An image assigned to the **Tile Image**.
Rotation Angle (Degrees) *text field, slider*	Changes the rotation of the Tiled image input. Default is 0.

Filtering

Filter Image *dropdown*	Chooses the type of texture filtering to use on your image when upscaling or downscaling it. Default is Trilinear.
Anisotropy *Text field, slider*	Changes the amount of **Anisotropy** on the filtering of the image. Default is 1.

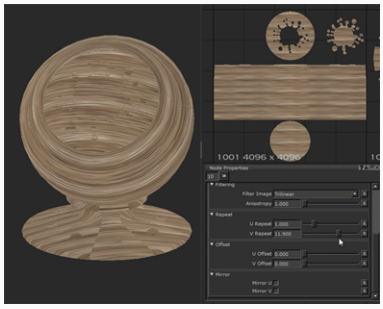
Repeat

U Repeat *text field, slider*	Changes the tiling amount horizontally along the U axis of your UVs. Increasing this value makes the image scale and repeat more.    **U Repeat** set to 0. **U Repeat** set to 1. **U Repeat** set to 8. Default is 1.
V Repeat *text field, slider*	Changes the tiling amount vertically along the V axis of your UVs. Increasing this value makes the image scale and repeat more.



V Repeat set to 0.

V Repeat set to 1.



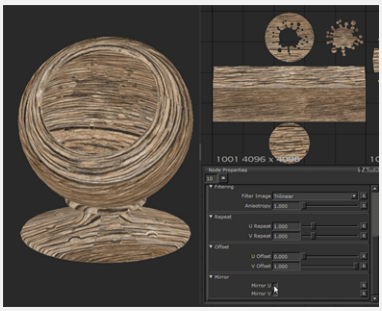
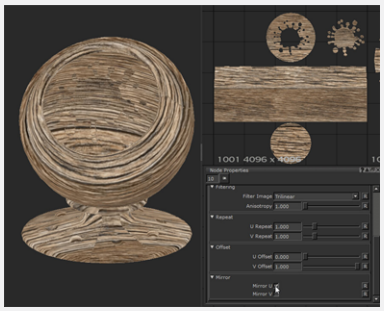
V Repeat set to 11.

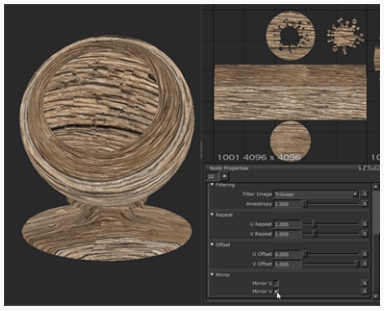
Default is 1.

Offset

U Offset *text field, slider*	Moves the tiled image horizontally along the U axis of your UVs. Default is 0.
V Offset *text field, slider*	Moves the tiled image vertically along the V axis of your UVs. Default is 0.

Mirror

Mirror U *checkbox*	Flips the image along the U axis.  **Mirror U** turned off.  **Mirror U** turned on.

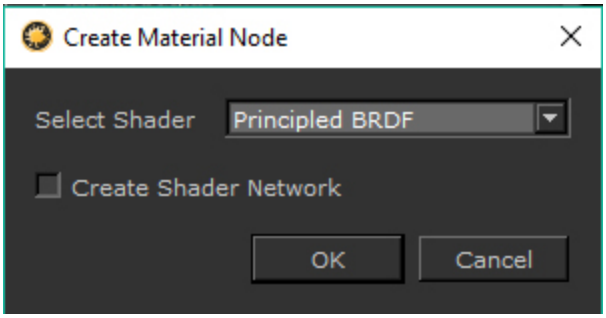
Mirror V *checkbox*	Flips the image along the V axis.  **Mirror V** turned off. **Mirror V** turned on.

Tiled Node Workflow Example

Material Creation with the Tiled Node in Mari 4.5

Material creation goes hand in hand with the Tiled node. Lets create a quick wooden material that defines every channel you need.

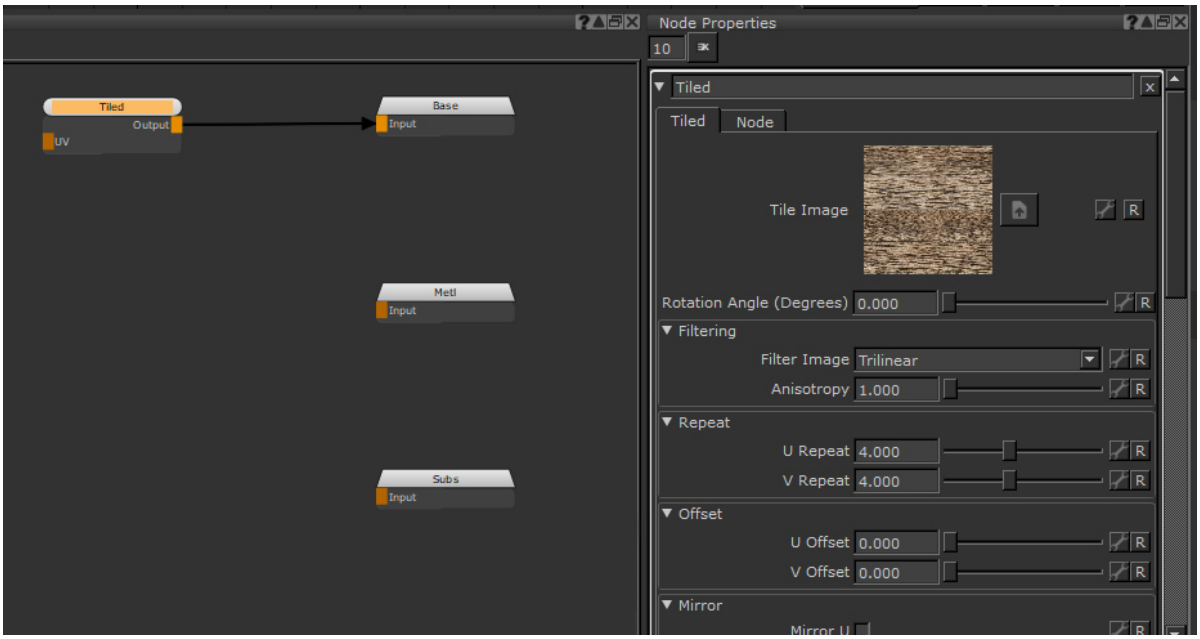
First you need to place down a Material node. The **Create Material Node** dialog asks for the shading model you would like to use. For this example **Principled BRDF** is selected but if you want to use another then the principles stay the same. If you toggle on **Create Shader Network** then a shader is also created that connects the Material node's outputs to the shader.



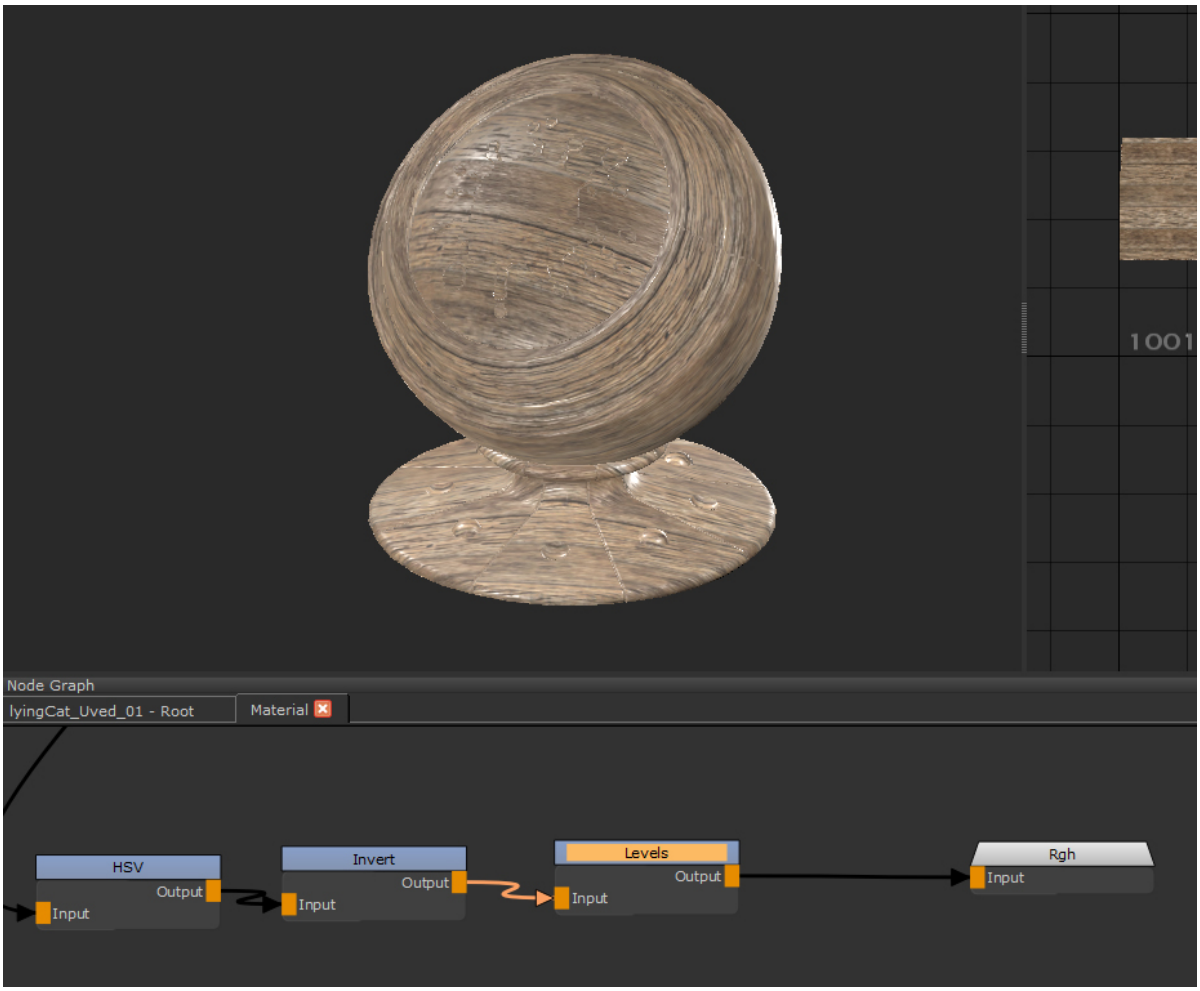
To edit it, **Ctrl** + double-click the node. This open ups a second tab in the **Node Graph**. To jump back to your main node tree just click the other tab.

Inside this new Node Graph, you can see a number of Output nodes, all named after different texture channels. When inside you can assign textures and inputs to any of the Material node's outputs.

Next we need a tiled texture. You can either make one by going through the normal node creation method or by dragging an image from the **Image Manager** into the Node Graph. This creates a Tiled node for you. From there you can change your scale so it looks right for you and make any other edits to the Tiled node until you get something you are happy with. Then it's time to plug that into the **Base Color** output.

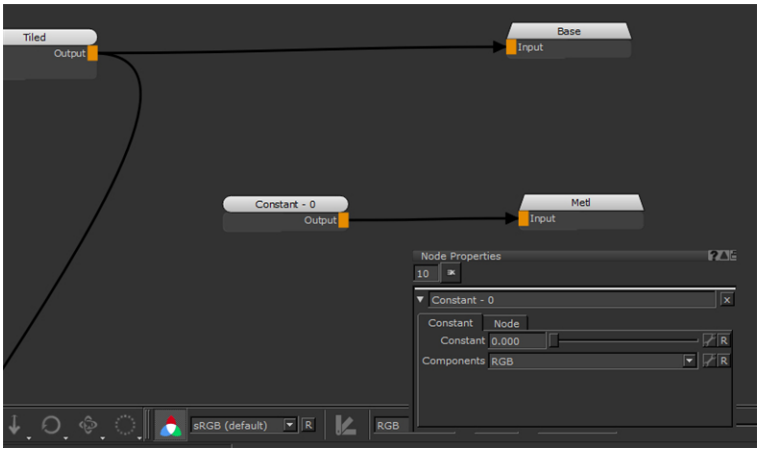


Using an HSV node and Invert node to desaturate and flip that same Tiled node we have a rough map that we can use for the specular roughness. You can use a Levels node to adjust the values until you have something that looks right. The easiest way to tell is by going back to your Node Graph and hooking the material up to a shader and viewing that so you can Lookdev values on the fly.

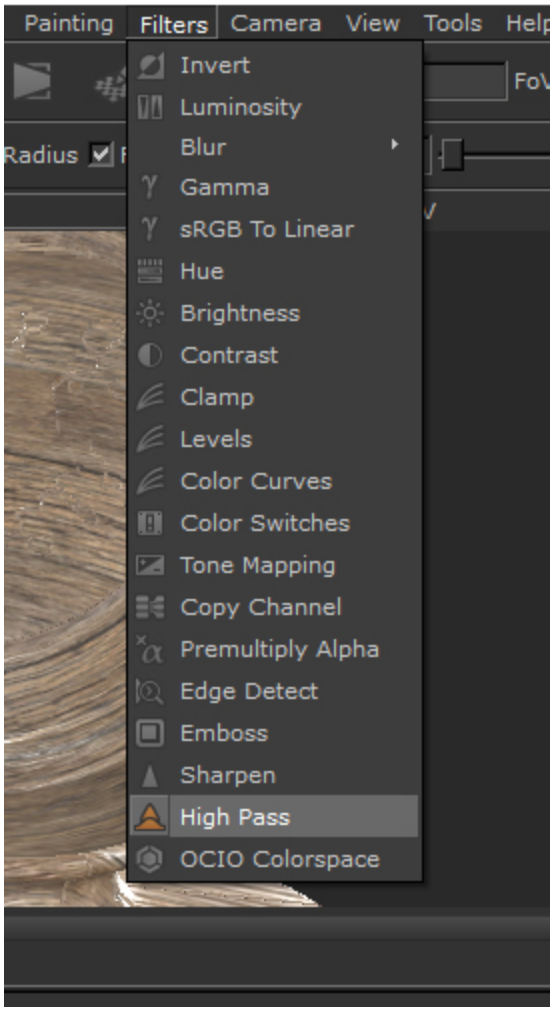


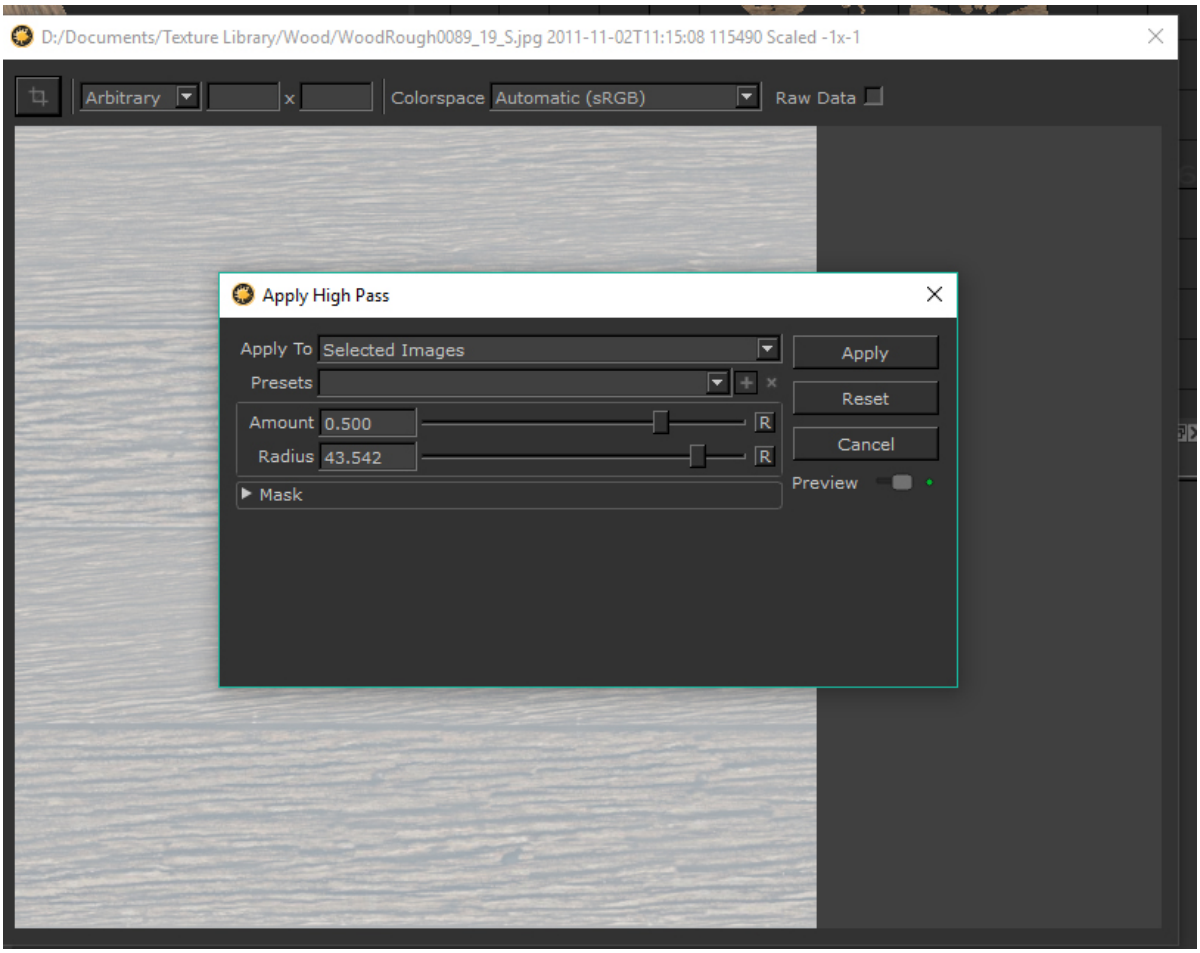
The Base Color and Rgh viewed through a shader.

Adding a Constant node set to **0** for the **RGB** component can be used for the metallic output of the Material, as wood is not metal.



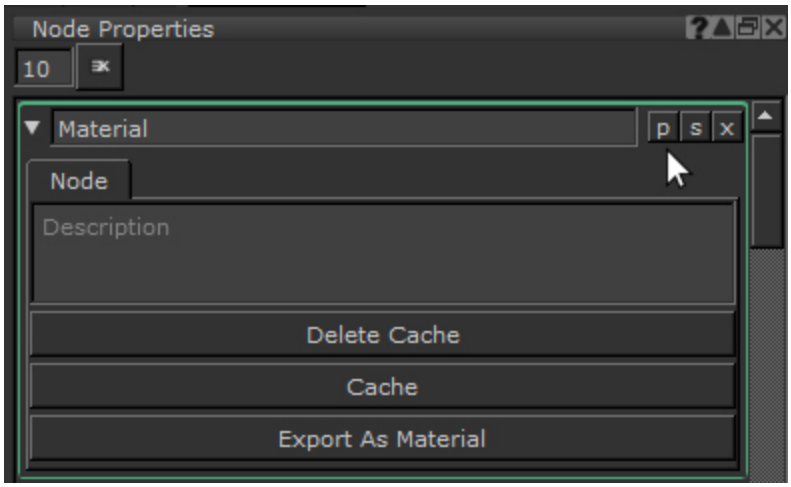
Finally to make a bump map you can use a **High Pass** filter on the wood texture in your **Image Manager** to get a quick height map from it. Make sure the **Amount** stays at 0.5 to get an image with mid-gray as the mid point like a bump map needs.



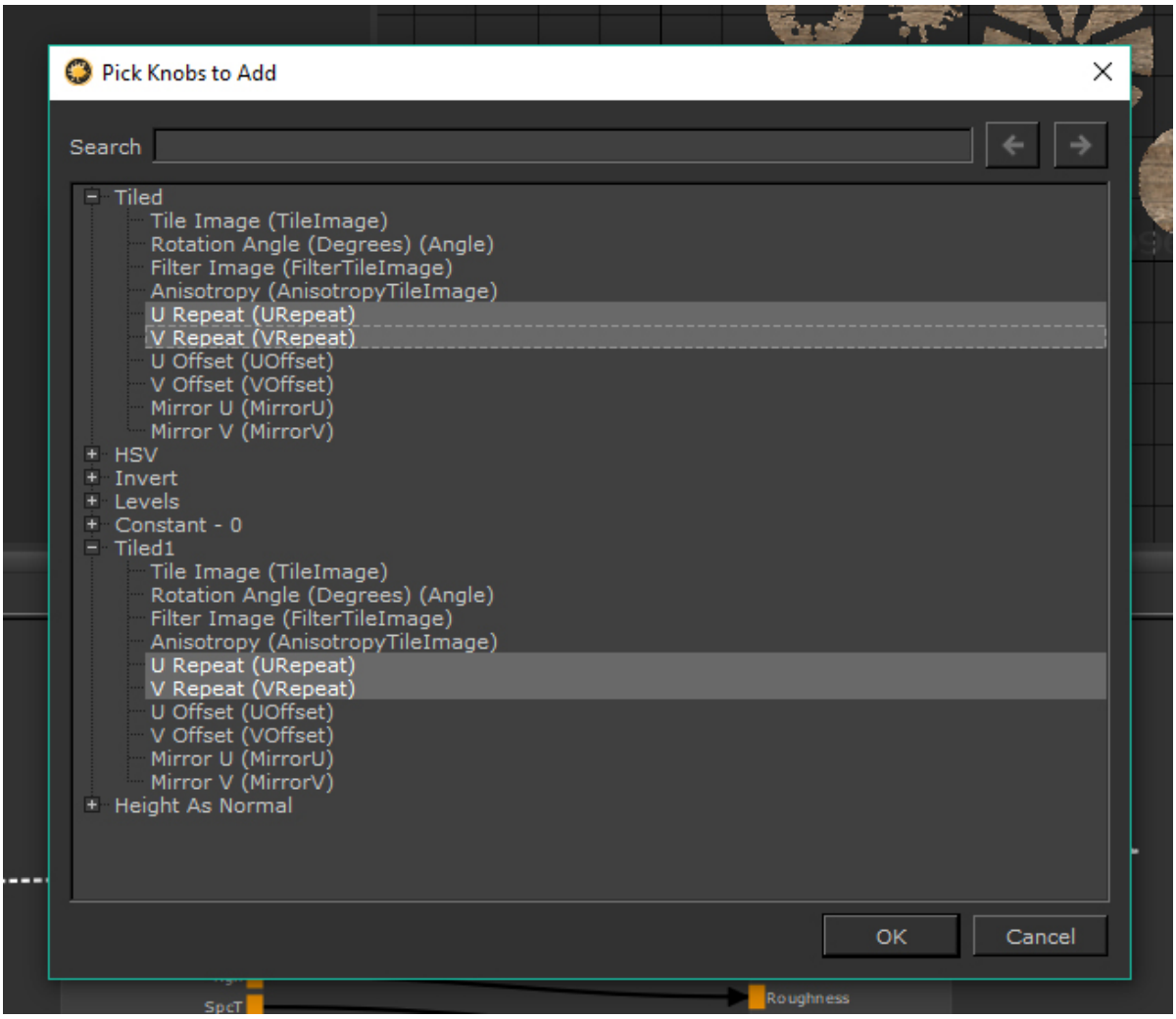


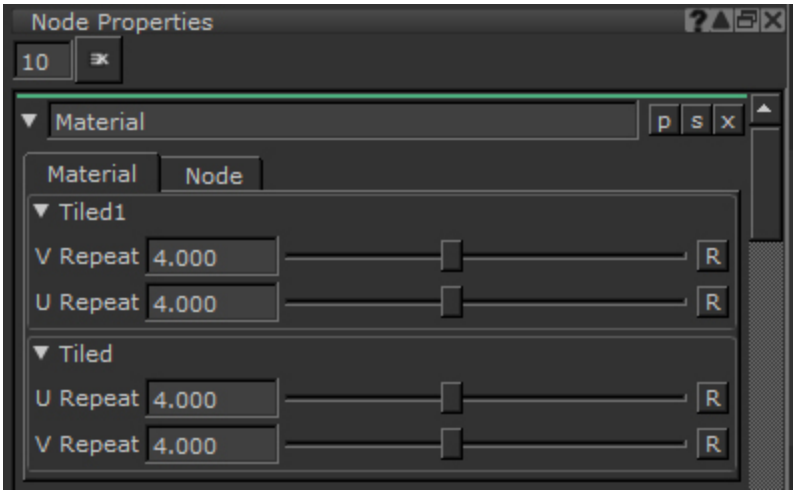
You can then use that with another Tiled node that plugs into the **Bump** output of the Material. Copying and pasting the first Tiled node is a good idea so the **Repeat** tiling amount lines up with the **Base Color**. Here's the final result, a material you can export and reuse in other scenes, quickly built up with a single tiled texture with the Tiled node.

Finally lets promote the Tiled node properties so they can be edited on the fly without having to jump in and out of the node every time. If you jump back to your Node Graph and click the **p** button on the **Node Properties** you open up a new **Group Node Knobs** dialog.

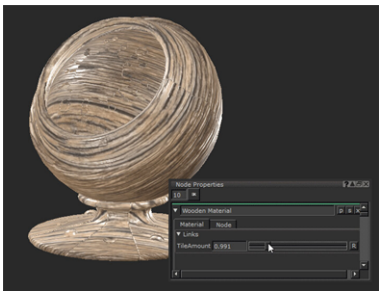


From here you need to choose which properties to promote. Click the **Pick** option and open up the two Tiled nodes with the plus icon next to their names. From there select the **U Repeat** and **V Repeat** of both Tiled nodes in the **Pick Knobs to Add** dialog and click **OK**. They are promoted to the Material node. Now they can be edited on the fly without going in and out of the Material node.

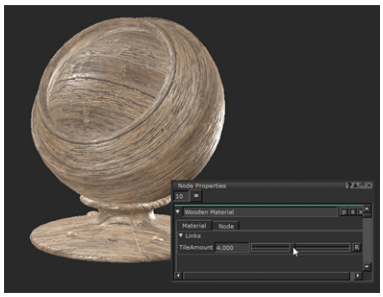




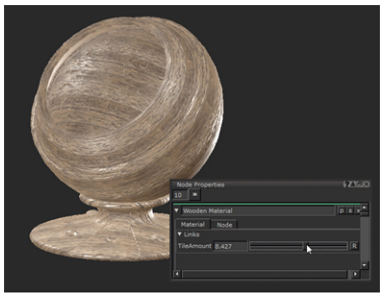
If you select those promoted properties back in the **Group Node Knobs** dialog and click the **Link** button you can merge them all into a single slider of your choosing. Make it something that is obvious, like **TileAmount**, and you will now see that new linked attribute in the **Node Properties**.



TileAmount set to 0.9.



TileAmount set to 4.



TileAmount set to 8.

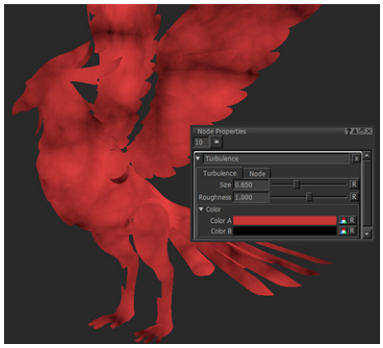
That is a way to quickly build up some of the output channels of a Material node from a single tiled texture. You can use this as a base and tweak things or add additional details with other procedural nodes like noises to add detail and complexity.

Turbulence Node

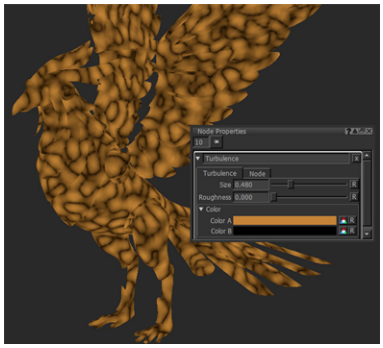
Access: **Nodes** > **Procedural** > **Fractal** > **Turbulence**

The Turbulence node is a type of noise node that generates random patterns to simulate stormy cloud-like textures.

Turbulence node's default settings



*Turbulence node's **Size** increased, **Color A** red*



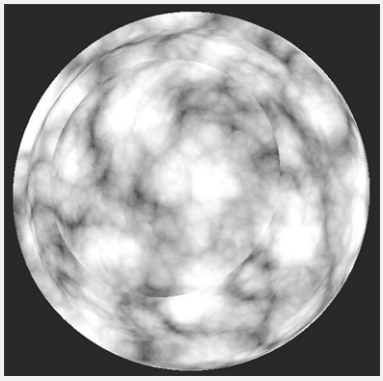
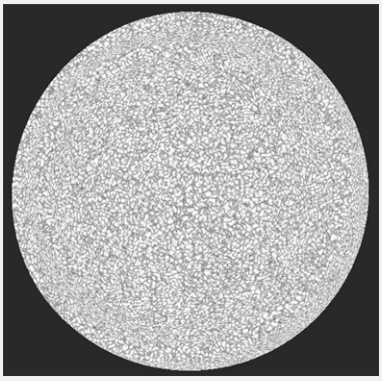
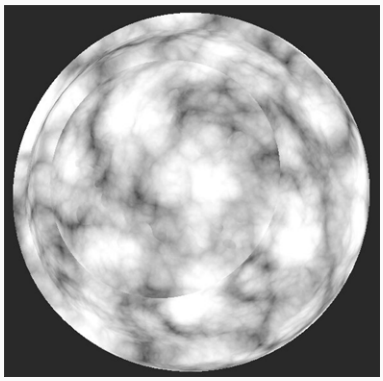
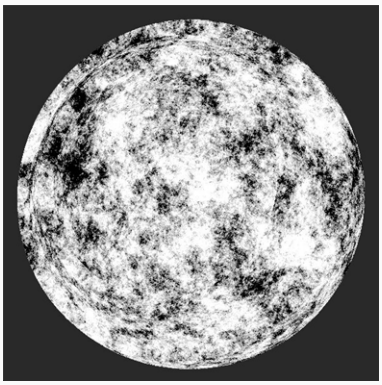
*Turbulence node's **Size** decreased, **Roughness** decreased, **Color A** yellow*

The Turbulence node relies on 2D or 3D position information (data) on the object. Its **Position** input can be connected to any position nodes and you can adjust the position in various ways.

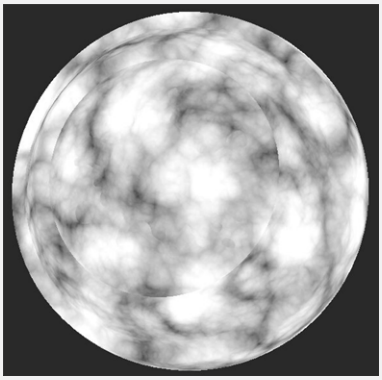
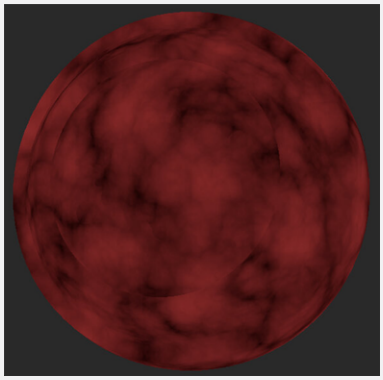
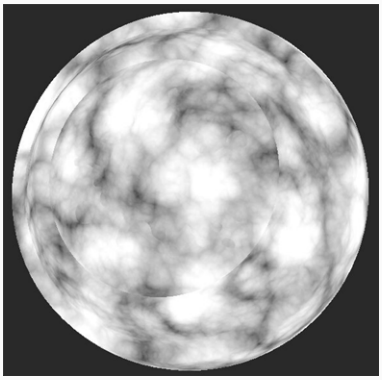
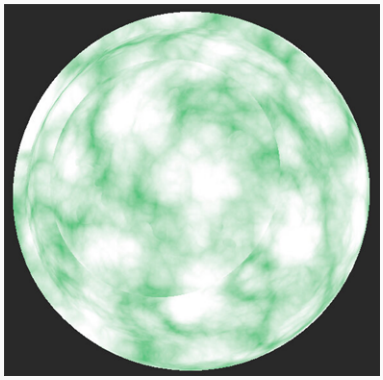
Turbulence Node Inputs

Position - Any position data can be connected to the **Position** input, such as the UV and Position nodes. You can use any type of node to modify the mapping of the pattern. For example, you can use the Scale or Vector node in conjunction with the Position or UV node to stretch or offset the position (data) from the Position or UV node.

Turbulence zNode Properties

Size *floating point control*	Changes the size of the noise.  Default: 0.500  0.190
Roughness *floating point control*	Changes the roughness appearance of the noise, from very smooth at low values to very rough.  Default: 0.500  0.300

Color

Color A *swatch*	Changes the background color.  Default: white  Red
Color B swatch	Changes the color around the stormy cloud-like pattern.  Default: black  Green

Turbulence: Node Graph Workflow Examples

Example - Obtaining a Blended Texture Using a Turbulence Node

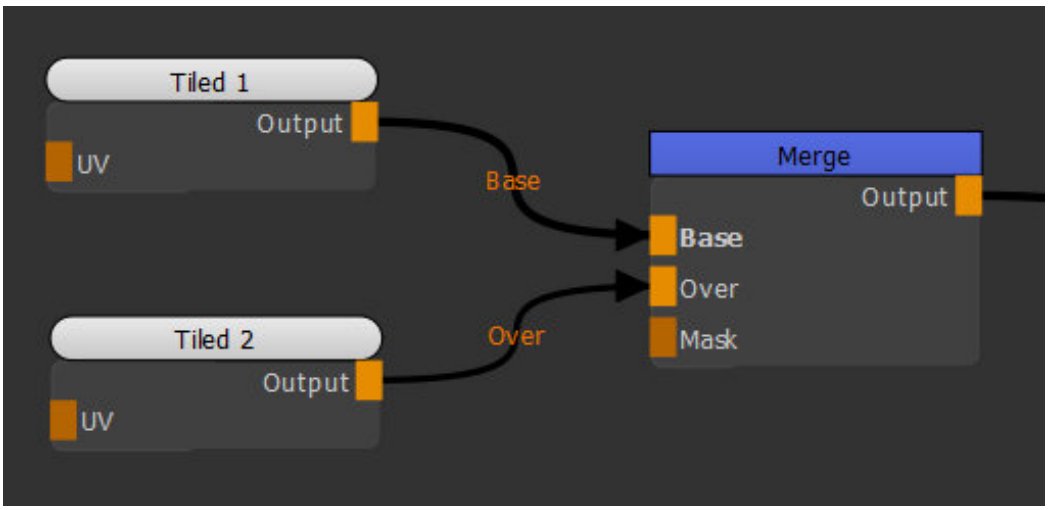
In this example, let's use the Turbulence node as a mask on top of two different textures (Tiled 1 and Tiled 2) to obtain a nice blended texture.

These are the visual outputs of Tiled 1 and Tiled 2.

Texture 1 is hidden under Texture 2.

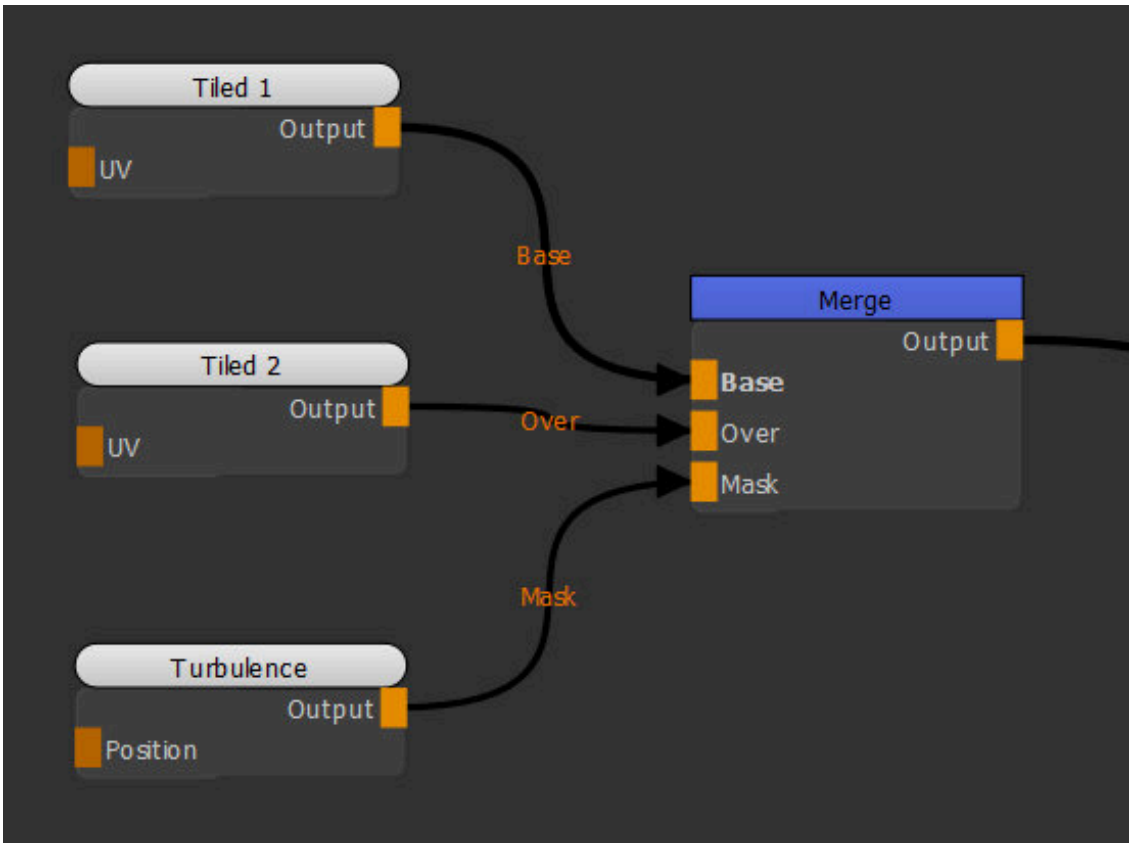
Texture 2 is visible.

1. Create two Tiled nodes and connect them to the **Base** and **Over** input of the Merge node.



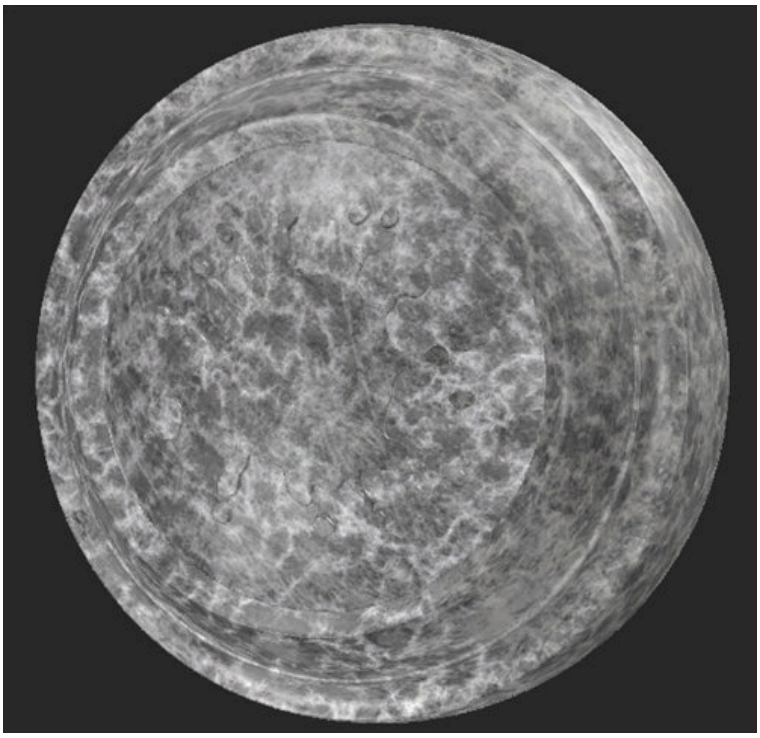
Adding two Tiled nodes to the Merge node to achieve a layering operation.

2. Connect a Turbulencenode to the **Mask** input of the Merge node.



Adding a Turbulence node the scene as a mask to blend Tiled 1 and Tiled 2 together.

Notice how Texture 1 and Texture 2 are both made visible due to the Turbulence node applied as a mask.

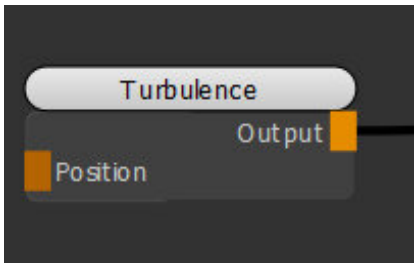


Texture 1 and Texture 2 are both made visible due to the Turbulence node applied as a mask.

Example - Switching Mapping from 3D space to UV space

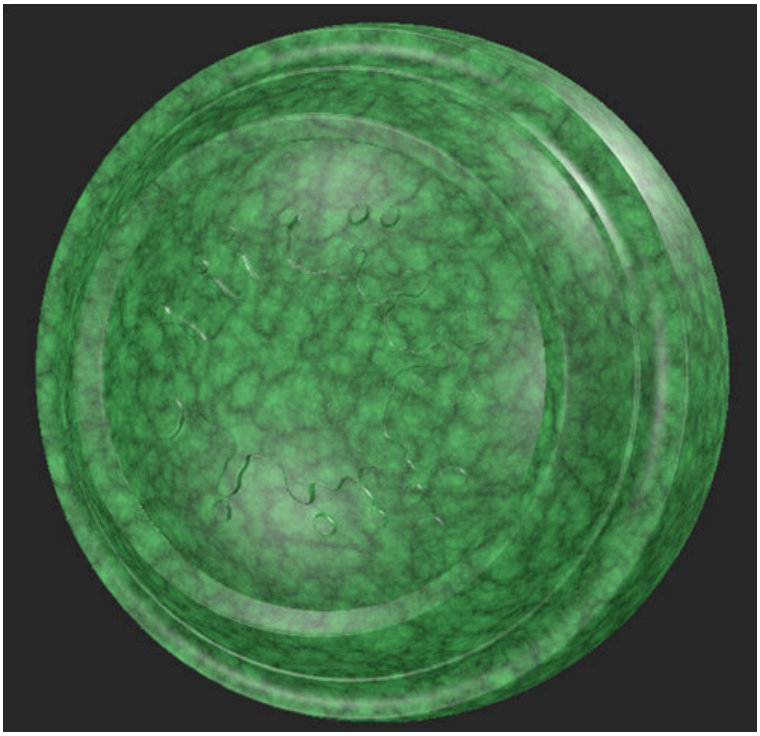
In this example, let's use a UV node to adjust the Turbulence noise from 3D space to UV space.

1. Create a Turbulence node.



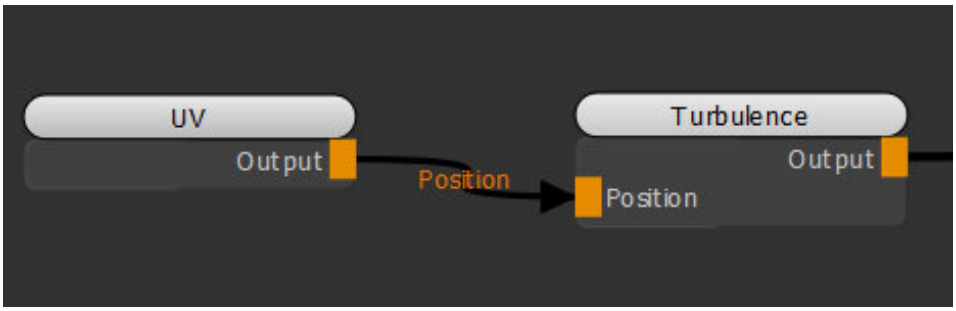
Adding a Turbulence node to the scene.

The Turbulence node is represented in 3D space. Notice that this method doesn't contain UV seams.



Turbulence node represented in 3D space.

2. Create a UV node and connect it to the **Position** input of the Turbulence node.



Adding a UV node to the scene by connecting it to the Turbulence node.

The Turbulence noise is now represented in 2D space.
Notice the UV seam on the right-hand side of the model, which shows the border of a patch.

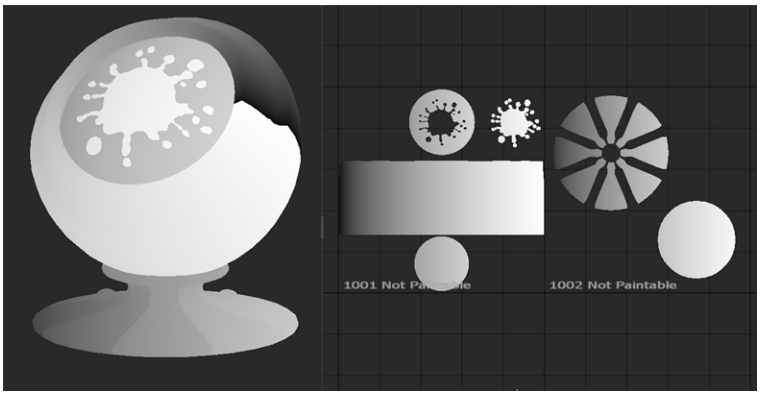
Turbulence node represented in 2D space.

The following image is the UV view of the model to give you a better idea of the noise layout.

UV view of the model.

UV Linear Gradient Node

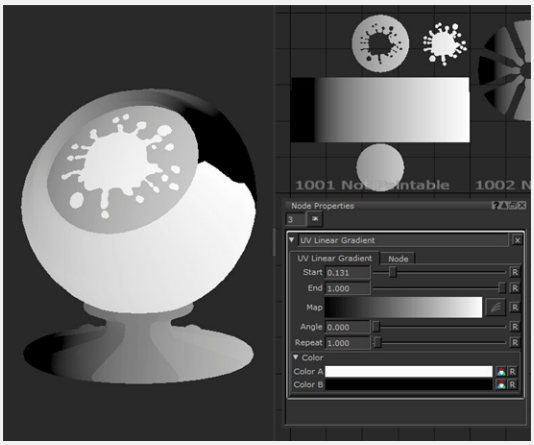
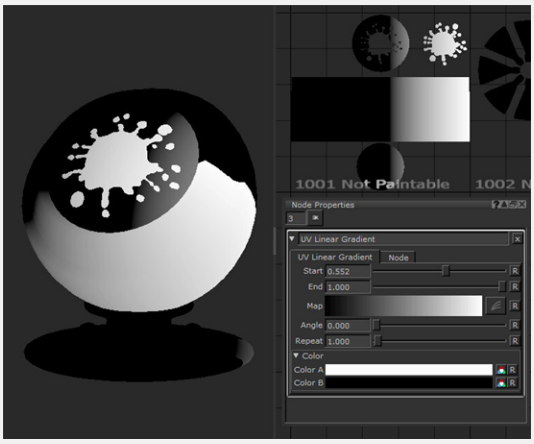
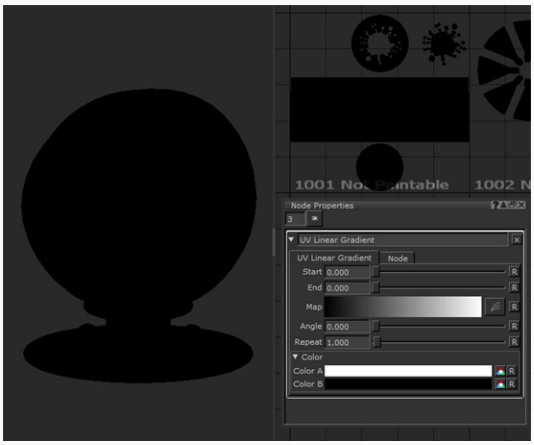
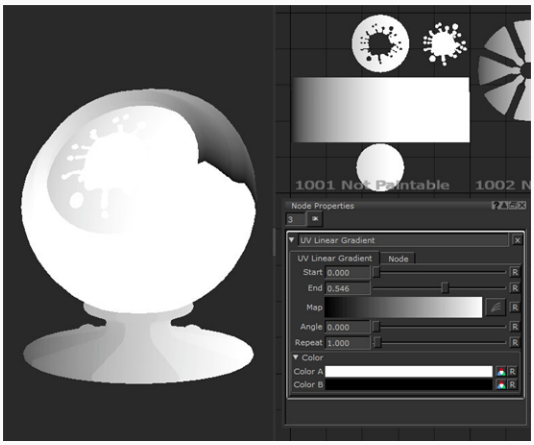
Access: **Nodes** > **Procedural** > **Pattern** > **UV Linear Gradient**

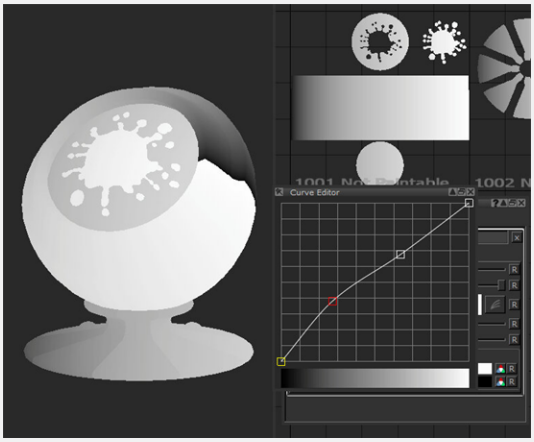
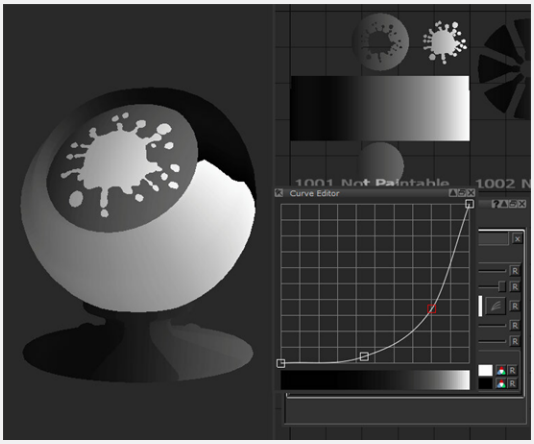


UV Linear Gradient default settings on geometry.

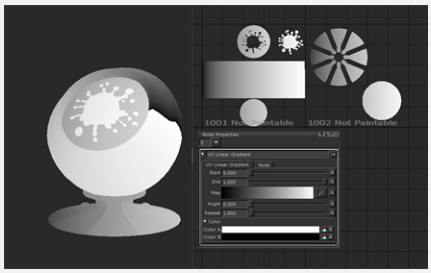
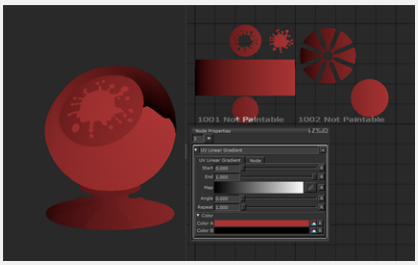
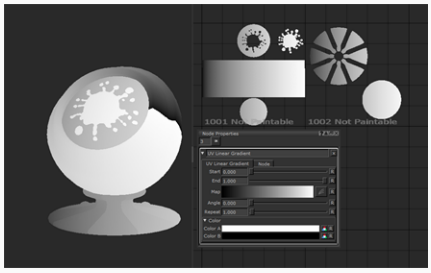
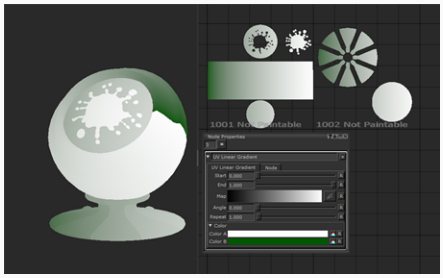
The UV Linear Gradient node is a great way to applied graduated colors across your objects. The node gives you a lot of options to change the result and it is a really handy node for creating quick masks based on the way your objects UVs have been laid out. By editing the properties you can make looser or tighter gradients and with the provided ramp options you can add extra levels of fidelity to the output.

UV Linear Gradient Node Properties

Start *text field, slider*	Changes the UV location of the start of the gradient. 0 is the edge of your mesh's dimensions. Increasing it causes the start point to move further along your UVs. The slider has a cap of 0 but you can type in minus numbers to push it further away, stretching out the gradation.  **Start** value of 0.1.  **Start** value of 0.5. Default is 0.
End *text field, slider*	Changes the UV location of the end of the gradient. 1 is the edge of your mesh's dimensions. Decreasing it causes the start point to move further along your UVs. The slider has a cap of 1 but you can type in larger numbers to push it further away, stretching out the gradation.  **End** value of 0.  **End** value of 0.5. Default is 1.

Map *curve editor*	By clicking the curve editor icon you can change the curve of the gradient instead of it being a linear 0 to 1 ramp.   Default is a black to white linear gradient.

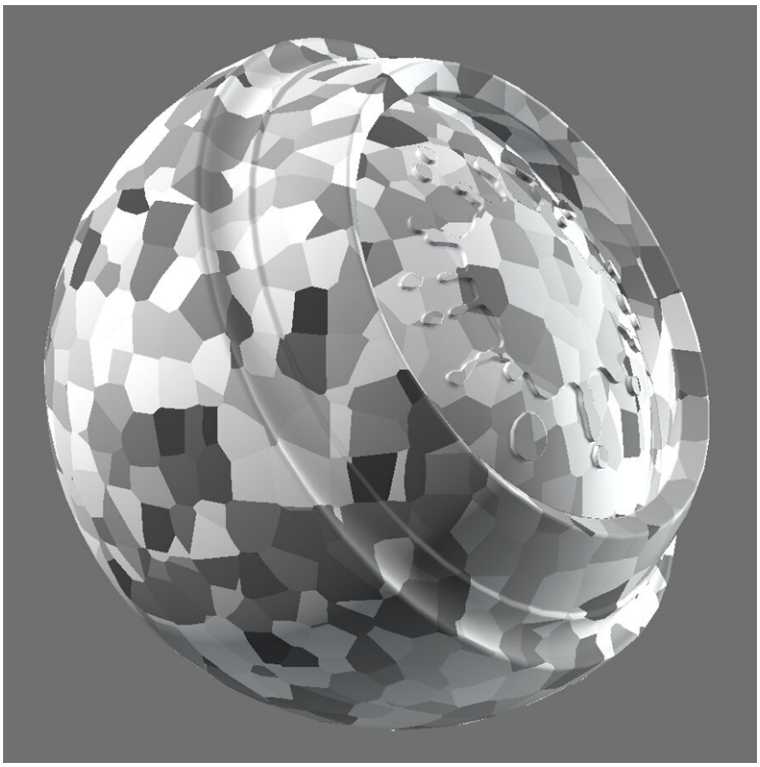
Color

Color A *swatch*	Changes the start color of the gradient.  Default: **Color A** set to white  **Color A** set to red
Color B *swatch*	Changes the end color of the gradient.  Default: **Color B** set to Black  **Color B** set to green

Voronoi Node

Access: **Nodes** > **Procedural** > **Noise** > **Voronoi**

The Voronoi node creates a type of procedural noise. It is used to generate cells based on a Voronoi algorithm. Each point on the surface is owned by the nearest vertex of a jittered grid and generates a value from that vertex.



Voronoi Noise on Geometry

Voronoi Node Inputs

Input	Description
Position	The position to use when sampling grid vertices. This defaults to surface world position when the **Space** attribute is **3D**, and surface UV when the **Space** attribute is **UV**.

Voronoi Node Outputs

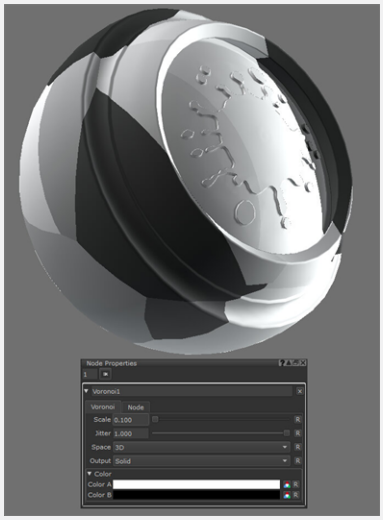
Output	Description
Output	The output specified by the **Output** attribute.
UV	XY components are UV coordinates for the cell, centered around the nearest grid vertex. W output is 1.0 if the UV coordinate is inside the 0-1 range, 0.0 if it is outside of it.

Voronoi Node Properties

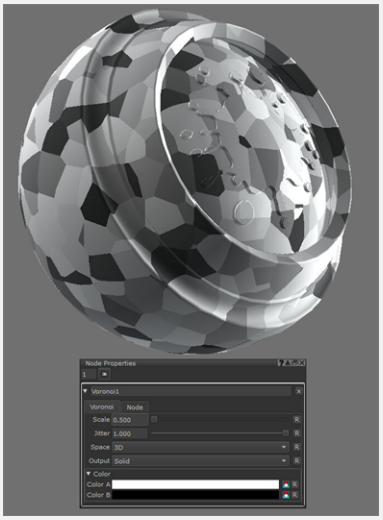
Scale

text field, slider

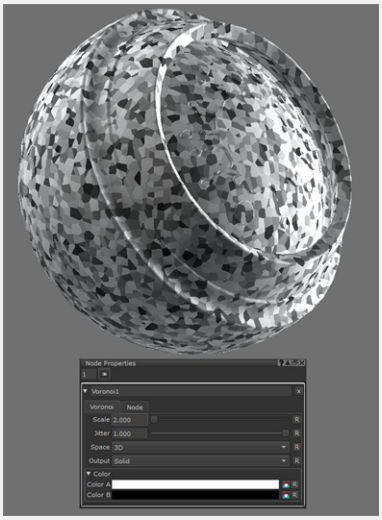
This value scales the jittered grid to sample from.



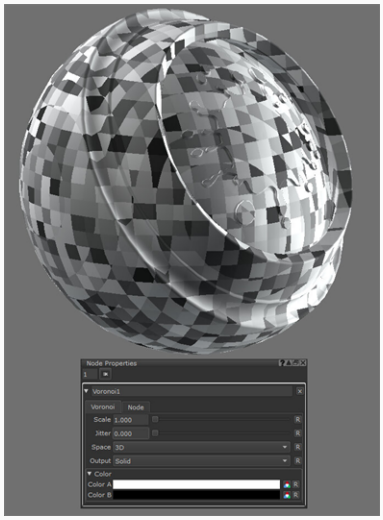
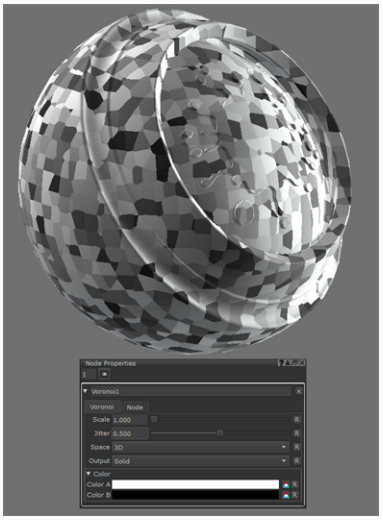
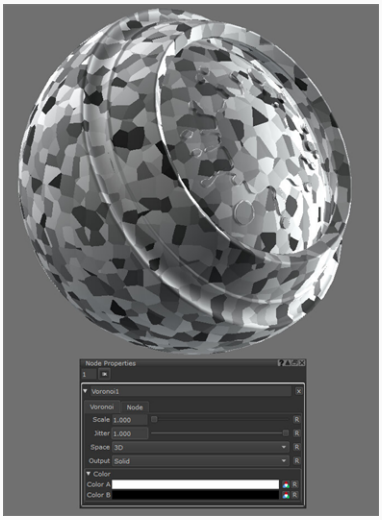
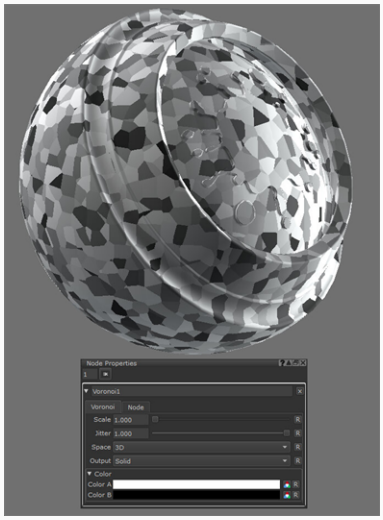
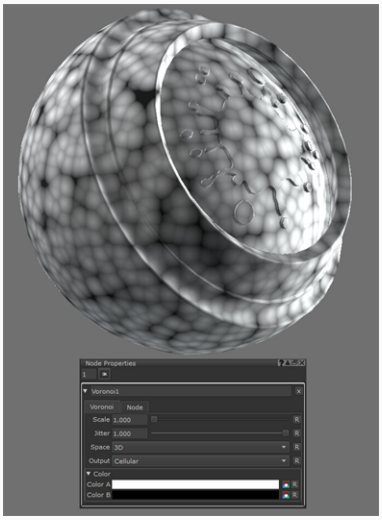
Scale: 0.1



Scale: 0.5



Scale: 2

Jitter *text field, slider*	This controls the amount the grid is jittered by. 0 being a uniform grid and 1 being fully randomized.    **Jitter**: 0 **Jitter**: 0.5 **Jitter**: 1
Space *dropdown list*	Select whether to work in **3D** or **UV** space. If **3D** is selected, the grid vertices are found using surface position. If **UV** is selected, the grid vertices are found using surface UV.
Output *dropdown list*	Select how to blend between the **Color A** and **Color B** attributes in each cell. **Solid** - Each cell is a solid color that is a random blend between **Color A** and **Color B**. **Cellular** - Each cell is a gradient from **Color A** at the center to **Color B** at the edges.   Output: **Solid** Output: **Cellular**

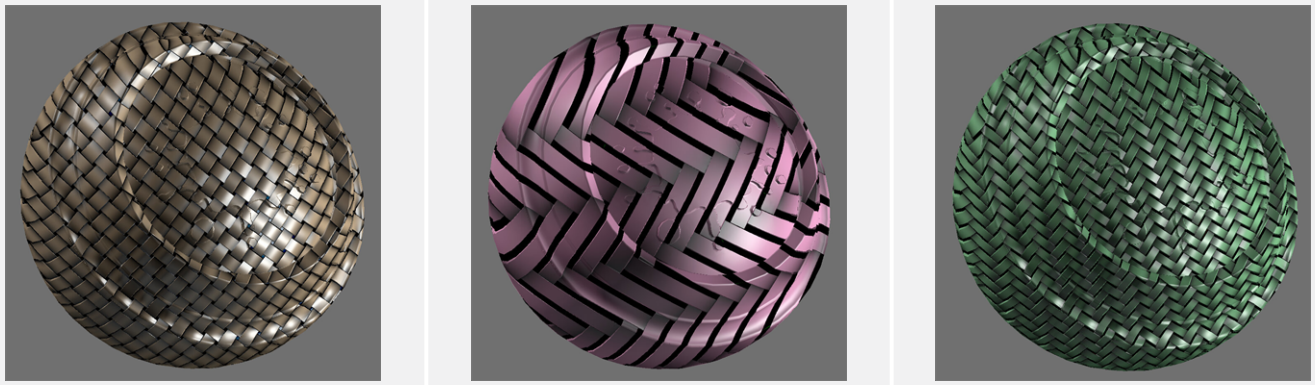
Color

Color A *color select*	The output value of a given cell is a random blend between this color and **Color B**.
Color B *color select*	The output value of a given cell is a random blend between this color and **Color A**.

Weave Node

Access: **Nodes** > **Procedural** > **Pattern** > **Weave**

This node generates a pattern of interwoven strips. This can then be used as a height map, mask or as both to texture objects with a woven material.



Weave node examples with varying parameters

Weave Node Inputs

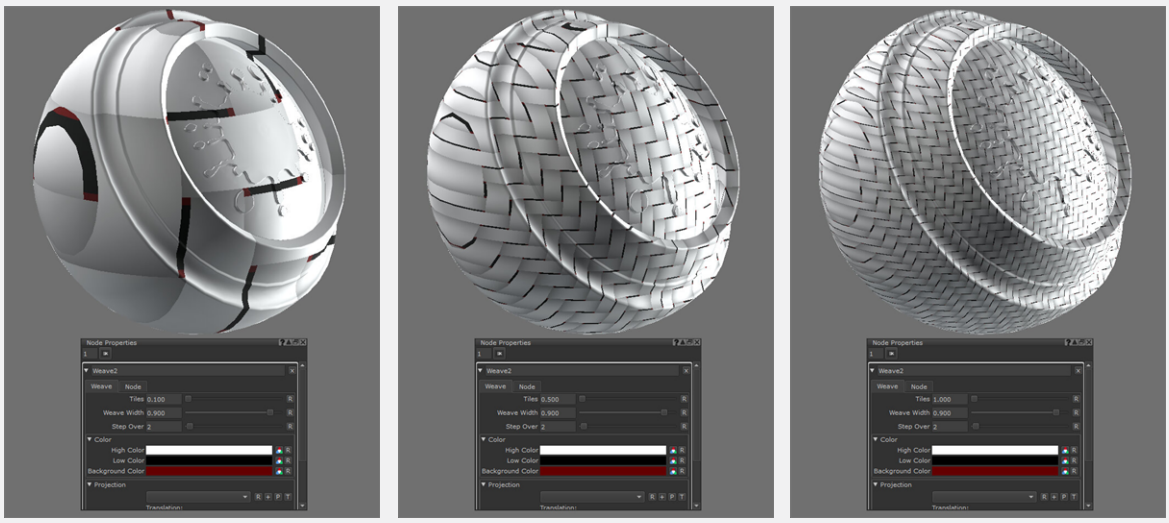
Input	Description

Position	The position at which to generate the pattern, a Position node is normally used as the input here. Defaults to surface world position.

Weave Node Outputs

Output	Description
Output	This will output the high areas, low areas and background.
Horizontal UV/Mask	R/G are UV coordinates for the horizontal weaves. B is a mask for the horizontal weaves. A is where the horizontal masks are the topmost weave. This can be used to output a mask of the horizontal strips only.
Vertical UV/Mask	R/G are UV coordinates for the vertical weaves. B is a mask for the vertical weaves. A is where the vertical masks are the topmost weave. This can be used to output a mask of the vertical strips only.

Weave Node Properties

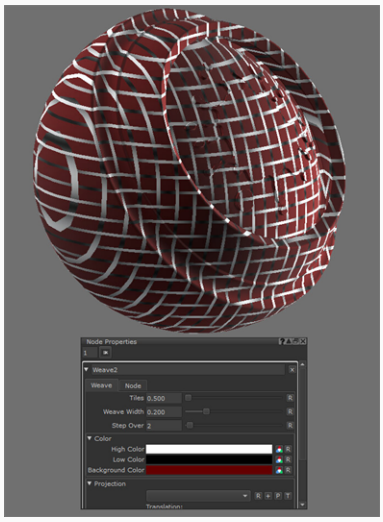
Tiles *text field, slider*	This controls the size of the weave pattern. 

	Tiles: 0.1	**Tiles**: 0.5	**Tiles**: 1

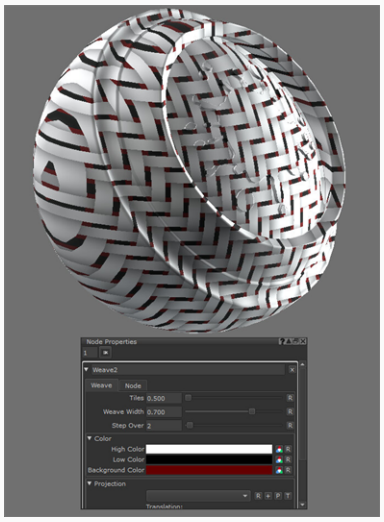
Weave Width

text field, slider

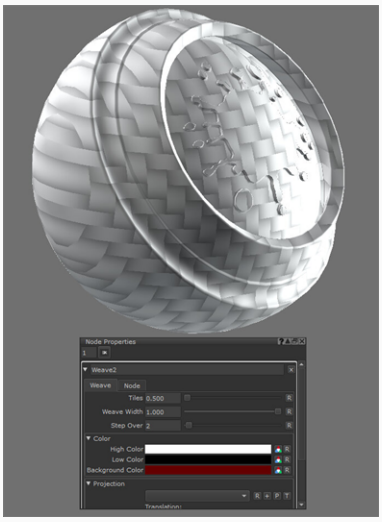
This controls the width of the strips.



Weave Width: 0.2



Weave Width: 0.7

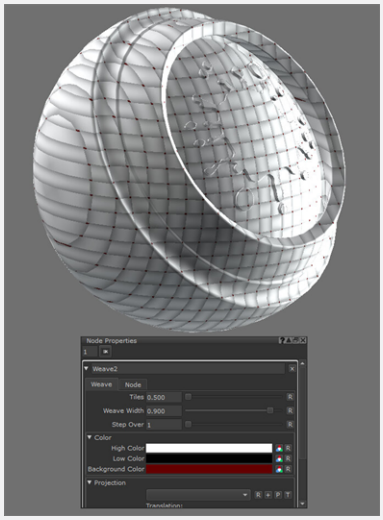


Weave Width: 1

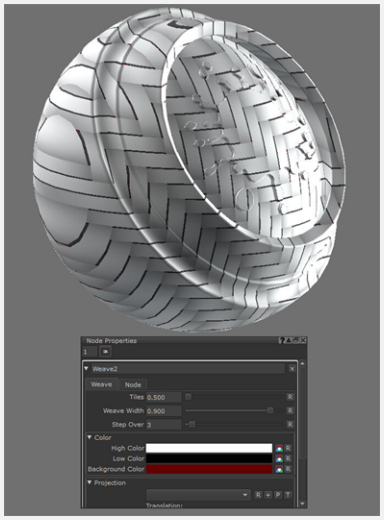
Step Over

text field, slider

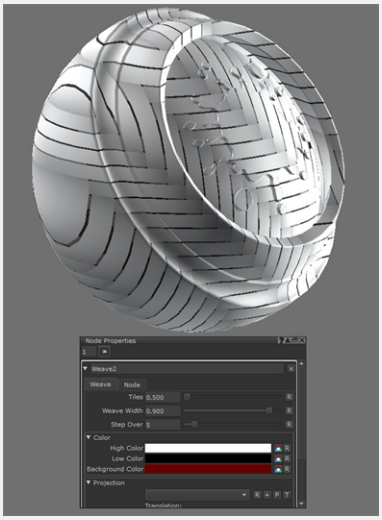
This controls how many opposing weaves to step over. For example, a value of 1 would give a standard over and under weave, a value of 2 means two weaves go under and two go over, and so on.



Step Over: 1

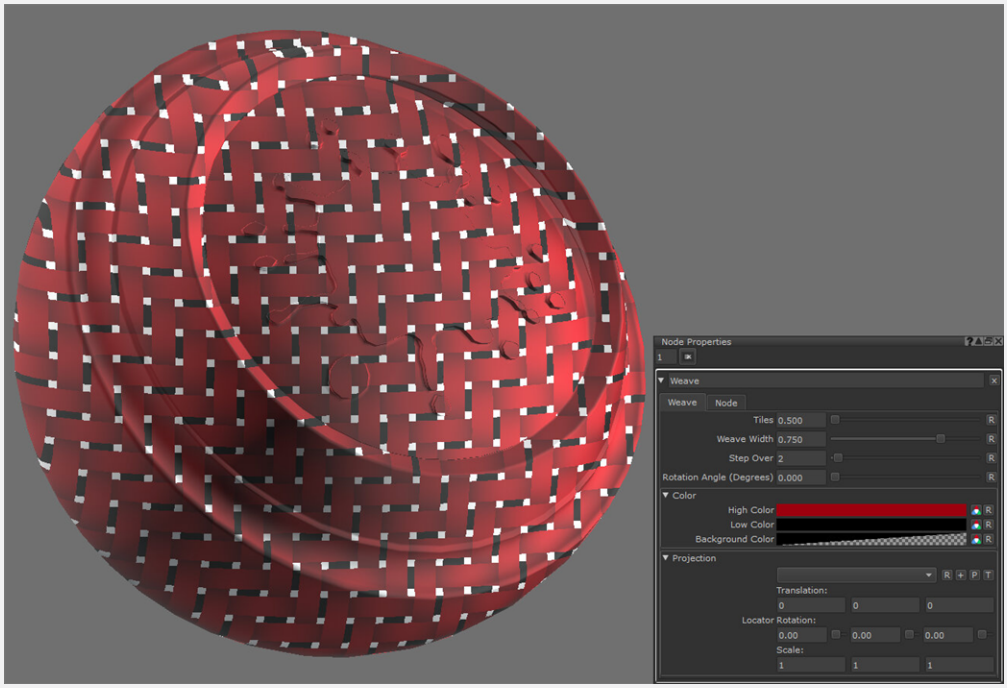


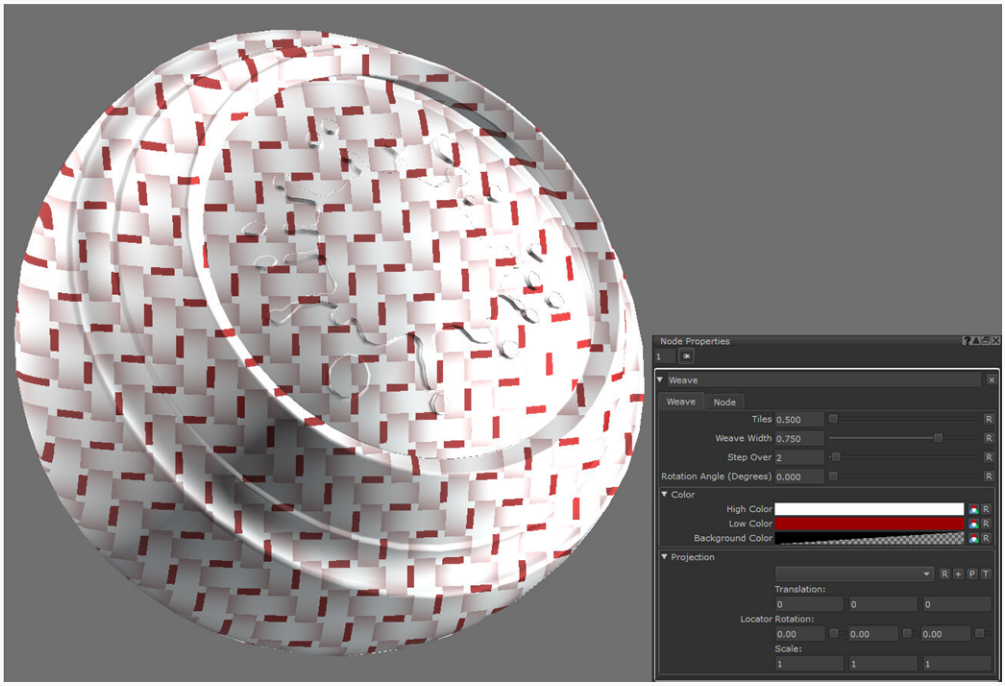
Step Over: 3



Step Over: 5

Color

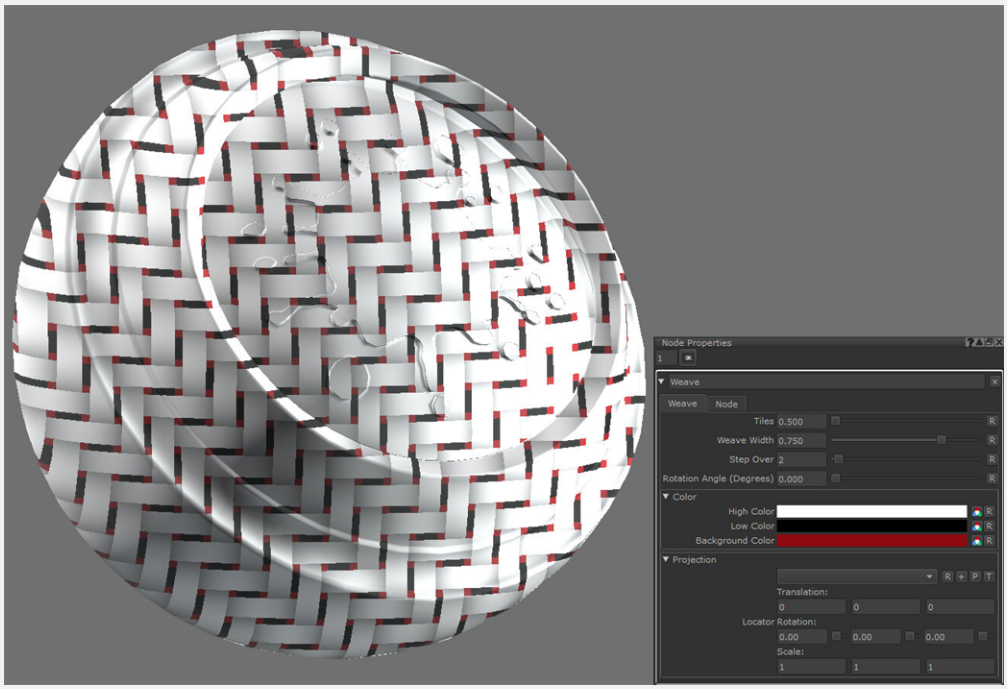
High Color *color select*	This is the color of a strip when it is over another strip. 
Low Color *color select*	This is the color of a strip when it is under another strip.



Background Color

color select

This defines the color that is used for the spaces in between strips.

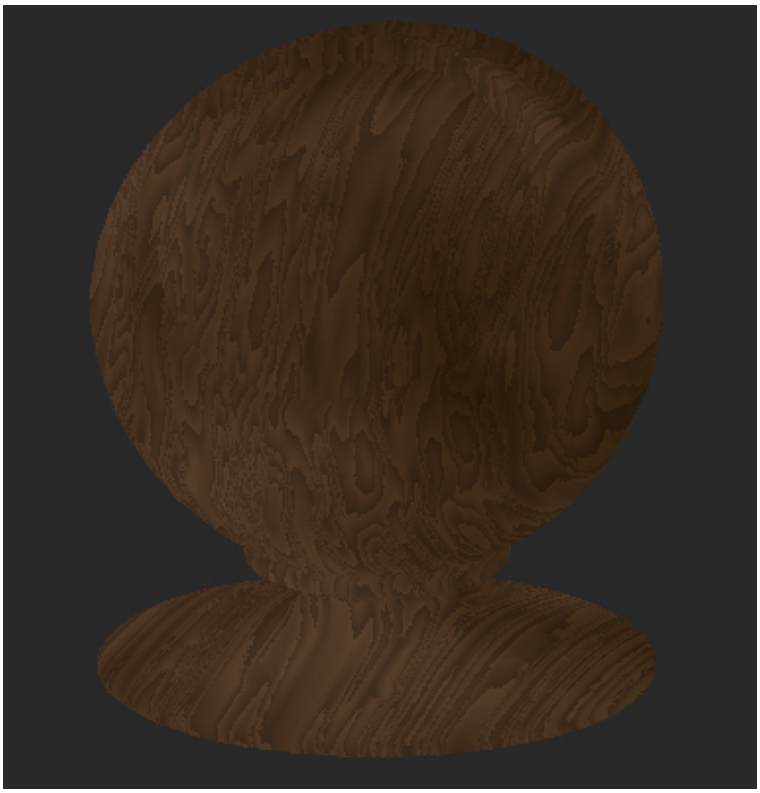


Projection

Projector *dropdown list*	This allows you to select a projector to align and scale the projected pattern. The list displays all the objects, locators, cameras and lights in the scene which can be used as a projector.
+ *button*	Add a new locator and set it as the projector.
P *button*	Make the projector locator current and switch to the Transform Object tool .
T *button*	Toggle the visibility of the transform fields.
Translation *text field*	Used to translate the projection in the x, y and z axis. **Note:** The transform fields (**Translation**, **Locator Rotation**, **Scale**) will lock if you have something selected in the field. The transform field values will mirror those of whatever you have selected.
Locator Rotation *text field*	Used to rotate the projection around the x, y and z axis.
Scale *text field*	Used to scale the projection in the x, y and z axis.

Wood Node

Access: **Nodes** > **Procedural** > **Plant** > **Wood**



A procedural wood texture built using the Wood node as the base node.

The Wood node is a procedural noise which emulates the grain of wood. It is a powerful procedural noise and has quite a few options to help you get the exact result you are after.

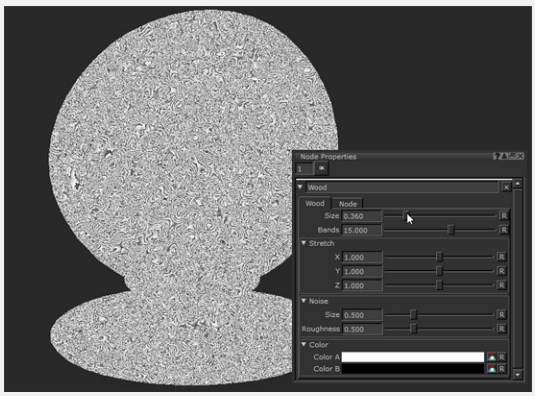
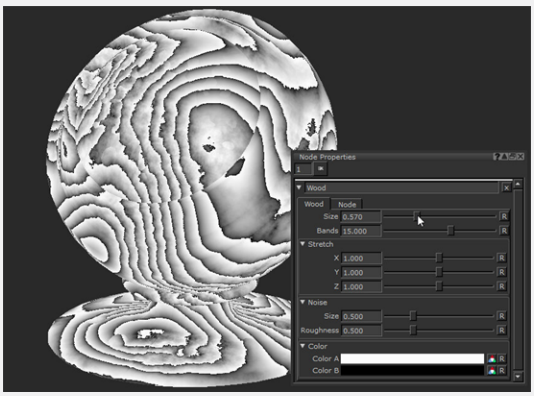
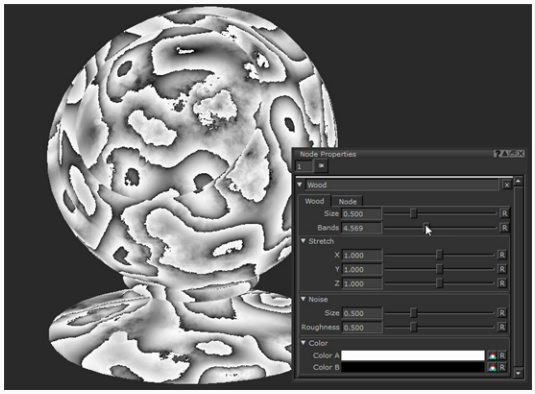
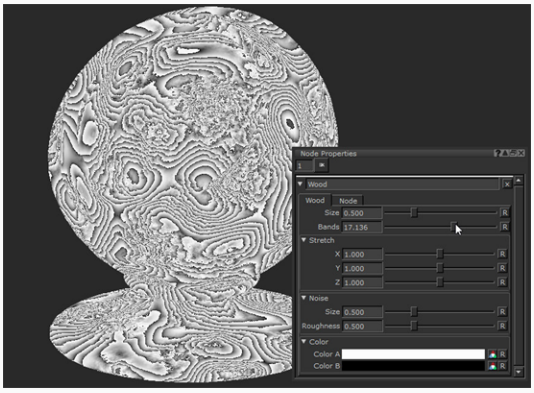
While creating a wooden effect is the most obvious use of this node, due to its striated nature it can be a great way to create other effects such as emulating layers of rock on a cliff texture or ripples in a sandy floor.

It is also useful for merging in organic detail to textures or masks. Since it uses 3D space by default, if an object is updated, even if the UVs are changed, the noise is still applied in a similar manner causing no repainting to be needed.

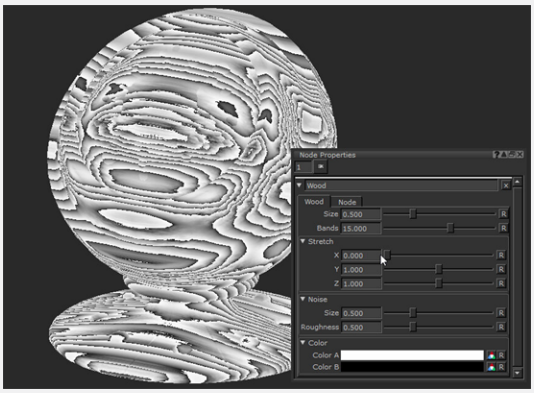
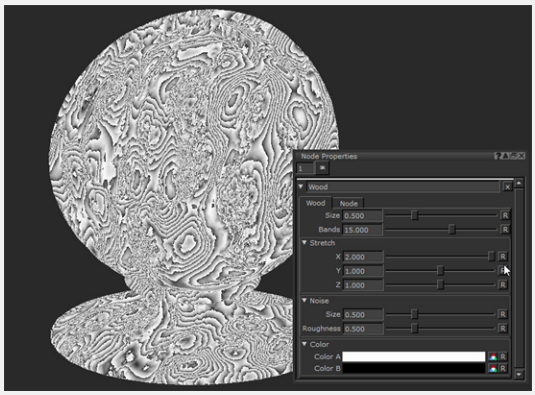
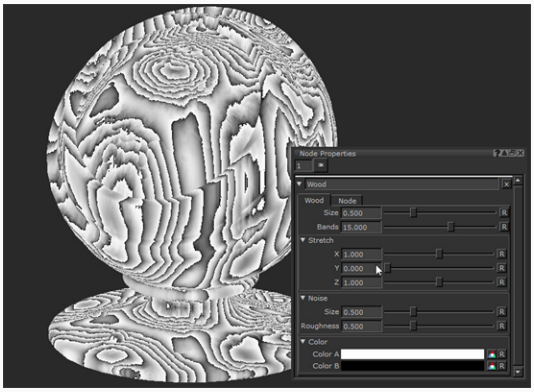
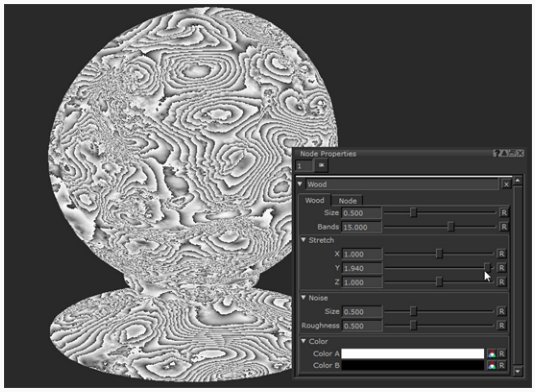
Wood Node Inputs

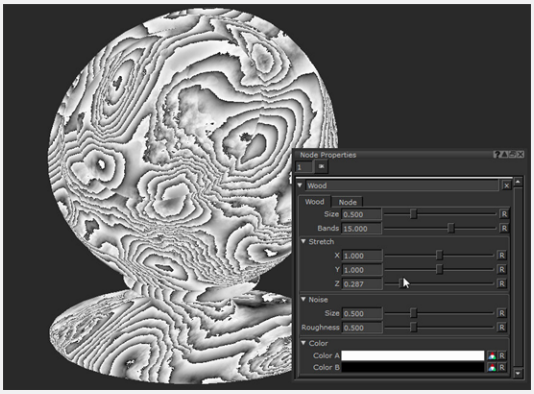
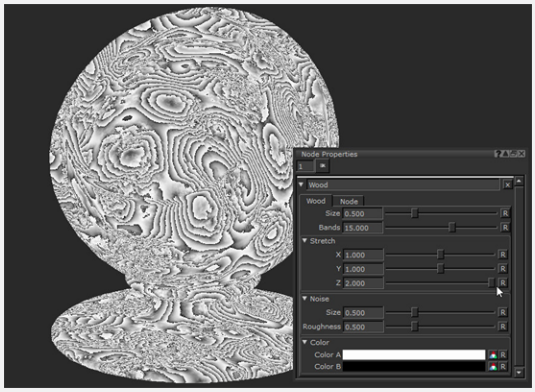
Position: The node works in a 3D space, but by adding a node to the **Position** input you can manipulate how the wood noise is applied. Other nodes could be the UV node to use a UV space rather than 3D space, or the position node.

Wood Node Properties

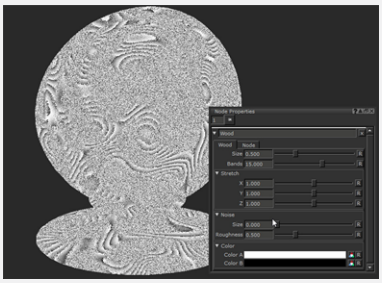
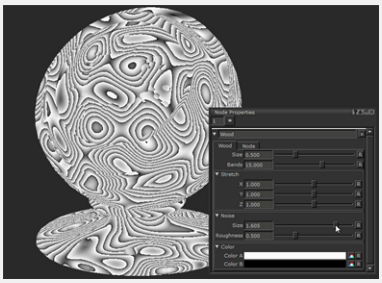
Size *text field, slider*	Changes the pattern frequency of the overall noise.  **Size** value set to 0.3.  **Size** value set to 0.5. Default is 0.5.
Bands *text field, slider*	Changes the number of wooden rings and bands in the noise. Raising this number creates more frequent wood ring effect.  **Bands** value set to 4.5.  **Bands** value set to 17. Default is 15.

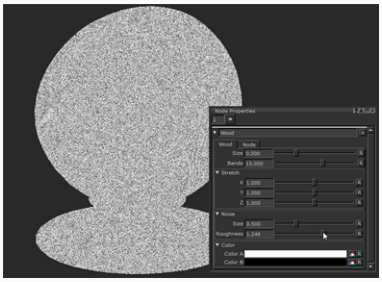
Stretch

X *text field, slider*	Changes the pattern scale in the X axis. Raising the value creates a more squashed up pattern whereas lowering it stretches it along the X axis.  Stretch **X** value set to 0.  Stretch **X** value set to 2. Default is 1.
Y *text field, slider*	Changes the pattern scale in the Y axis. Raising the value creates a more squashed up pattern whereas lowering it stretches it along the Y axis.  Stretch **Y** value set to 0.  Stretch **Y** value set to 1.9. Default is 1.
Z *text field, slider*	Changes the pattern scale in the Z axis. Raising the value creates a more squashed up pattern whereas lowering it stretches it along the Z axis.

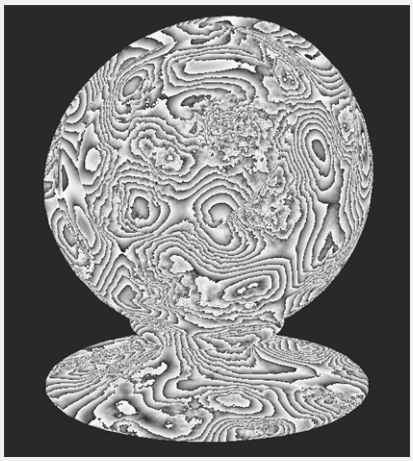
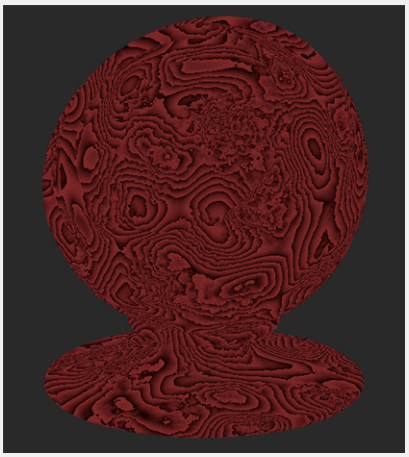
	 Stretch **Z** set to 0.2.  Stretch **Z** set to 2. Default is 1.

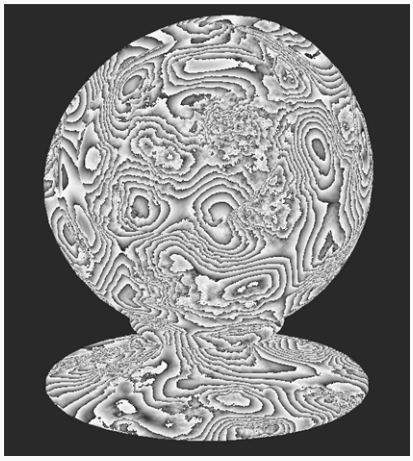
Noise

Size *text field, slider*	Changes the pattern frequency of the noise through the bands.  Noise **Size** value set to 0.  Noise **Size** value set to 1.6. Default is 0.5.
Roughness *text field, slider*	Changes how granular the noise gets through the procedural.

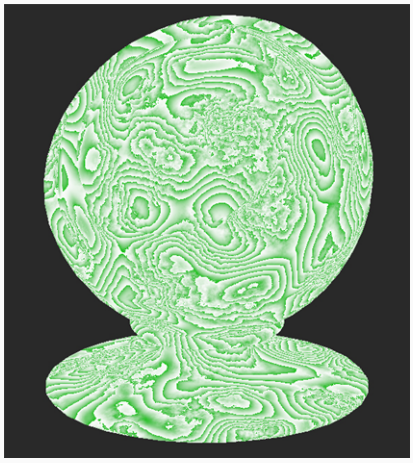
	Noise **Roughness** value of 0.1.  Noise **Roughness** value of 1.2. Default is 0.5.

Color

Color A *swatch*	Changes the first color of the noise.  Default: **Color A** set to black.  **Color A** set to green.
Color B *swatch*	Changes the color on top of **Color A**.



Default: **Color B** set to black.



Color B set to green.

Wood Node Workflow Example

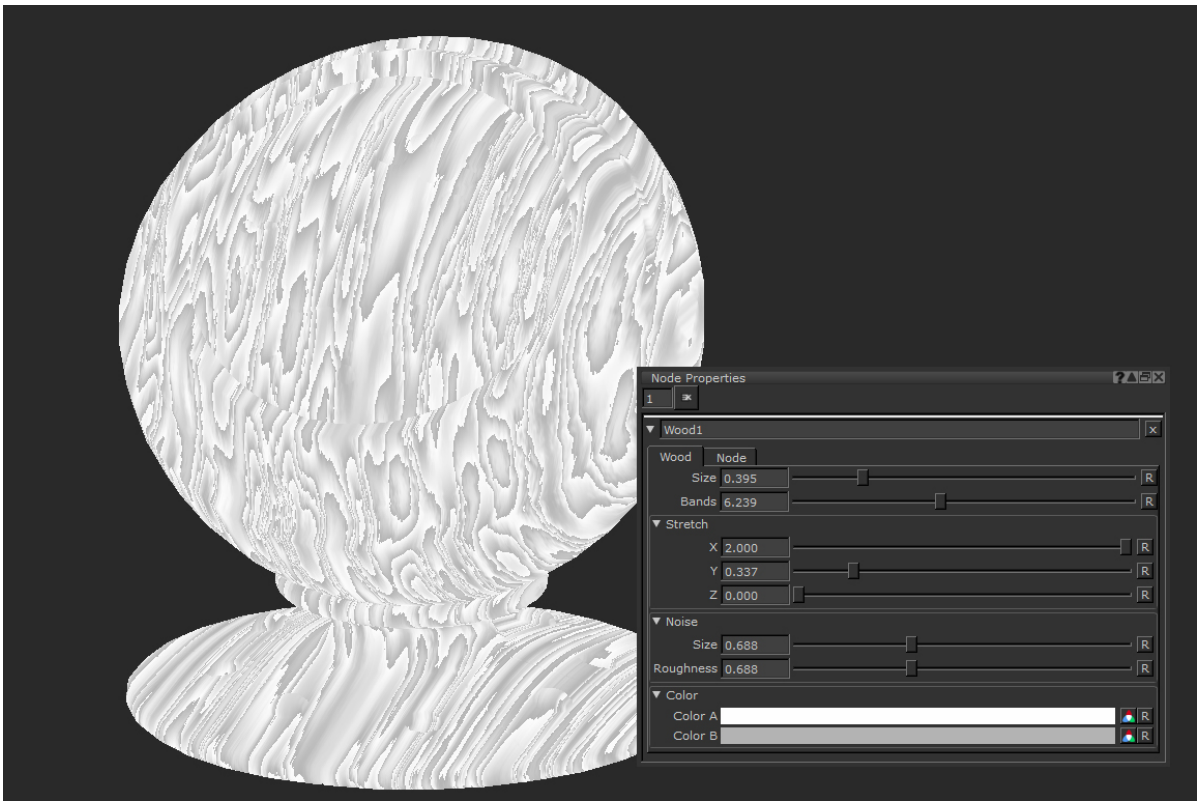
Creating a Procedural Wood Texture

Using noises is a great starting place to create procedural texture graphs. Since they don't rely on UVs, just 3D space, if your model needs a different UV layout or gets new geometry added it doesn't matter, the textures remain the same. Also you can share your nodes across multiple assets regardless of their UVs.

In this example we're going to create a quick node based wood color texture that can be used on any asset as it is procedural.

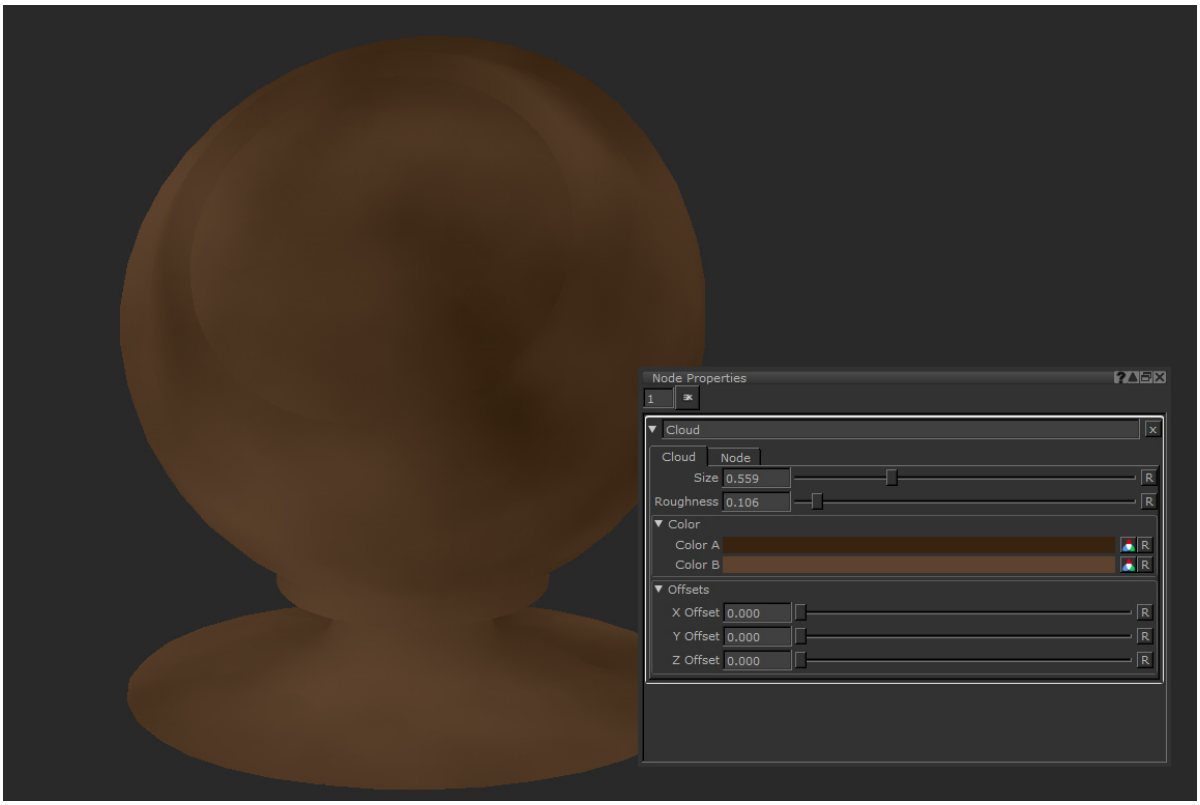
First, place down the Wood node. This is going to be the base of the texture.

By adjusting the properties you can make the grain of the Wood node stretch out in any axis, and by playing with the noise options you can either make a smooth looking wood or a gritty old style grain. By reducing the black of **Color B** to a lighter gray, the final result is going to be less contrasted like the wood grain of a polished table top.



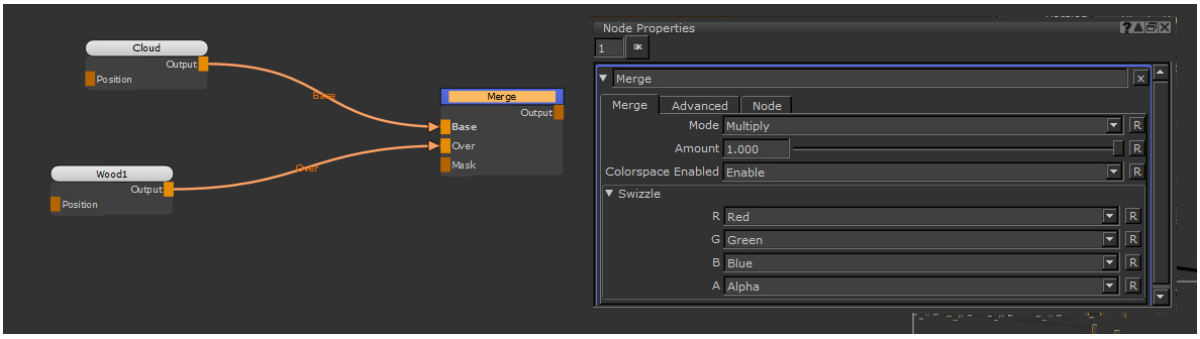
This node informs the pattern of the wood grain.

From here we want to add some color. While we could change the **Color** properties of the Wood node, by using a Merge node with a different procedural you give yourself more space to add variety. A Cloud node is a great node for getting soft variation over a large area. By dropping one down and reducing the **Roughness** you get soft changes in color. Let's also chose some colors closer to wood.



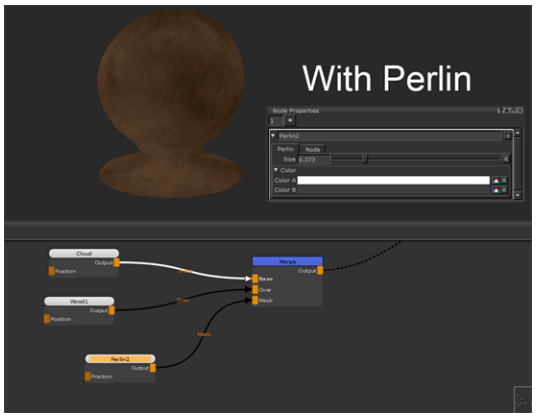
This node informs the color of our wood texture.

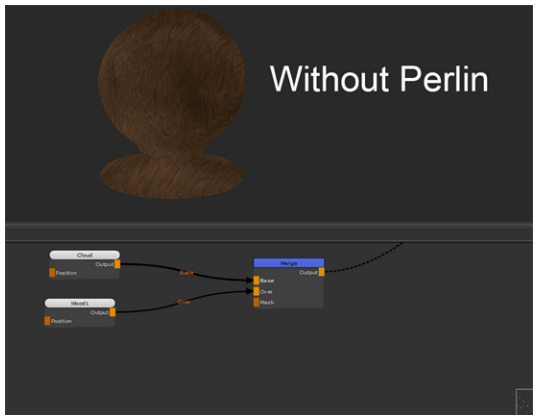
We need to merge these together next. A **Multiple** blend mode means we just get the darkness of the Wood node which emulates wood grain nicely. Make sure the Wood node is being multiplied on top of the colored Cloud by being plugged in the **Over** input.



Multiplying the wood grain over the color.

To add an extra level of realism, a Perlin node can be used as the mask. The result this has means the Wood node does not multiply evenly, in some places the effect is less obvious, breaking up the CG look which is always best avoided when texturing.

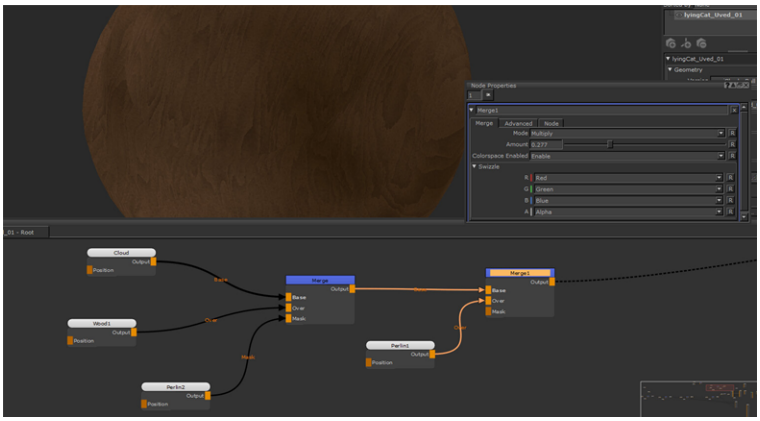




The addition of a masks help add some breakup. If you feel it's affect is too strong you can play with the colors of the Perlin node.

Let's add one final node for more detail. Wood often has a lot of little dots that go with the grain that the base Wood node isn't capturing. We can use another noise to add this effect.

By dropping down another Perlin node and changing the **Size** to a value of **0.1** in the **Node Properties**, you can get a small organic spotty effect. Using another Merge set to **Multiply** means we can add this effect on top of our current texture. This time amount seems a little high so let's bring it down to make it a bit more subtle.



Changing the Merge node's **Amount** property makes the grain effect more subtle.

And there we have it, a very quick and easy procedural wood texture.

The final result.

This can be used on any object now as the UVs do not matter with procedural noises. The other great thing is if you want to have a rougher grain, or a different colored wood, it's all really easy to change with no repainting or baking.

Camera Distance Gradient Node

Access: **Nodes** > **Projection** > **Camera Distance Gradient**

The Camera Distance Gradient node creates a gradient based on the angle of a designated projector in the scene. You can control the gradient between the near and far colors using the **Distance Blend** controls and assign multiple projectors to the node to create as many gradients as you require from any angle.

The default gradient created by a projector on the left side of an object.

The same gradient viewed from above the object.

Camera Distance Gradient Node Inputs

Near Color	Sets the start color of the gradient, which is applied to geometry closest to the selected projector.
Far Color	Sets the end color of the gradient, which is applied to geometry farthest away from the selected projector.
Position	An optional input to set the start coordinates of the gradient, measured from the value of the input to the locator/position specified in the node properties **Distance Blend: Start** control.

Camera Distance Gradient Node Outputs

Output	A gradient based on the angle of a designated projector in the scene in relation to geometry.

Camera Distance Gradient Node Properties

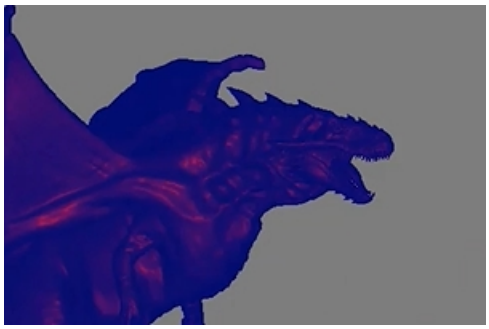
Near Color *swatch*	Sets the default start color of the gradient if nothing is connected to the **Near Color** input.

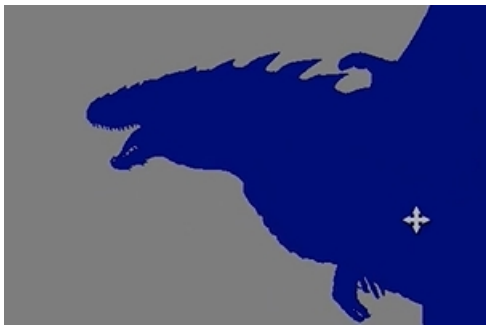
Far Color *swatch*	Sets the default end color of the gradient if nothing is connected to the **Far Color** input.
Distance Blend: Start *numeric field, slider*	Sets the point at which the gradient blend begins. The default 0.0 causes the gradient to blend towards the **Far Color** immediately.
Distance Blend: Range *numeric field, slider*	Sets the point at which the gradient blend ends. The default 10.0 causes the gradient to be equal to the **Far Color** after 10 units, which depends on the scale of your imported geometry.
Projector *dropdown*	Selects the projector to use as the basis for the angled gradient. You can use one of the standard projection angles, **Perpective**, **Othographic**, or **UV**, or create a custom projector by pressing the **+** button.
+	Click to add a custom projector to the scene to control the gradient applied to your geometry. See Projectors Palette for more information on projectors in Mari.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.

Camera Facing Gradient Node

Access: **Nodes** > **Projection** > **Camera Facing Gradient**

The Camera Facing Gradient node creates a gradient based on the facing of geometry relative to a projector in the scene. You can control the gradient between the start and end colors using the **Facing Blend** controls and assign multiple projectors to the node to create as many gradients as you require from any angle.





A red to blue gradient where full red is facing the camera and full blue is facing away from the camera.

Camera Facing Gradient Node Inputs

Start Color	Sets the start color of the gradient, which is applied to geometry facing the selected projector.
End Color	Sets the end color of the gradient, which is applied to geometry facing away from the selected projector.
Normal	An optional input to feed in a normal map to determine geometry facing.

Camera Facing Gradient Node Outputs

Output	A facing gradient calculated from normals in relation to a selected projector.

Camera Facing Gradient Node Properties

Start Color *swatch*	Sets the default start color of the gradient if nothing is connected to the **Start Color** input.
End Color *swatch*	Sets the default end color of the gradient if nothing is connected to the **End Color** input.
Facing Blend: Start *numeric field, slider*	Sets the facing angle at which the gradient blend begins. Facing angles greater than this value blend towards the **End Color**.
Facing Blend: End *numeric field, slider*	Sets the facing angle at which the gradient blend ends. Facing angles less than this value blend towards the **Start Color**.
Absolute *checkbox*	When enabled, all negative facing values are considered positive and given the same value.

Projector *dropdown*	Selects the projector to use as the basis for the facing gradient. You can use one of the standard projection angles, **Perpective**, **Othographic**, or **UV**, or create a custom projector by pressing the **+** button.
+	Click to add a custom projector to the scene to control the gradient applied to your geometry. See Projectors Palette for more information on projectors in Mari.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.

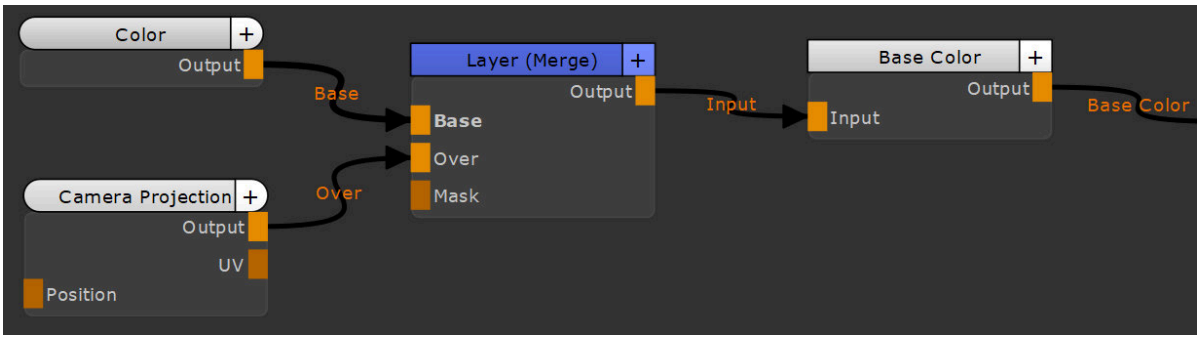
Camera Projection Node

Access: **Nodes** > **Projection** > **Camera Projection**

This node projects an image through a camera onto the surface of a mesh in world space. The image projection's origin and orientation can be linked to a camera.

The difference between the Camera Projection node and the Projection node is that the Projection node uses the transform values (position, rotation, scale) of the camera to project the image whilst the Camera Projection node fully projects the image through the camera as well as taking perspective into account.

This node can be useful if you would like to project a texture from a particular angle as you can use a camera to project an image onto an object and then use a Bake Point node to cache it and create a texture which won't be affected by the camera's transform values.



Example Node Network for Camera Projection Node

Camera Projection Example

Camera Projection Node Inputs

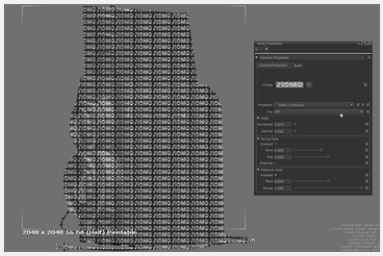
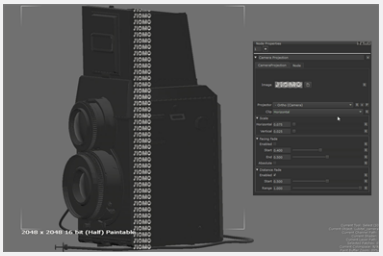
Input	Description
Position	Assigns the input position to the position of the projection.

Camera Projection Node Outputs

Output	Description
Output	Assigns the projection output to the input of another node. For example, the **Base** or **Over** inputs of a Merge node or the **Input** of a Channel node.
UV	Outputs the UV coordinates that are in use when projecting the image so they can be used repeatedly when sampling multiple images.
Masks	Contains the various mask values packed together. When output to specific RGBA

	channels, this is how the output is broken down: • R - Clipping mask • G - Distance mask • B - Facing mask • A - All masks multiplied together

Camera Projection Node Properties

Image *file upload*	Upload an image to project.
Projector *dropdown list*	Select the camera to use as the projector.
Clip *dropdown list*	This defines how the image is repeated across the projection. **Off** - The image is not clipped, therefore repeated vertically and horizontally. **Horizontal** - The image is clipped horizontally, therefore only repeated vertically. **Vertical** - The image is clipped vertically, therefore only repeated horizontally. **Both** - The image is clipped vertically and horizontally, therefore only shown once.  Clip: Off  Clip: Horizontal Clip: Vertical

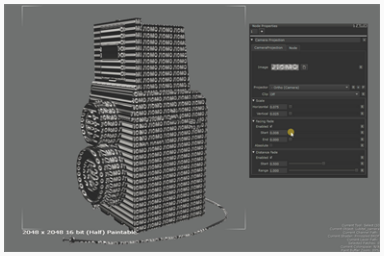
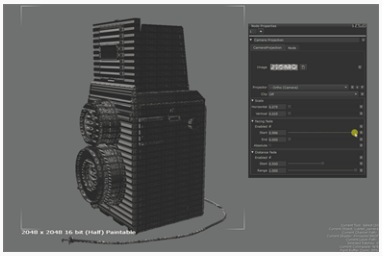
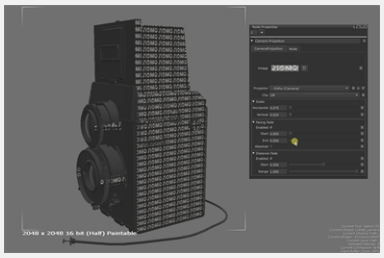
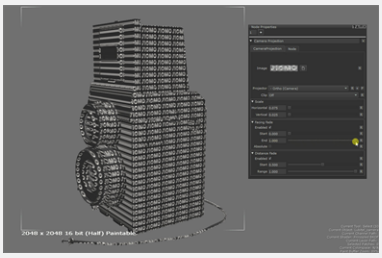
	 Clip: Both

Scale

Horizontal *text field, slider*	Scale the image horizontally.
Vertical *text field, slider*	Scale the image vertically.

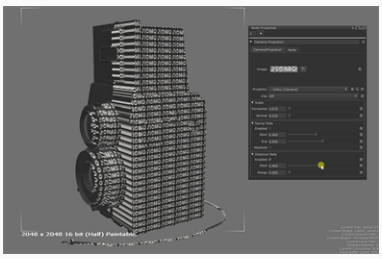
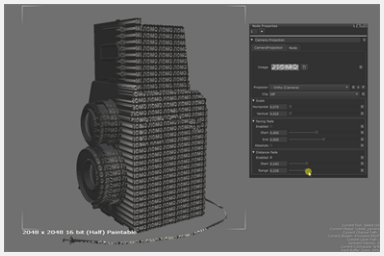
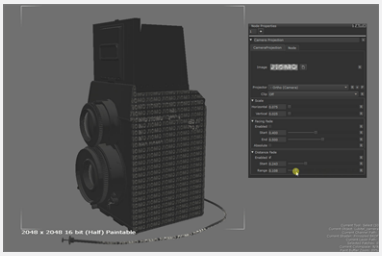
Facing Fade

Enabled *checkbox*	This option determines whether the image fades out based on how much it faces towards the direction of projection.
Start *text field, slider*	This defines at what point the image starts to fade out based on the amount that the image is facing the direction of projection. A value of **0** means the image starts to fade out if it is facing the direction of projection. A value of **0.5** means the image starts to fade out if is it facing perpendicular to the projection direction. A value of **1** means the image starts to fade out if it is facing away from the projection direction.

	 Start: 0.008  Start: 0.996
End *text field, slider*	This defines at what point the image is entirely faded out based on the amount that the image is facing the direction of projection. A value of **0** means the image is entirely faded out if it is facing the direction of projection. A value of **0.5** means the image is entirely faded out if is it facing perpendicular to the projection direction. A value of **1** means the image is entirely faded out if it is facing away from the projection direction.  End: 0.05  End: 1
Absolute *checkbox*	This makes the facing value absolute.

Distance Fade

Enabled *checkbox*	This option determines whether the image fades out based on the distance to the origin or the projector.
Start	This determines the distance from the projector origin at which the image begins to

text field, slider	fade out. Start: 0.344  Start: 0.469
Range *text field, slider*	This determines the distance beyond the Start value at which the image has faded out entirely.  Range: 0.228  Range: 0.108

Object Distance Gradient Node

Access: **Nodes** > **Projection** > **Object Distance Gradient**

The Object Distance Gradient node creates a gradient based on a locator's proximity to your geometry. You can control the gradient between the near and far colors using the **Distance Blend** controls.

With the locator far from the geometry , the Far Color is used.

The same locator translated to nearer the geometry produces the gradient.

Object Distance Gradient Node Inputs

Near Color	Sets the start color of the gradient, which is applied to geometry closest to the locator.
Far Color	Sets the end color of the gradient, which is applied to geometry farthest away from the locator.
Position	An optional input to set the start coordinates of the gradient, measured from the value of the input to the locator/position specified in the node properties **Distance Blend: Start** control.

Object Distance Gradient Node Outputs

Output	A distance gradient calculated from locator position in relation to geometry.

Object Distance Gradient Node Properties

Distance from Locator	Sets how the distance from the locator to the geometry is calculated, either from the center, along a single axis, or on two axes. **Note:** If you select a plane option, YZ, XZ, or XY, you can calculate absolute distance rather than relative distance to the geometry by enabling the **Absolute** checkbox.
Near Color *swatch*	Sets the default start color of the gradient if nothing is connected to the **Near Color** input.
Far Color *swatch*	Sets the default end color of the gradient if nothing is connected to the **Far Color** input.

Distance Blend: Start *numeric field, slider*	Sets the point at which the gradient blend begins. The default 0.0 causes the gradient to blend towards the **Far Color** immediately.
Distance Blend: Range *numeric field, slider*	Sets the point at which the gradient blend ends. The default 10.0 causes the gradient to be equal to the **Far Color** after 10 units, which depends on the scale of your imported geometry.
Absolute *checkbox*	When enabled, all negative distances are considered positive. For example, a point 5m "behind" the selected plane is given the same value as a point 5m in "front" of the selected plane. **Note: Absolute** distance only applies when a plane,YZ, XZ, or XY, is selected from the **Distance from Locator** dropdown.
Object *dropdown*	Selects the object to use as the basis for the distance gradient. You can use one of the standard options, objects, cameras, and lights, or create a custom locator by pressing the **+** button.
+	Click to add a locator to the scene to control the gradient applied to your geometry.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.
T	Click to show and hide the transform controls at the bottom of the panel.

Object Facing Gradient Node

Access: **Nodes** > **Projection** > **Object Facing Gradient**

The Object Facing Gradient node creates a gradient based on object facing in relation to a locator. You can control the gradient between the start and end colors using the **Facing Blend** controls and assign multiple projectors to the node to create as many gradients as you require from any angle.

The gradient changes dependent on geometry facing in relation to a locator.

Object Facing Gradient Node Inputs

Start Color	Sets the start color of the gradient, which is applied to geometry closest to the locator.
End Color	Sets the end color of the gradient, which is applied to geometry farthest away from the locator.
Position	An optional input to set the start coordinates of the gradient.
Normal	An optional input to feed in a normal map to determine geometry facing.

Object Facing Gradient Node Outputs

Output	A facing gradient calculated from locator position in relation to geometry.

Object Facing Gradient Node Properties

Mode *dropdown*	Sets how the facing from the geometry to the locator is calculated, either planar on two axes or **3D**. **3D** mode gives you full control over the locator to produce the gradient you require.
Start Color	Sets the default start color of the gradient if nothing is connected

swatch	to the **Start Color** input.
End Color *swatch*	Sets the default end color of the gradient if nothing is connected to the **End Color** input.
Facing Blend: Start *numeric field, slider*	Sets the facing angle at which the gradient blend begins. Facing angles greater than this value blend towards the **End Color**.
Facing Blend: End *numeric field, slider*	Sets the facing angle at which the gradient blend ends. Facing angles less than this value blend towards the **Start Color**.
Absolute *checkbox*	When enabled, all negative facing values are considered positive and given the same value.
Object *dropdown*	Selects the object to use as the basis for the facing gradient. You can use one of the standard options, objects, cameras, and lights, or create a custom locator by pressing the **+** button.
+	Click to add a locator to the scene to control the gradient applied to your geometry.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.
T	Click to show and hide the transform controls at the bottom of the panel.

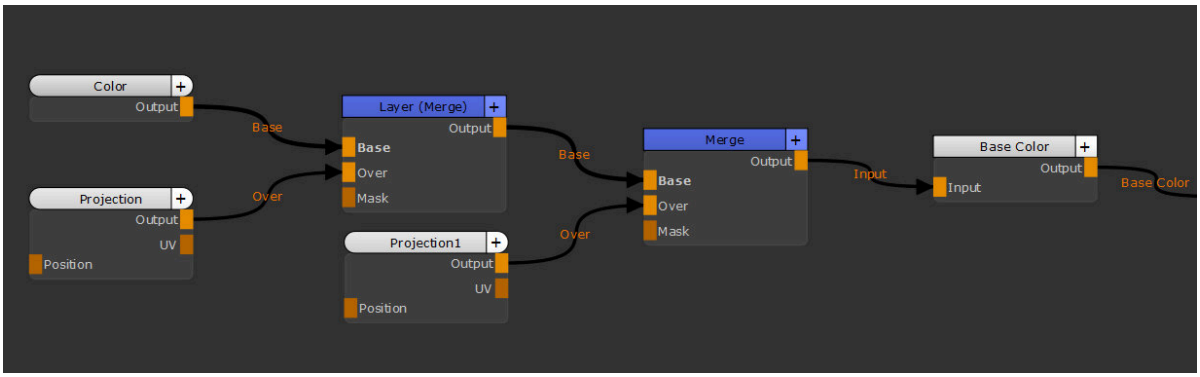
Projection Node

Access: **Nodes** > **Projection** > **Projection**

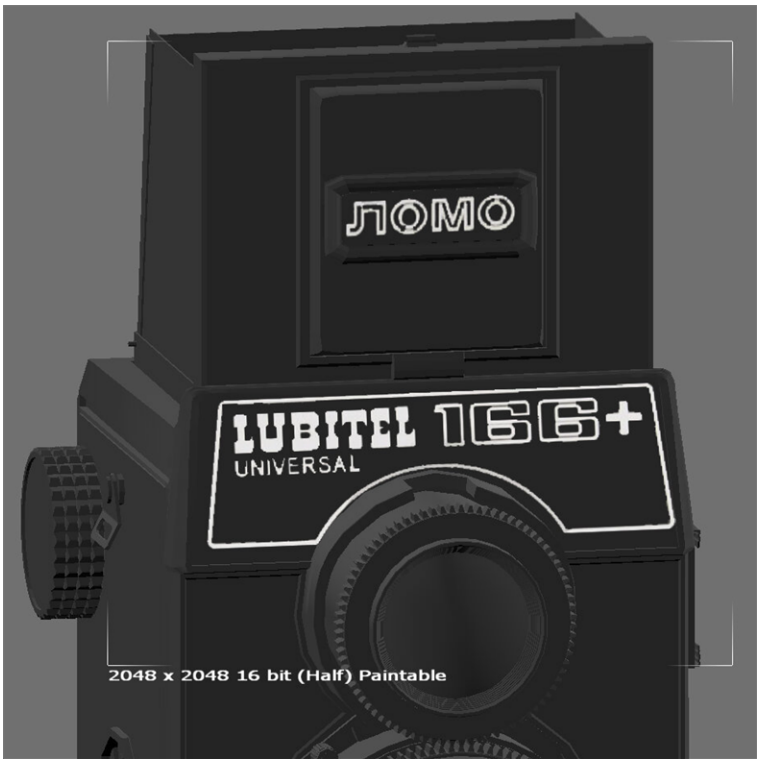
The Projection node is used to project an image onto the surface of an object in world space. The node can either use planar, cylindrical or spherical projection and allows you to use any object, camera, locator or light in your scene as the projector.

This node can be especially useful for placing an image in a specific position on an object. It avoids the need for the paint buffer and, if the image is being projected through a locator, it allows for continuous transformation using the **Transform Selected Objects** tool.

Because the image is being projected, moving the perspective or orthographic camera does not disturb the image's position unless the projector is set to **Perspective** or **Ortho**.



Example Node Network for Projection Node



Planar Projection Example

Projection Node Inputs

Input	Description
Position	Assigns the input position to the position of the projection.

Projection Node Outputs

Output	Description
Output	Assigns the Projection output to the input of another node. For example, the **Base** or **Over** inputs of a Merge node or the **Input** of a Channel node.
UV	Outputs the UV coordinates that are in use when projecting the image so they can used repeatedly when sampling multiple images.
Masks	Contains the various mask values packed together. When output to specific RGBA channels, this is how the output is broken down: • R - Clipping mask • G - Distance mask • B - Facing mask • A - All masks multiplied together

Projection Node Properties

Image *file upload*	Upload an image to project.
Projection Mode *dropdown list*	This defines how the image is projected onto the surface. **Planar** - The image is projected onto the XY plane defined by the projector. **Cylindrical** - The image is projected cylindrically, with the cylinder axis aligned along projector's Y axis. **Spherical** - The image is projected spherically, with the poles aligned along projector's Y axis.
Clip *dropdown list*	This defines how the image is repeated across the projection. **Off** - The image is not clipped, therefore repeated vertically and horizontally. **Horizontal** - The image is clipped horizontally, therefore only repeated vertically.

	Vertical - The image is clipped vertically, therefore only repeated horizontally. **Both** - The image is clipped vertically and horizontally, therefore only shown once.

Scale

Horizontal *text field, slider*	Scale the image horizontally.
Vertical *text field, slider*	Scale the image vertically.

Facing Fade

Enabled *checkbox*	This option determines whether the image fades out based on how much it faces towards the direction of projection.
Start *text field, slider*	This defines at what point the image starts to fade out based on the amount that the image is facing the direction of projection. A value of **0** means the image starts to fade out if it is facing the direction of projection. A value of **0.5** means the image starts to fade out if is it facing perpendicular to the projection direction. A value of **1** means the image starts to fade out if it is facing away from the projection direction.
End *text field, slider*	This defines at what point the image is entirely faded out based on the amount that the image is facing the direction of projection. A value of **0** means the image is entirely faded out if it is facing the direction of projection.

A value of **0.5** means the image is entirely faded out if is it facing perpendicular to the projection direction.

A value of **1** means the image is entirely faded out if it is facing away from the projection direction.

An **End** value of 0.07.

An **End** value of 0.3.

	 An **End** value of 0.6.
Absolute *checkbox*	This makes the facing value absolute.

Distance Fade

Enabled *checkbox*	This option determines whether the image fades out based on the distance to the origin or the projector.
Start *text field, slider*	This determines the distance from the projector origin at which the image begins to fade out.
Range *text field, slider*	This determines the distance beyond the **Start** value at which the image has faded out entirely.

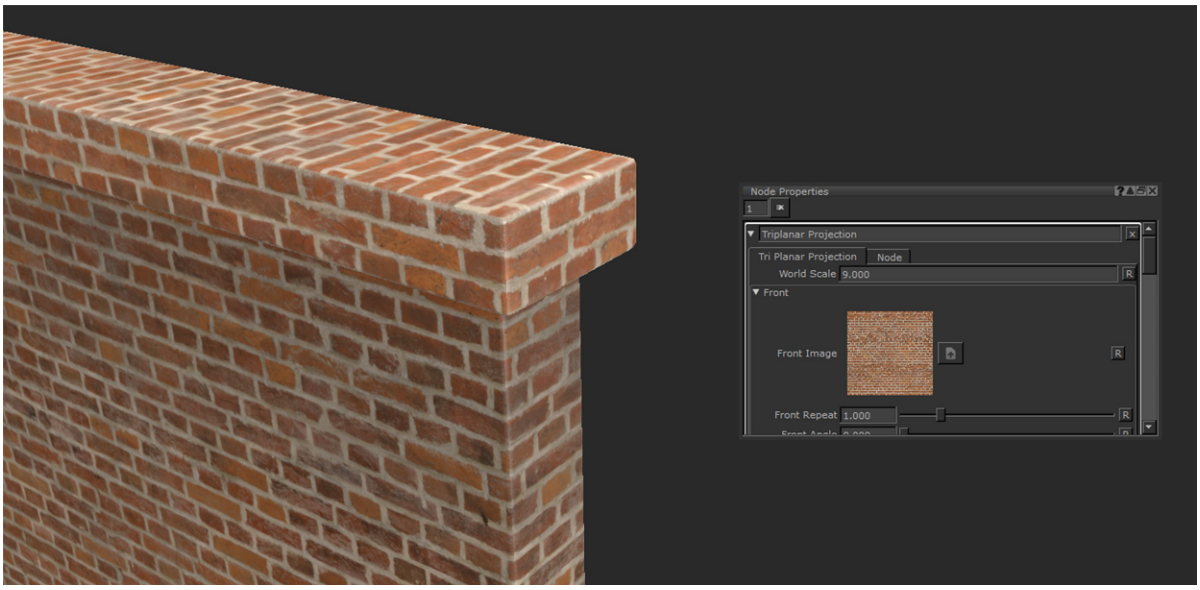
Projection

Projector *dropdown list*	Select what the image is projected from. The list displays all the Objects, Locators, Cameras and Lights in the scene which can be used as a projector.

+ *button*	Add a new locator and set it as the projector.
P *button*	Make the projector locator current and switch to the **Transform Object** tool.
T *button*	Toggle the visibility of the transform fields.
Translation *text field*	Used to translate the projection in the x, y and z axis. **Note:** The transform fields (**Translation**, **Projector Rotation**, **Scale**) lock if you have something selected in the field. The transform field values mirror those of whatever you have selected.
Projector Rotation *text field*	Used to rotate the projection around the x, y and z axis.
Scale *text field*	Used to scale the projection in the x, y and z axis.

Triplanar Projection Node

Access: **Nodes** > **Projection** > **Triplanar Node**



Using the Triplanar Projection node to quickly texture hard surface assets with poorly laid out UVs.

The Triplanar Projection node is a node which allows you project up to three different images from the three different axis of your mesh in 3D space.

It is similar to the Tiled node however it does not rely on an objects UVs, so it is great for hard surface texturing, such as bricks or tiles, which would look odd if there was UV stretching or squashing. Unlike the Tiled node, it allows multiple images at once giving you a greater level of control.

The Triplanar Projection node can also be great if a Tiled node is giving obvious seams at UV shell borders.

Tip: In some instances a Tiled node may be better on curved surfaces, as a Triplanar Projection node can give strange and blurred results in those scenarios.

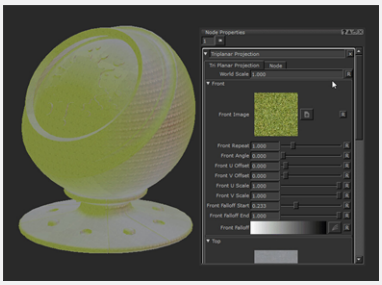
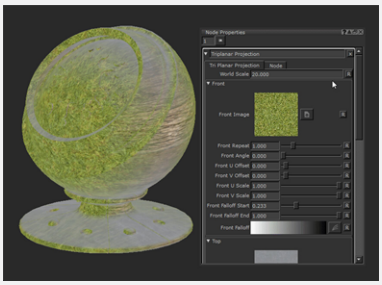
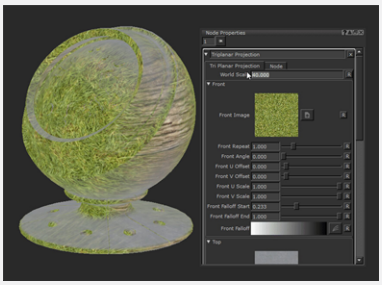
Triplanar Projection Node Inputs

Position: The node works in 3D space, but by adding a node to the **Position** input you can manipulate how the projection is applied. Examples include the UV node to use a UV space rather than 3D space. This helps if you want to apply three different images to different axis but still retain UV direction, stretch, and squash.

Normal:Overrides the surface normals used to define how the top front side projections are mapped to the object.

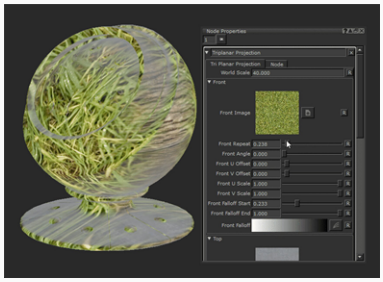
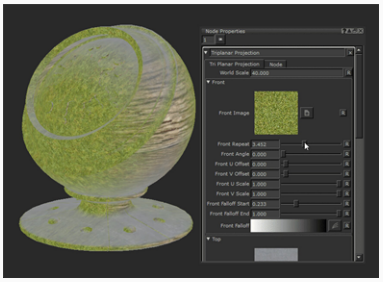
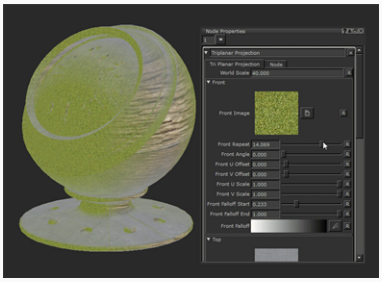
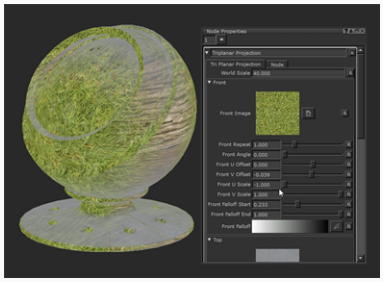
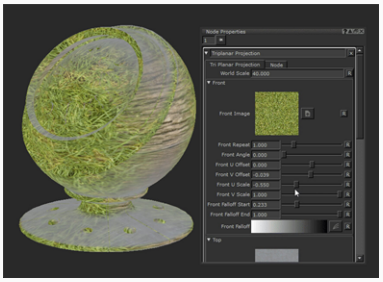
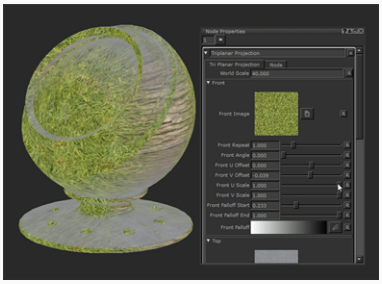
Triplanar Projection Node Properties

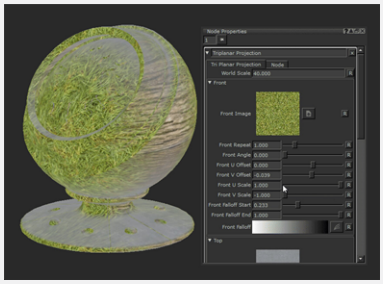
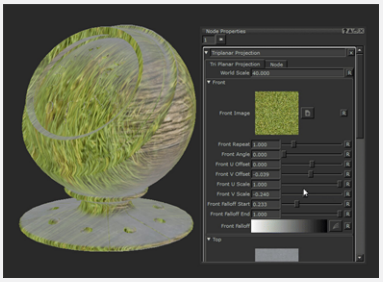
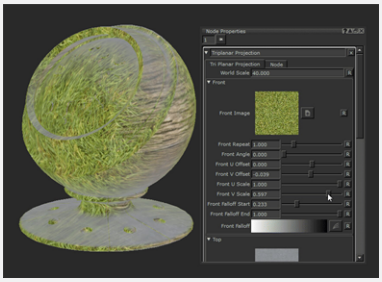
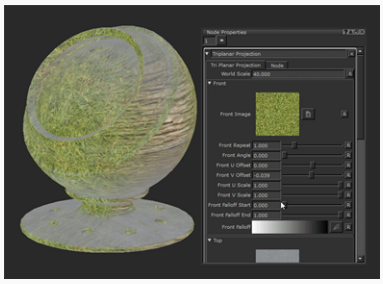
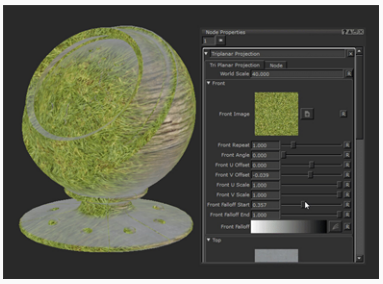
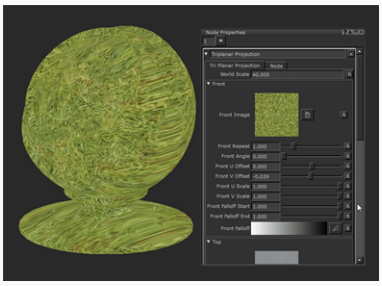
The following example images will be using a different image on each axis for demonstration purposes.

World Scale *text field*	Changes the overall scale of all three projected images at once - front, top, and right. This saves you having to change the repeat amount of all the images individually if you want to do a uniform scale of the projections.  **World Scale** value set to 1.  **World Scale** value set to 20.  **World Scale** value set to 40. Default is 4.

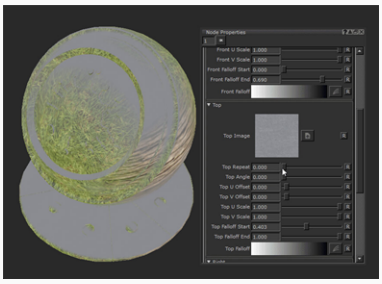
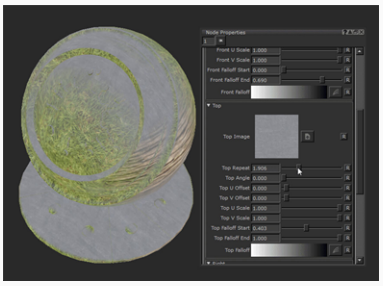
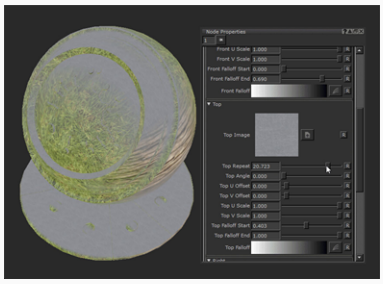
Front

Front Image *image*	Chooses the image to project across the front of your geometry.
Front Repeat *text field, slider*	Changes the tiling amount of your projected image. Increasing this value makes the image scale down.

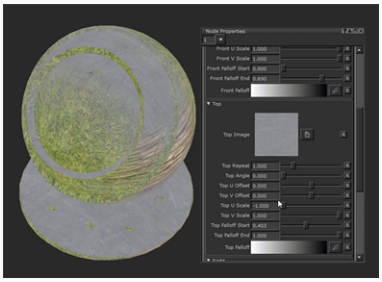
	   **Front Repeat** set to 0.2. **Front Repeat** set to 3.4. **Front Repeat** set to 14. Default is 0.
Front Angle *text field, slider*	Changes the rotation of the projected image. Default is 0.
Front U Offset *text field, slider*	Moves the tiled image horizontally along the U axis of the projection. Default is 0.
Front V Offset *text field, slider*	Moves the tiled image vertically along the V axis of the projection. Default is 0.
Front U Scale *text field, slider*	Scales the tiled image horizontally along the U axis of the projection. If this number is higher than the V Scale the image will start to elongate vertically.    **Front U Scale** set to -1. **Front U Scale** set to -0.5. **Front U Scale** set to 1. Default is 1.

Front V Scale *text field, slider*	Scales the tiled image vertically along the V axis of the projection. If this number is higher than the U Scale the image will start to elongate horizontally .    **Front V Scale** set to -1. **Front V Scale** set to -0.2. **Front V Scale** set to 0.5. Default is 1.
Front Falloff Start *text field, slider*	Changes the fall off of the front projection and the threshold at which angles the projection will cover.    **Front Falloff Start** set to 0. **Front Falloff Start** set to 0.3. **Front Falloff Start** set to 1. Raising this slider will make the projection cover more area on curved or glancing angles and may eventually overlap the other axes projections. Default is 0.
Front Falloff End *text field, slider*	Changes the fall off of the front projection and the threshold at which angles the projection will cover. Lowering this slider will make the projection cover less area on curved or glancing angles. Default is 1.
Front Falloff *curve editor*	This curve editor gives control of the falloff of the front projection - by manipulating the curve you can make the falloff harsher. This may be useful when texturing hard surface assets to avoid soft and blended textures.

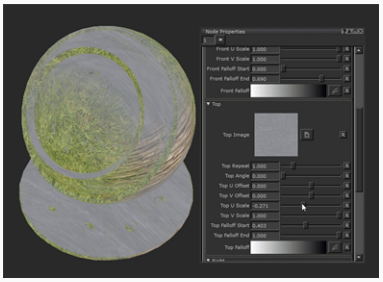
Top

Top Image *image*	Chooses the image to project across the top of your geometry.
Top Repeat *text field, slider*	Changes the tiling amount of your projected image. Upping this value will make the image scale down.  **Top Repeat** set to 0.  **Top Repeat** set to 1.9.  **Top Repeat** set to 20. Default is 0.
Top Angle *text field, slider*	Changes the rotation of the projected image. Default is 0
Top U Offset *text field, slider*	Moves the tiled image horizontally along the U axis of the projection. Default is 0.
Top V Offset *text field, slider*	Moves the tiled image vertically along the V axis of the projection. Default is 0.
Top U Scale	Scales the tiled image horizontally along the U axis of the projection. If this number is higher than the V Scale the image will start to elongate vertically.

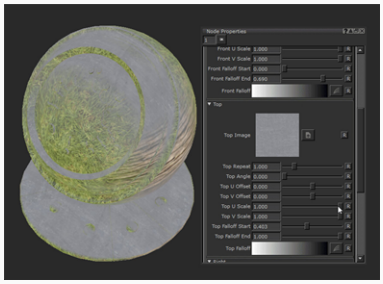
text field, slider



Top U Scale set to -1.



Top U Scale set to -0.2.



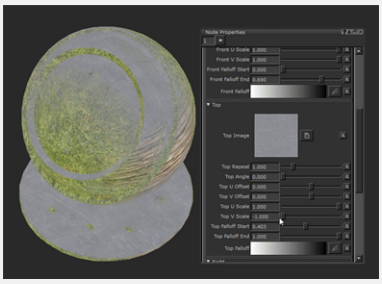
Top U Scale set to 1.

Default is 1.

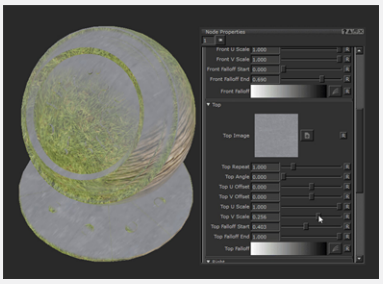
Top V Scale

text field, slider

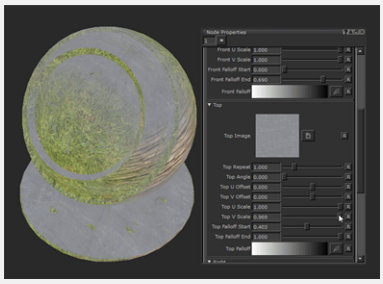
Scales the tiled image vertically along the V axis of the projection. If this number is higher than the U Scale the image will start to elongate horizontally.



Top V Scale set to -1.



Top V Scale set to 0.2.



Top V Scale set to 1.

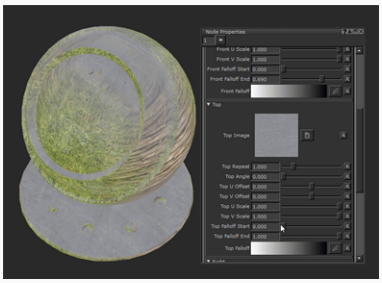
Default is 1.

Top Falloff Start

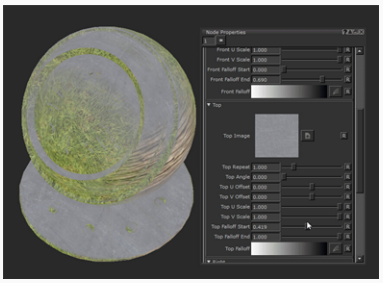
text field, slider

Changes the fall off of the top projection and the threshold at which angles the projection will cover.

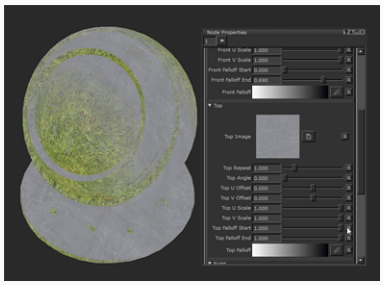
Raising this slider will make the projection cover more area on curved or glancing angles and may eventually overlap the other axes projections.



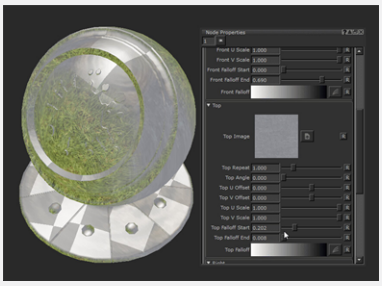
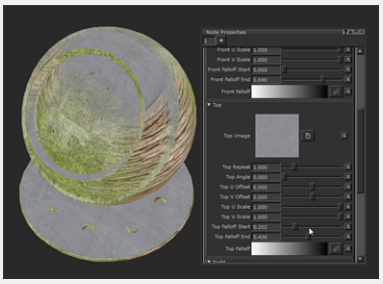
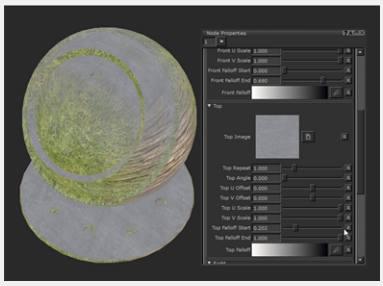
Top Falloff Start set to 0.



Top Falloff Start set to 0.4.

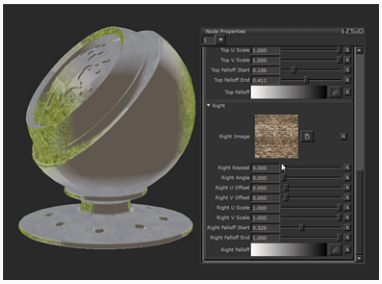
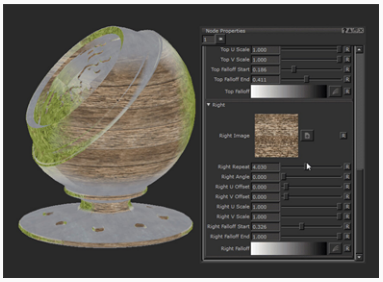
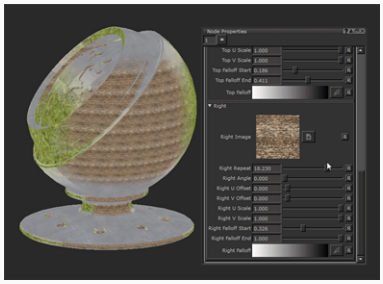
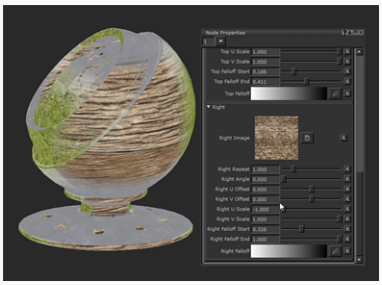
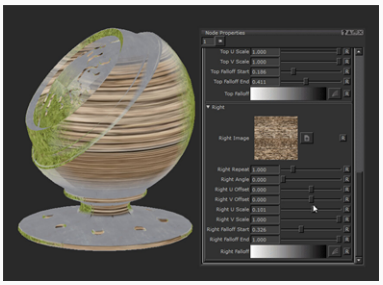
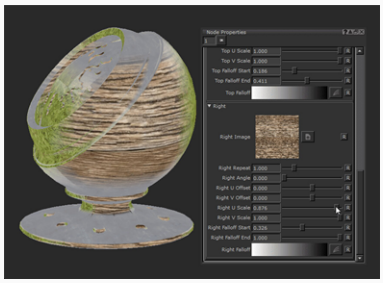


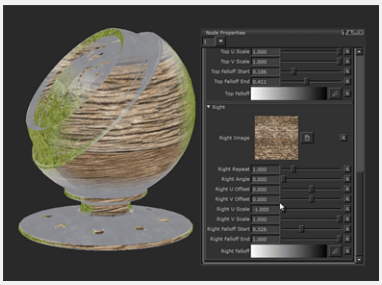
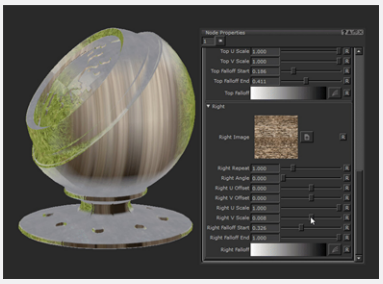
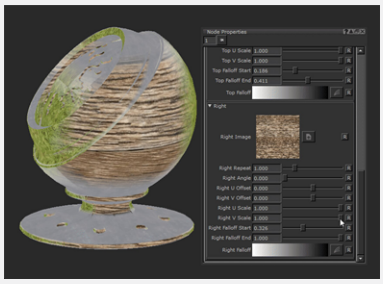
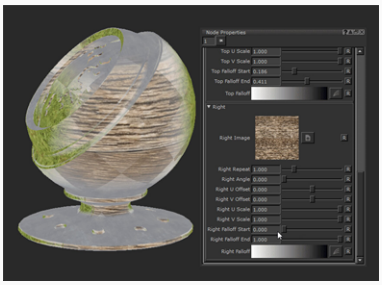
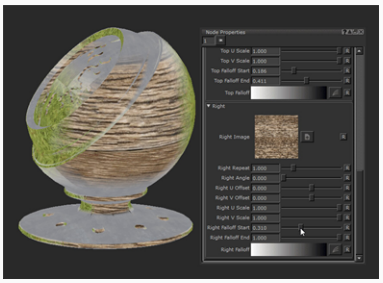
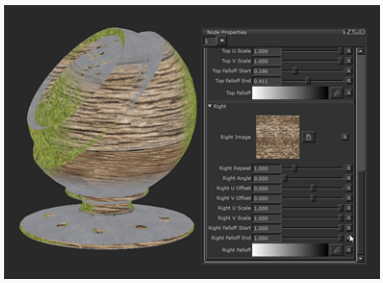
Top Falloff Start set to 1.

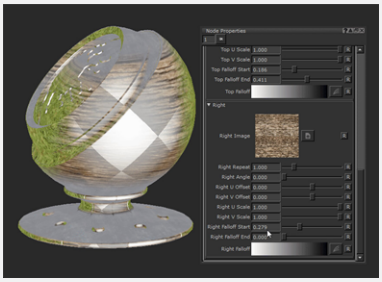
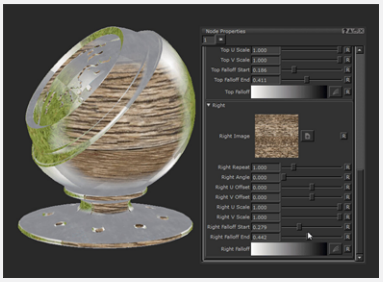
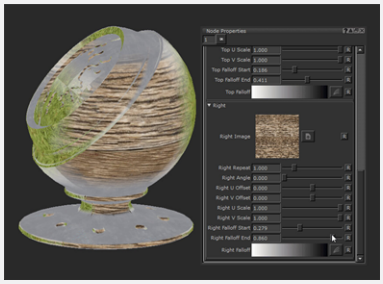
	Default is 0.
Top Falloff End *text field, slider*	Changes the fall off of the top projection and the threshold at which angles the projection will cover. Lowering this slider will make the projection cover less area on curved or glancing angles.  **Top Falloff End** set to 0.  **Top Falloff End** set to 0.4.  **Top Falloff** End set to 1. Default is 1.
Top Falloff *curve editor*	This curve editor gives control of the falloff of the front projection. By manipulating the curve you can make the falloff harsher. This may be useful when texturing hard surface assets to avoid soft and blended textures.

Right

Right Image *image*	Chooses the image to project across the right of your geometry.
Right Repeat *text field, slider*	Changes the tiling amount of your projected image. Upping this value makes the image scale down.

	 **Right Repeat** set to 0.  **Right Repeat** set to 4.  **Right Repeat** set to 18. Default is 0.
Right Angle *text field, slider*	Changes the rotation of the projected image. Default is 0.
Right U Offset *text field, slider*	Moves the tiled image horizontally along the U axis of the projection. Default is 0.
Right V Offset *text field, slider*	Moves the tiled image vertically along the V axis of the projection. Default is 0.
Right U Scale *text field, slider*	Scales the tiled image horizontally along the U axis of the projection. If this number is higher than the V Scale the image starts to elongate vertically.  **Right U Scale** set to -1.  **Right U Scale** set to 0.1.  **Right U Scale** set to 0.8.

	Default is 1.
Right V Scale *text field, slider*	Scales the tiled image vertically along the V axis of the projection. If this number is higher than the U Scale the image starts to elongate horizontally .    **Right V Scale** set to -1. **Right V Scale** set to 0. **Right V Scale** set to 1. Default is 1.
Right Falloff Start *text field, slider*	Changes the fall off of the right projection and the threshold at which angles the projection will cover. Raising this slider makes the projection cover more area on curved or glancing angles and may eventually overlap the other axes projections.    **Right Falloff Start** set to 0. **Right Falloff Start** set to 0.3. **Right Falloff Start** set to 1. Default is 0.
Right Falloff End *text field, slider*	Changes the fall off of the right projection and the threshold at which angles the projection covers. Lowering this slider makes the projection cover less area on curved or glancing angles.

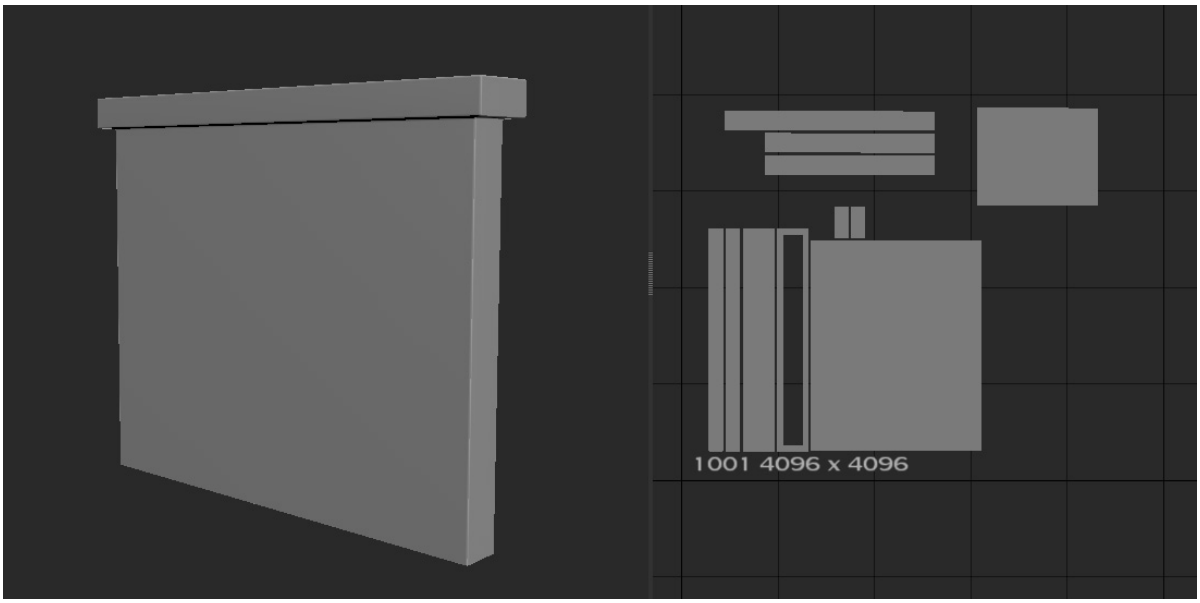
	 **Right Falloff End** set to 0.  **Right Falloff End** set to 0.4.  **Right Falloff End** set to 0.8. Default is 1.
Right Falloff *curve editor*	This curve editor gives control of the falloff of the front projection. By manipulating the curve you can make the falloff harsher. This may be useful when texturing hard surface assets to avoid soft and blended textures.

Triplanar Projection Node Workflow Example

Hard Surface Texturing with the Triplanar Projection vs a Tiled Node

You can use the Triplanar Projection node to help with hard surface texturing. Since it doesn't rely on UVs then you won't get any texture stretching due to poorly laid out UVs.

This basic rectangular wall object will be our geometry for this example. If we look at the UVs they are not ideal. They've had a quick automatic unwrap and they've been purposefully made worse by differing the scale of the shells and rotating some differently.



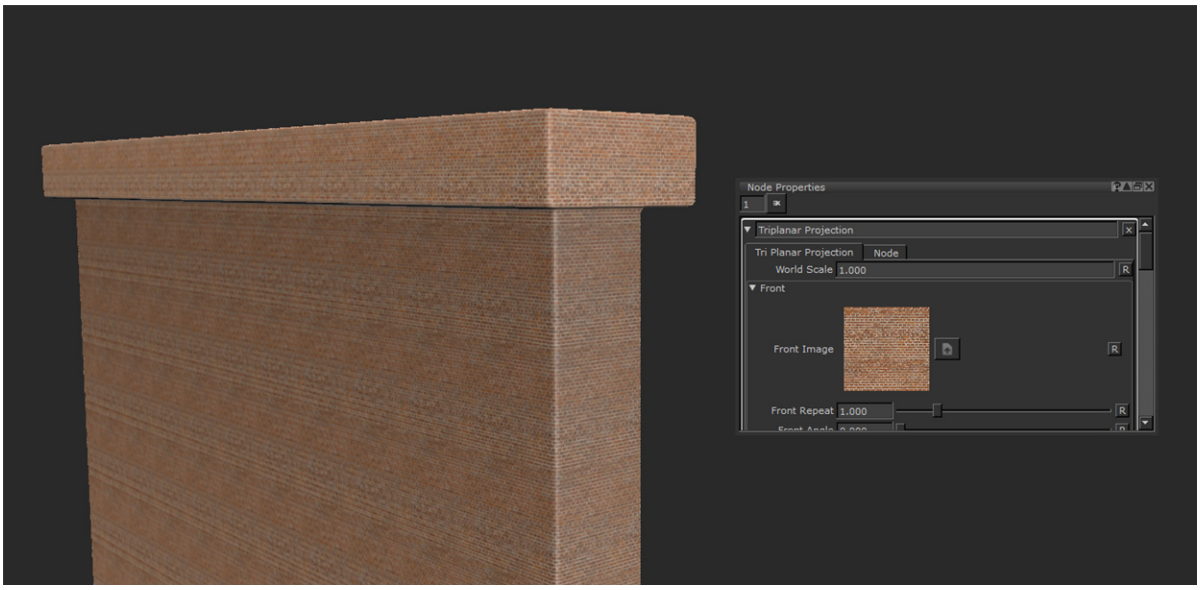
A basic wall model with bad UVs.

You may get less than ideal UVs when working with someone else. This would be an issue if you are working with the Tiled node, but the Triplanar Projection node has no issue with it since it works in 3D Space.

The Tiled node is not working well with the bad UV layout.

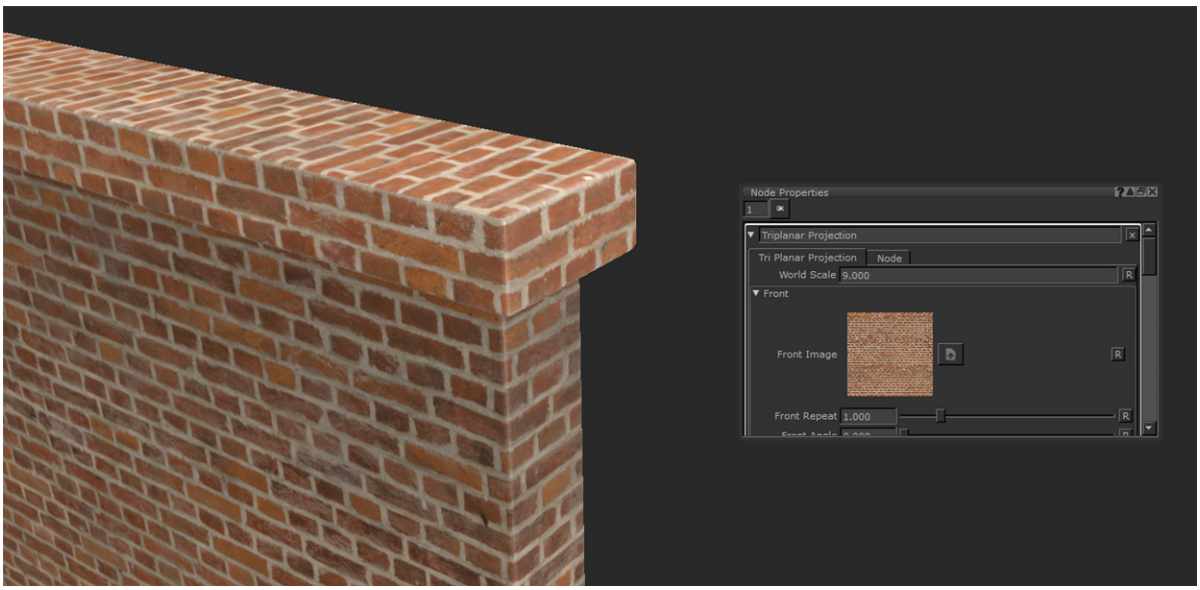
Here with a Tiled node using a brick texture we can see that having different scaled UVs and different rotation of the shells gives us bad results - we would need to use different Tiled nodes with different properties masked off in different places to get it to work.

Lets use a Triplanar Projection instead.



Triplanar Projection handles the tiling much better.

After connecting that same image from our **Image Manager** into the **Front**, **Top** and **Right** image of the node, already you can see the difference. While the scale may be off the tiles all look consistent and at the edges they line up correctly, even with our poor UVs.



The world scale helps scale all the axis up equally.

Finally by changing the **World Scale** of the node, all the bricks have been sized up. This saves you having to change the **Repeat** value or the **Scale Amount** on each axis. This looks great and a perfect base to add some wear and dirt onto for this wall asset. Getting something similar with the Tiled node would have been difficult.

Bricks and tiles can be very temperamental when texturing as you need straight UVs, the same UV scale, and often your UV shells to be inline for the corners to match up correctly. Unlike other hard surface or organic materials it is very obvious when they are not right due to their high frequency and straight grid formation. The Triplanar Projection node is a great help when your UVs are giving you trouble.

Tri-Planar Tiled node

Access: **Nodes** > **Projection** > **Tri-Planar Tiled Node**

The Tri-Planar Tiled node projects a single 2D image texture across a 3D model along the X, Y, and Z axes.

By sampling the image from top (XZ), front (XY), and right (ZY) orientations simultaneously, it applies uniform surface detailing while bypassing UV distortion and seam placement issues. Useful for organic textures that don't have a lot of directionality, such as grass or sand.

You can adjust transformations—such as scale, rotation, offset, and repeat values—either globally or individually per plane, and supply dedicated normal maps or noise inputs to further refine surface transitions.

Tri-Planar Tiled node inputs

- **Image** — The 2D image to project onto the 3D model.
- **Position** — Use this input to override the 3D coordinates of the projection (e.g. with a Coordinate Transform node node).
- **Normal** — Use this input to override the geometry's surface normals.
- **Blending Noise** — When **Blend Mode** of this node is set to **Blending Noise** Input, it uses the input to this port for blending with the RGB of the input adjusting the weights of the XYZ axis projections, respectively.

> **Note:** A grayscale input in **Blending Noise** will adjust the weights of each projection equally, and therefore will not appear to have any effect.

Tri-Planar Tiled node outputs

• **Output** — The 2D image projected on the 3D model.

Tri-Planar Tiled node properties

Image	The image to project onto the surface. If an Image Input node is connected to the Image input of this node, it will be used instead of this parameter.
Normal Map	If the images used are tangent space normal maps, enable this to account for any rotation.
Green Channel	Whether the normal map is treated as having a Y+ or Y- green channel.
Plane Settings	Select which of the plane settings to use for all planes.
Blending	
Blending Mode	The mode used to adjust the blending between projections, to avoid obvious blending seams. • **None** is a linear blend. • **Smart** tries to keep prominent details across the blend. • **Noise** applies a Perlin noise to the blend. • **Blending Noise Input** uses the input to the Blending Noise port, with the RGB of the input adjusting the weights of the XYZ axis projections, respectively.
Blend Strength	How strong the blending adjustment is. A value of zero becomes a standard linear blend, similar to the None option of the Blend Mode input.
Noise Scale	The scale of the noise used to blend the different projections. Only applies if Blend Mode is Noise.
Noise Contrast	Adjust the contrast of the noise used to blend the different projections. Only applies if **Blend Mode** is **Noise**.
Right (ZY plane)	
Projection	Choose which directions along the X axis (projection on the ZY plane) should be visible.

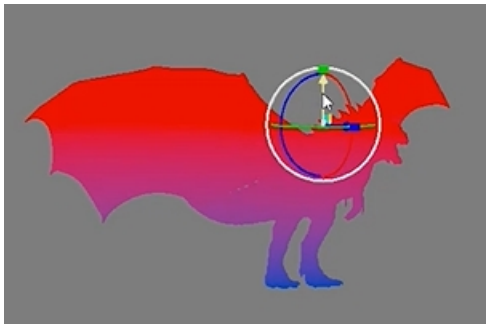
Repeat	How many tiles of the image should appear. A value of zero will tile infinitely.
Scale	Adjust the scale of the projection, larger values tile the image more.
Rotation	Adjust the rotation of the projection around its center.
Offset	Offset the projection across the projection plane.
Falloff Range	Control where the falloff transparency starts and ends relative to the angle between the projection axis and the surface. 0.0 is where the surface faces the projection axis, 1.0 is where the surface is perpendicular to the projection axis.
Falloff Curve	Allows fine-tune control of the falloff of the projection alpha. The curve is applied to the falloff range.
Top (XZ plane)	
Projection	Choose which directions along the Y axis (projection on the XZ plane) should be visible.
Repeat	How many tiles of the image should appear. A value of zero will tile infinitely.
Scale	Adjust the scale of the projection, larger values tile the image more.
Rotation	Adjust the rotation of the projection around its center.
Offset	Offset the projection across the projection plane.
Falloff Range	Control where the falloff transparency starts and ends relative to the angle between the projection axis and the surface. 0.0 is where the surface faces the projection axis, 1.0 is where the surface is perpendicular to the projection axis.
Falloff Curve	Allows fine-tune control of the falloff of the projection alpha. The curve is applied to the falloff range.
Front (XY plane)	
Projection	Choose which directions along the Z axis (projection on the XY plane) should be visible.
Repeat	How many tiles of the image should appear. A value of zero will tile infinitely.
Scale	Adjust the scale of the projection, larger values tile the image more.
Rotation	Adjust the rotation of the projection around its center.
Offset	Offset the projection across the projection plane.

Falloff Range	Control where the falloff transparency starts and ends relative to the angle between the projection axis and the surface. 0.0 is where the surface faces the projection axis, 1.0 is where the surface is perpendicular to the projection axis.
Falloff Curve	Allows fine-tune control of the falloff of the projection alpha. The curve is applied to the falloff range.

Two Point Gradient Node

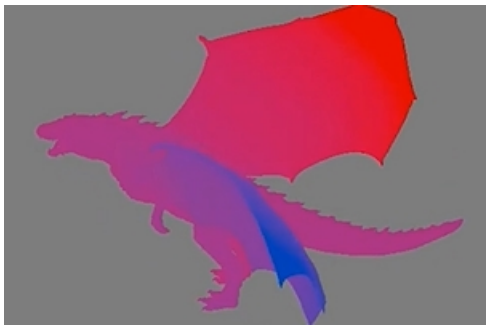
Access: **Nodes** > **Projection** > **Two Point Gradient**

The Two Point Gradient node creates a gradient based on two locator positions relative to geometry. You can control the gradient between the start and end colors using the **Mode** controls and assign multiple projectors/locators to the node to create as many gradients as you require from any angle.



Locator A with a red **Color A**.

Locator B with a blue **Color B**.



The gradient color changes dependent on distance between locator A and locator B.

Two Point Gradient Node Inputs

Color A	Sets the start color of the gradient, which is applied to geometry closest to locator A.
Color B	Sets the start color of the gradient, which is applied to geometry closest to locator B.
Position	An optional input to set the start coordinates of the gradient.

Two Point Gradient Node Outputs

Output	A two point gradient calculated from two locator positions relative to geometry.

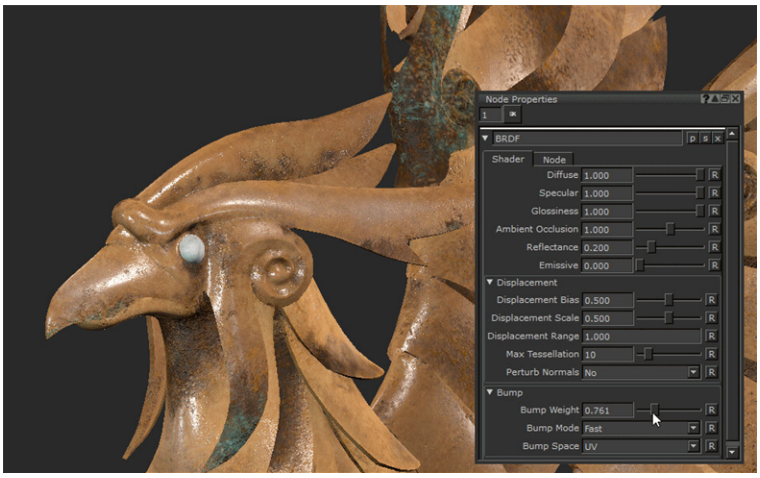
Two Point Gradient Node Properties

Mode *dropdown*	Sets how the gradient between the two locators/projectors is calculated, either **Linear** or **Radial**.
Color A *swatch*	Sets the default start color of the gradient if nothing is connected to the **Color A** input.
Color B *swatch*	Sets the default end color of the gradient if nothing is connected to the **Color B** input.
Locator A *dropdown*	Selects the object to use as locator A. You can use one of the standard options, objects, cameras, and lights, or create a custom locator by pressing the **+** button.
+	Click to add a locator to the scene to control the gradient applied to your geometry.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.

T	Click to show and hide the **Locator A** transform controls.
Locator B *dropdown*	Selects the object to use as locator B. You can use one of the standard options, objects, cameras, and lights, or create a custom locator by pressing the **+** button.
+	Click to add a locator to the scene to control the gradient applied to your geometry.
P	Click to switch to the **Transform Selected Objects** tool so you can move the selected object around the scene.
T	Click to show and hide the **Locator B** transform controls.

BRDF (Shader Network) Node

Access: **Nodes** > **Shader Network** > **BRDF**



The BRDF node and its inputs.

The BRDF node is realistic shading model with recreates the way light reflects at different angles, providing Fresnel effects.

A shader node is a way to view multiple texture maps at once. You can view what they affect at render time, rather than as a single color map on your mesh. They are a great way to texture and Lookdev on the fly and make sure you textures have the correct values.

The Shader Network version of the node includes a few extra properties that have been set up inside the node's group. These are the **Bump**, **Vector** and **Displacement** inputs which can be added to the shader.

Tip: Press **Ctrl**+double click on the node to jump into the group and see the construction of the aspects of the node.

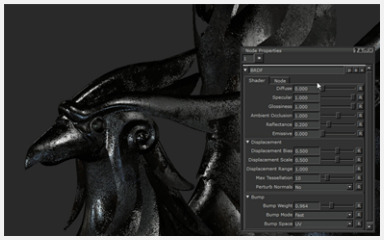
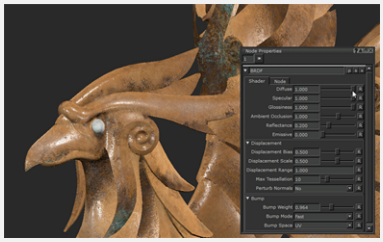
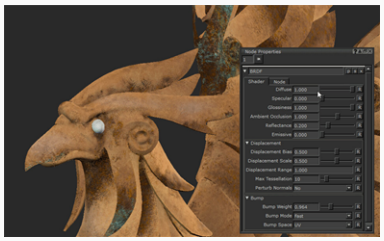
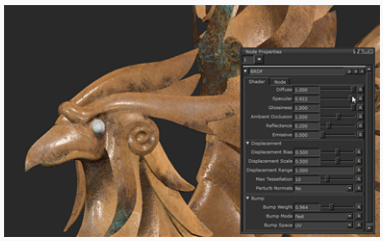
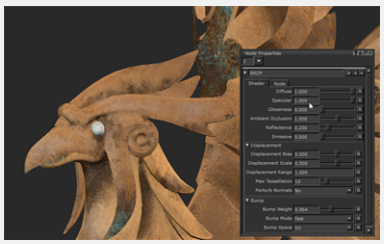
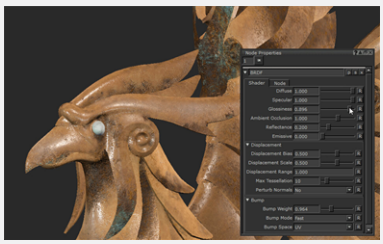
BRDF (Shader Network) Node Inputs

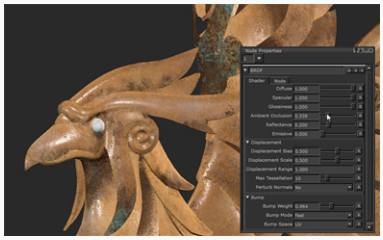
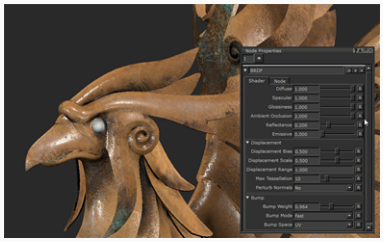
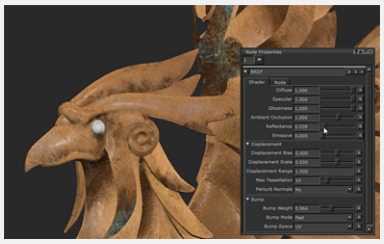
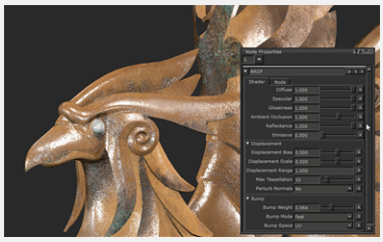
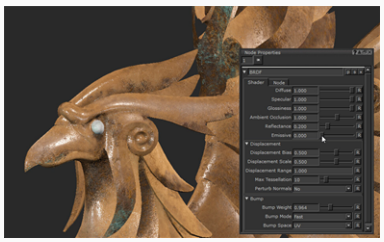
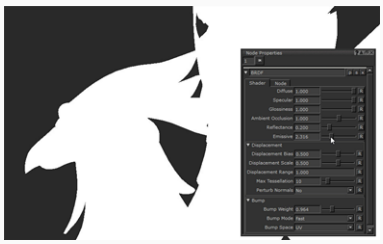
Input	Description
Diffuse Color	Assigns the input texture to the Diffuse Color of the shader.
Specular Color	Assigns the input texture to the Specular Color of the shader.
Glossiness	Assigns the input texture to the Glossiness value of the shader.
Reflectance	Assigns the inputted texture to the Reflectance value of the shader.
Ambient Occlusion	Assigns the input texture to the Ambient Occlusion value of the shader.
Emissive Color	Assigns the input texture to the Emissive Color of the shader.
Normal	Assigns the input texture to the Normal Map of the shader.
Bump	Assigns the input texture to the Bump Map of the shader.
Vector	Assigns the input flow map to the Vector Map of the shader. See Vector Brush for more information.
Displacement	Assigns the input texture to the Displacement value of the shader.

BRDF (Shader Network) Node Properties

The images in the examples here are using a fully texture asset with channels plugged into the associated inputs of the shader and an environment light for reflections..

Diffuse *Text field, slider*	A multiplier of your connected Diffuse Color input. Lowering this to 0 will give you a black diffuse.

	 **Diffuse** amount set to 0.  **Diffuse** amount set to 1. Default is 1.
Specular *Text field, slider*	A multiplier of your connected Specular Color input. Lowering this to 0 will give you a black specular value.  **Specular** amount set to 0.  **Specular** amount set to 0.9. Default is 1.
Glossiness *Text field, slider*	Changes the tightness of the specular highlights and reflections. A value of 1 will give you very glossy highlights, and 0 will give a much rougher broad specular.  **Glossiness** amount of 0.  **Glossiness** amount of 0.8. Default is 0.2.
Ambient Occlusion *Text field, slider*	A multiplier which raises and lowers the effect of Ambient Occlusion on your model. Requires Ambient Occlusion to be calculated from the **Object** menu to have any affect - (**Object** > **Ambient Occlusion**)

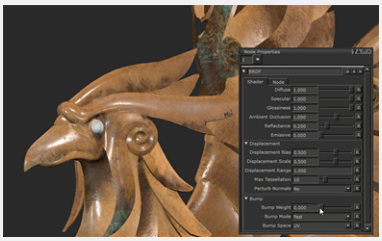
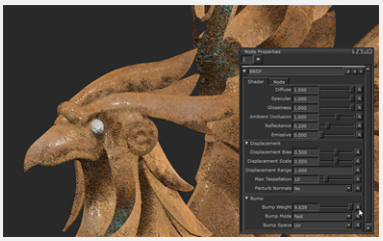
	 **Ambient Occlusion** amount of 0.3.  **Ambient Occlusion** amount of 2. Default is 1.
Reflectance *Text field, slider*	Raises and lowers the reflected light off the surface. This property along with Specular Colour and Glossiness will affect how specular highlights and reflections are displaced on your object.  **Reflectance** amount of 0.  **Reflectance** amount of 1. Default is 0.2.
Emissive *Text field, slider*	A multiplier to the input Emissive Color. Requires an input to have affect.  **Emissive** amount of 0.  **Emissive** amount of 2. Default is 1.

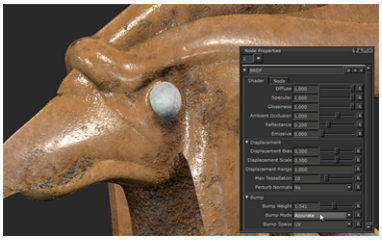
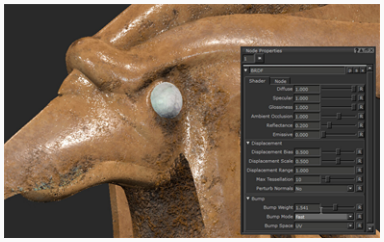
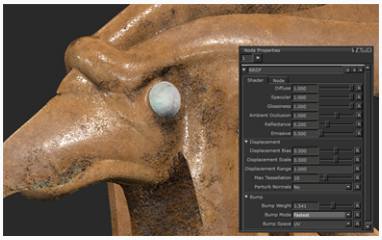
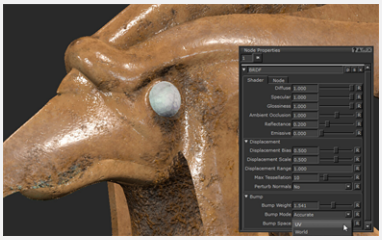
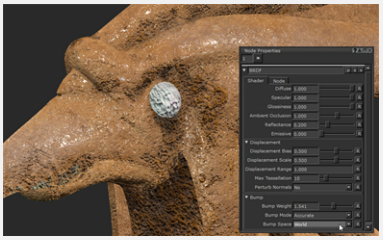
Displacement

Note: Displacement in Mari can cause a strain on your machine, especially in heavy scenes.

Displacement Bias *Text field, slider*	Changes the midpoint of the displacement, and what value will not displace. 0.5 means mid gray will have no change, 1.0 will push out, 0.0 will push in. Default is 0.5.
Displacement Scale *Text field, slider*	Changes the intensity of the displacement of the geometry. Default is 0.5.
Displacement Range *Text field*	Sets the range for the search distance. Default is 1.000.
Max Tessellation *Text field, slider*	Raising this will increase the fidelity of the displacement being displayed. This will increase the look and accuracy of the displacement but decrease performance. Default is 10.
Perturb Normals *dropdown*	If **Yes** is selected the normals will remain as they were when not-displaced after the geometry is moved. Default is **No**.

Bump

Bump Weight *Text field, slider*	Changes the strength of the Bump being applied to the geo's surface.  **Bump Weight** of 0.  **Bump Weight** of 9. Default is 0.1.
Bump Mode	Defines the calculation method of the bump being displayed. You can change it between Fast, Accurate or Fastest.

dropdown	Accurate will give you the smoothest calculation method but will take longer to calculate which can slow down your scene and shader. Fastest is the quickest to calculate but you can get odd results and anomalies on a per pixel basis giving you an inaccurate result.  **Bump Mode** set to **Accurate**.  **Bump Mode** set to **Fast**.  **Bump Mode** set to **Fastest**. Default is Fast.
Bump Space *dropdown*	Changes the calculation of the normals for the Bump map between UV and World coordinate space. • **UV** - Normals are calculated relative to the UV coordinates space. The UV Bump Space mode is the more traditional approach but can lead to noticeable seams and color variations. • **World** - Normals are calculated relative to the World coordinates space. Using the World Bump Space mode gives a more seamless result.  **Bump Space** set to **UV**.  **Bump Space** set to **World**. Default is UV.

BRDF (Shader Network) Node Workflow

Shaders are an important aspect of material creation in Mari. Regardless of which shading model you use, being able to view multiple channels at once is the best way to view your textures as you have an accurate representation of how they will behave at render time. This is important in any situation, whether it's working

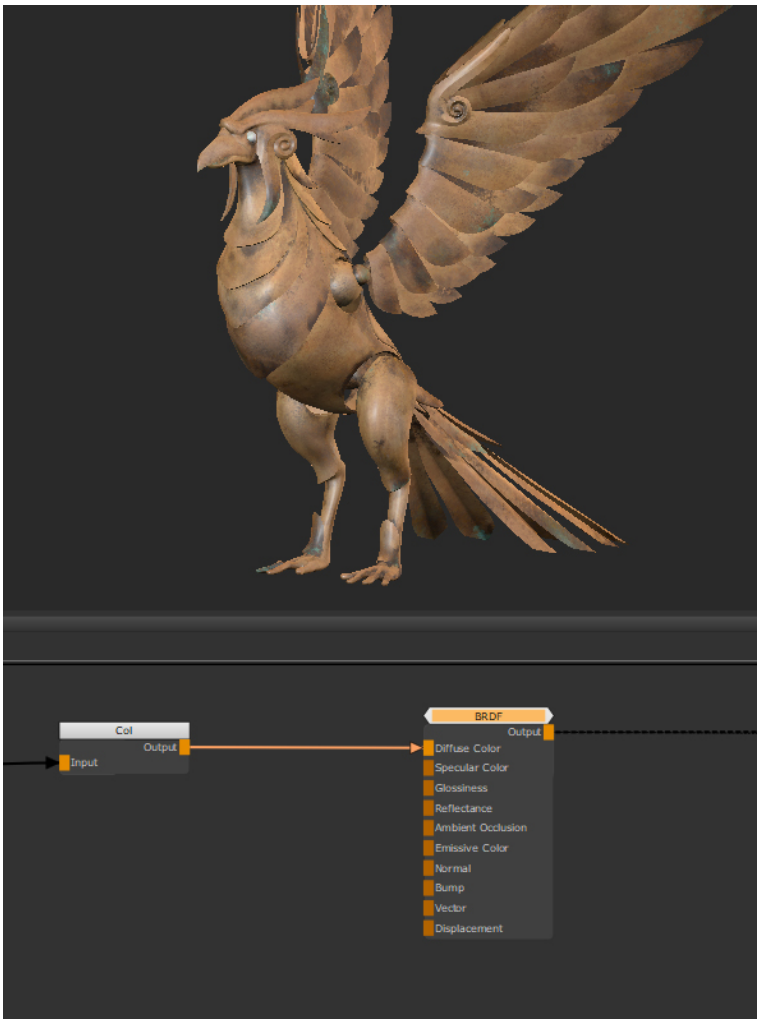
on your own work at home or in a big studio where your textures will go to the lookdev department to be rendered.

Having the knowledge of seeing what you plug into your shaders is important and will save you time tweaking and editing them in the renderer.

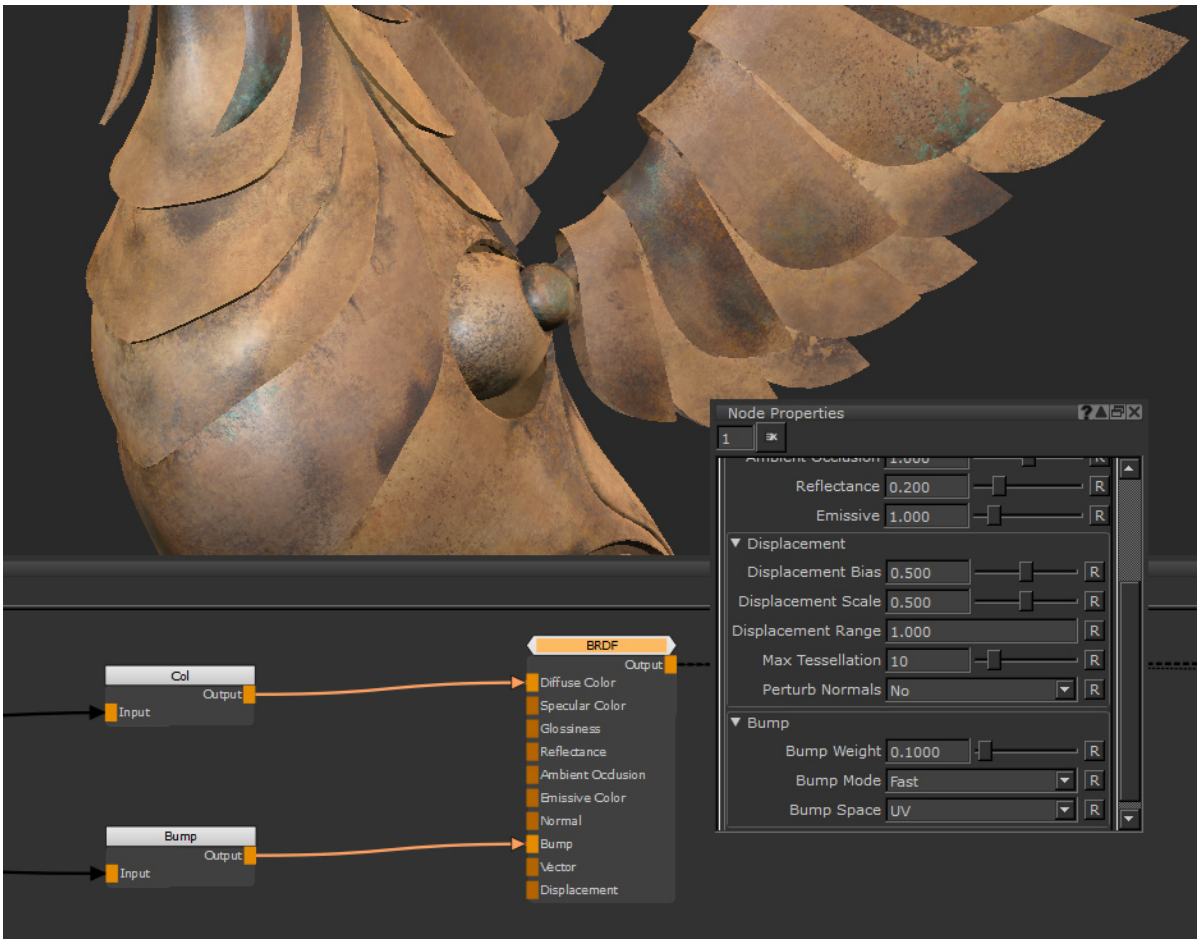
Here's the Color map for a mechanical bird asset.

Viewing a texture map as a flat color on your geometry is important, but also inputting it into a shader to view the shaded versions of maps like Bump, Roughness, and Normal helps you see where your textures are or aren't working.

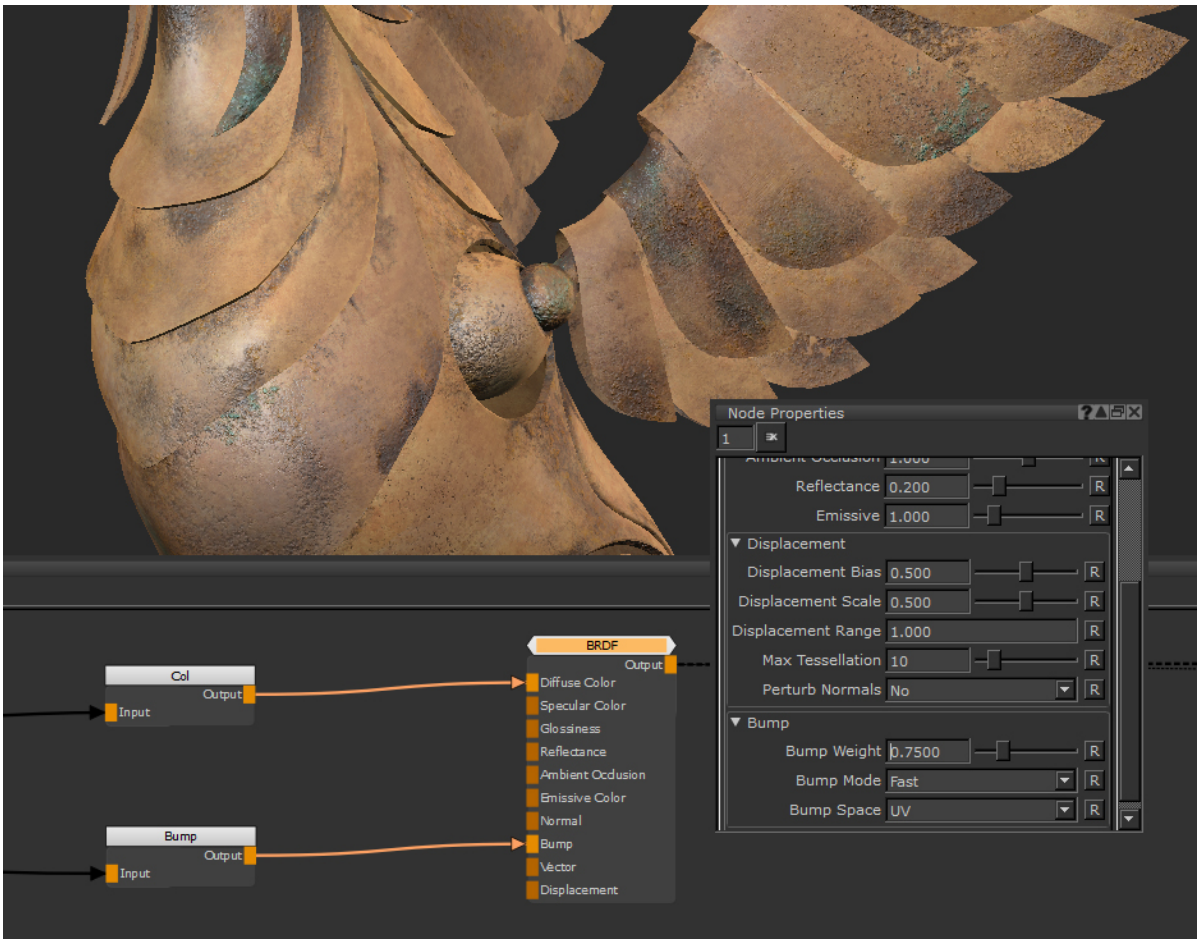
To set up a shader, connect a Base Color. Drag and drop a connection from your color channel to the **Diffuse Color** input. By doing this you are telling the shader what the value of the base color is.



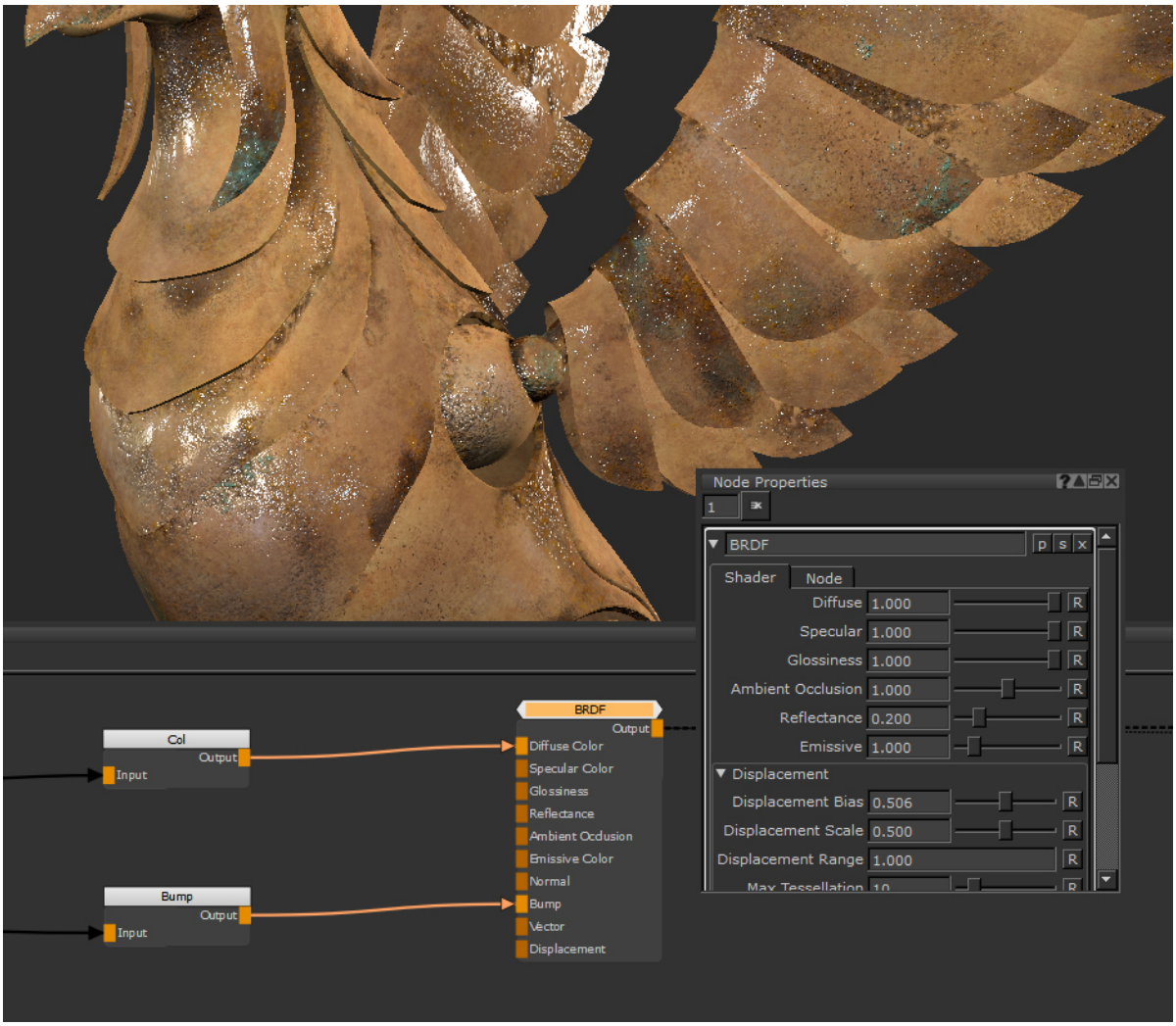
We can now start to visualize the object with specular and shadows rather than just flat color. From there, you can start plugging in other channels you have textured into your shader, such as a bump map into the **Bump** input.



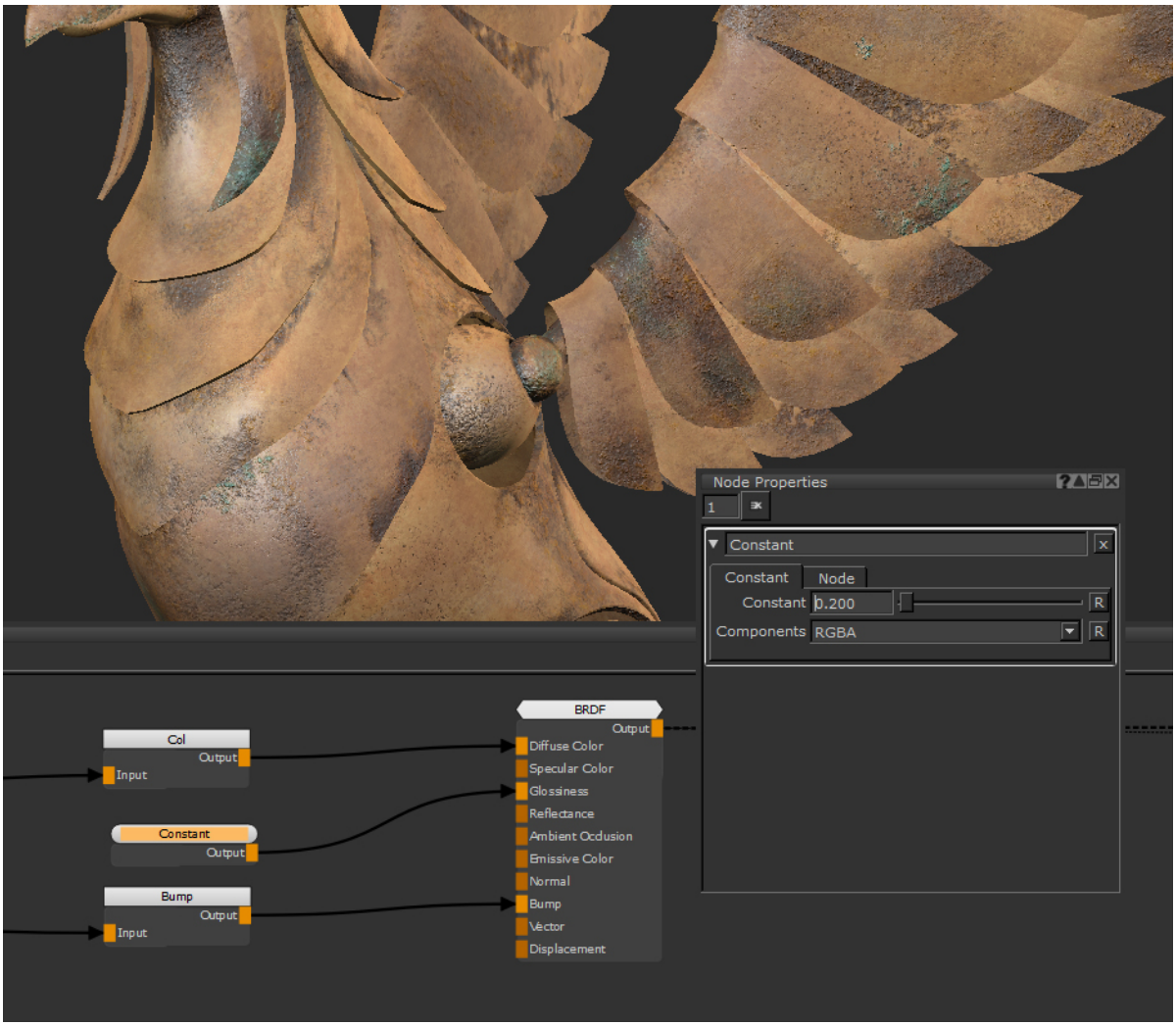
It's not made much of a difference on the **Bump Weight** base setting of 0.1, so we can increase that to 0.75 to see the result a bit more clearly.



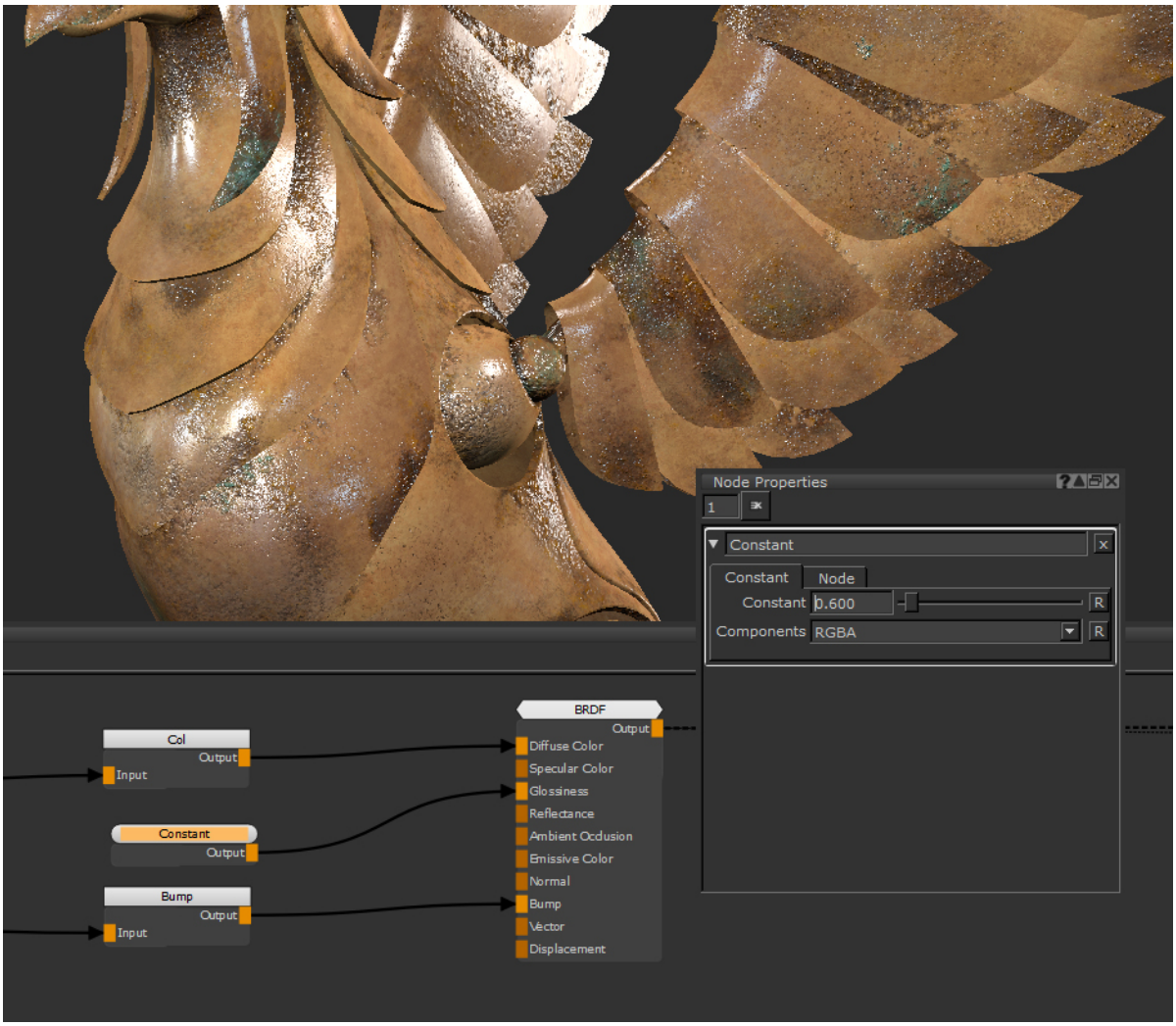
But what if you don't have a map to input? We haven't yet painted a gloss map for the **Glossiness** input of the shader, but by using the slider on the shader you can find a rough value that works well to get you the results you like. First though, make sure to turn glossiness up to 1.0. That way the value we plug in will not be multiplied so we know we are seeing it correctly.



We can use a Constant node plugged into **Glossiness**. This is just a value between 0 and 1 that we can use to find a value to start painting our roughness map with.



This value of 0.2 works well for a broader glossiness value on the rusty areas.



And this tighter specular Glossiness value of 0.6 looks good on the smoother areas for a less damaged metal effect.

We now have two values of 0.2 and 0.6 to use in our Glossiness map. This is now a value to aim for, you can make sure whatever you end up inputting is roughly around that level. This is a great way that a shader can be used to help you texture and lookdev at the same time.

Coordinate Transform node

Access: **Nodes** > **Transformation** > **Coordinate Transform**

This node drives any node with a Position input, by scaling, rotating and translating coordinates in one step. It allows external drivers to override internal controls for procedural, non-destructive coordinate manipulation. You can also connect other nodes to drive the transformations.

There are also 3 separate nodes to achieve the desired effect separately — see more at Translation Transform, Scale Transform and Rotate Transform.

Coordinate Transform node inputs

- **Coordinates** — The coordinates of the input node (scale, rotation, translation).
- **Scale** — The scale coordinates of the input node (used if you only want to input scale information).
- **Rotation** — The rotation coordinates of the input node (used if you only want to input rotation information).
- **Translation** — The translation coordinates of the input node (used if you only want to input translation information).

Coordinate Transform node outputs

- **Output** — Outputs the transformed Scale, Rotation and Translation coordinates.
- **Scale** — Outputs the transformed Scale coordinates.
- **Rotation** — Outputs the transformed Rotation coordinates.
- **Translation** — Outputs the transformed translation coordinates.

Coordinate Transform node properties

World Scale	*text field, slider*	Scales the pattern evenly in every direction, on top of Scale X/Y/Z. Values outside the slider range can still be typed in directly.
Scale X	*text field, slider*	How much to scale along each axis. 1.0 is no change. Works together with World Scale to give the overall size. Only used when nothing is connected to the Scale input.
Scale Y	*text field, slider*	How much to scale along each axis. 1.0 is no change. Works together with World Scale to give the overall size. Only used when nothing is connected to the Scale input.
Scale Z	*text field, slider*	How much to scale along each axis. 1.0 is no change. Works together with World Scale to give the overall size. Only used when nothing is connected to the Scale input.
Rotate X	*text*	How far to rotate around the X axis, in degrees. Only used when nothing is

	field, slider	connected to the Rotation input.
Rotate Y	*text field, slider*	How far to rotate around the Y axis, in degrees. Only used when nothing is connected to the Rotation input.
Rotate Z	*text field, slider*	How far to rotate around the Z axis, in degrees. Only used when nothing is connected to the Rotation input.
Translate X	*text field, slider*	How far to translate along the X axis. Combined with the Offset values to give the final position. Only used when nothing is connected to the Translation input.
Translate Y	*text field, slider*	How far to translate along the Y axis. Combined with the Offset values to give the final position. Only used when nothing is connected to the Translation input.
Translate Z	*text field, slider*	How far to translate along the Z axis. Combined with the Offset values to give the final position. Only used when nothing is connected to the Translation input.
Offset X	*text field, slider*	A small, constant adjustment added on top of Translate X. Unlike Translate X, this stays active even when Translation is driven by a connected node, so it's useful for a fixed nudge. Strengthened or weakened by the Offset Multiplier.
Offset Y	*text field, slider*	A small, constant adjustment added on top of Translate Y. Unlike Translate Y, this stays active even when Translation is driven by a connected node, so it's useful for a fixed nudge. Strengthened or weakened by the Offset Multiplier.
Offset Z	*text field, slider*	A small, constant adjustment added on top of Translate Z. Unlike Translate Z, this stays active even when Translation is driven by a connected node, so it's useful for a fixed nudge. Strengthened or weakened by the Offset Multiplier.
Offset Multiplier	*text field, slider*	Strengthens or weakens the offset Values.

Locator Transform node

Access: **Nodes** > **Transformation** > **Locator Transform**

The Locator Transform node translates coordinates so they follow the position, rotation, and scale of a chosen locator in the scene.

Also outputs the locator's own Scale, Rotation, and Translation values so they can be reused elsewhere, and lets those values be overridden by connecting other nodes.

Locator Transform node inputs

- **Coordinates** — The coordinates of the input locator (scale, rotation, translation).
- **Scale** — The scale coordinates of the input locator (used if you only want to input scale information).
- **Rotation** — The rotation coordinates of the input locator (used if you only want to input rotation information).
- **Translation** — The translation coordinates of the input locator (used if you only want to input translation information).

Locator Transform node outputs

- **Output** — Outputs the transformed Scale, Rotation and Translation coordinates.
- **Scale** — Outputs the transformed Scale coordinates.
- **Rotation** — Outputs the transformed Rotation coordinates.
- **Translation** — Outputs the transformed translation coordinates.

Locator Transform node properties

Locator	*dropdown*	Defines the locator to be transformed.
Translation	*text field*	How far to translate along each axis (X, Y, Z) .
Locator Rotation	*text field, slider*	How far to rotate around each axis, in degrees.
Scale	*text field*	How much to scale along each axis. 1.0 is no change.

Rotation Transform node

Access: **Nodes** > **Transformation** > **Rotation Transform**

This node drives any node with a Position input, by rotating the coordinates. It allows external drivers to override internal controls for procedural, non-destructive coordinate manipulation. You can also connect other nodes to drive the transformations.

Rotation Transform node inputs

- **Coordinates** — The coordinates of the input node (scale, rotation, translation).
- **Rotation** — The rotation coordinates of the input node (used if you only want to input rotation information).

Rotation Transform node outputs

- **Output** — Outputs the transformed Rotation coordinates.

Rotation Transform node properties

Rotate X	*text field, slider*	How far to rotate around the X axis, in degrees. Only used when nothing is connected to the Rotation input.
Rotate Y	*text field, slider*	How far to rotate around the Y axis, in degrees. Only used when nothing is connected to the Rotation input.
Rotate Z	*text field, slider*	How far to rotate around the Z axis, in degrees. Only used when nothing is connected to the Rotation input.

Scale Transform node

Access: **Nodes** > **Transformation** > **Scale Transform**

This node drives any node with a Position input, by scaling the coordinates. It allows external drivers to override internal controls for procedural, non-destructive coordinate manipulation. You can also connect other nodes to drive the transformations.

Scale Transform node inputs

• **Coordinates** — The coordinates of the input node (scale, rotation, translation).

• **Scale** — The scale coordinates of the input node (used if you only want to input scale information).

Scale Transform node outputs

• **Output** — Outputs the transformed Scale coordinates.

Scale Transform node properties

Scale X	*text field, slider*	How much to scale along each axis. 1.0 is no change. Only used when nothing is connected to the Scale input.
Scale Y	*text field, slider*	How much to scale along each axis. 1.0 is no change. Only used when nothing is connected to the Scale input.
Scale Z	*text field, slider*	How much to scale along each axis. 1.0 is no change. Only used when nothing is connected to the Scale input.

Translation Transform node

Access: **Nodes** > **Transformation** > **Translation Transform**

This node drives any node with a Position input, by translating the coordinates. It allows external drivers to override internal controls for procedural, non-destructive coordinate manipulation. You can also connect other nodes to drive the transformations.

Translation Transform node inputs

• **Coordinates** — The coordinates of the input node (scale, rotation, translation).

- **Translation** — The translation coordinates of the input node (used if you only want to input translation information).

Translation Transform node outputs

- **Output** — Outputs the transformed Translation coordinates.

Translation Transform node properties

Translate X	*text field, slider*	How far to translate along the X axis. Only used when nothing is connected to the Translation input.
Translate Y	*text field, slider*	How far to translate along the Y axis. Only used when nothing is connected to the Translation input.
Translate Z	*text field, slider*	How far to translate along the Z axis. Only used when nothing is connected to the Translation input.

UV Radial Transform node

Access: **Nodes** > **Transformation** > **UV Radial Transform**

UV Radial Transform converts UV coordinates into a radial (distance and angle) pattern, tile by tile. This node is useful for dial-like, circular, or spoke patterns that repeat consistently across UDIMs.

The node's **Interpolate Quadrants** property (which is on by default) keeps the pattern symmetrical; turn it off for a plain full-circle sweep.

Optional Center/Rotate/Scale inputs let another node drive these values instead of the sliders.

UV Radial Transform node inputs

- **Coordinates** — The coordinates of the input UV radial (scale, rotation, translation).
- **Center** — The center coordinates of the input UV radial (used if you only want to input center information).

- **Scale** — The scale coordinates of the input UV radial (used if you only want to input scale information).
- **Rotate** — The rotation coordinates of the input UV radial (used if you only want to input rotation information).

UV Radial Transform node outputs

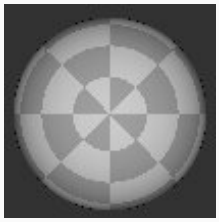
- **Output** — Outputs the transformed Scale, Rotation and Translation coordinates.

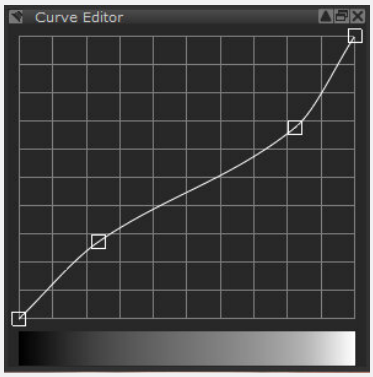
Locator Transform node properties

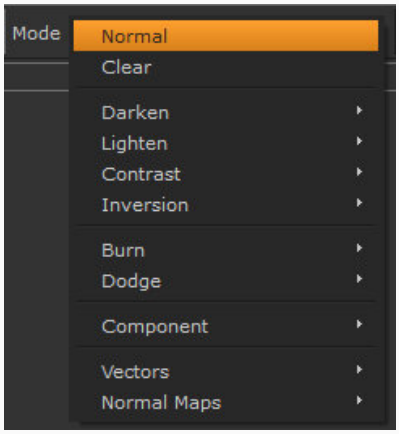
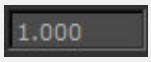
Center		
Center X	*slider*	Shifts the center of the radial pattern left/right within the tile. Only used when nothing is connected to the Center input.
Center y	*slider*	Shifts the center of the radial pattern up/down within the tile. Only used when nothing is connected to the Center input.
Scale		
Repeat R	*slider*	How many times the pattern repeats from the center out to the tile edge. Only used when nothing is connected to the Scale input.
Repeat angle	*slider*	How many times the pattern repeats going around a full circle. Only used when nothing is connected to the Scale input.
Rotate		
Rotate angle	*slider*	Spins the whole pattern around its center, in degrees. Only used when nothing is connected to the Rotate input.
Mode		
Interpolate Quadrants	*checkbox*	When enabled, mirrors the pattern across all four quadrants for a seamless, symmetrical look. Turn off for a plain full-circle sweep instead.

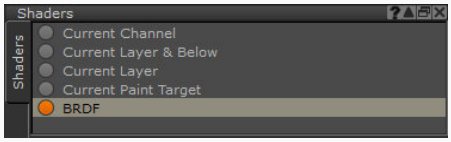
Types of Controls

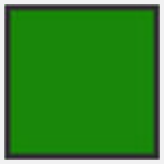
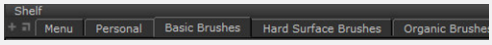
The types of controls that can be found in Mari are listed in the table below. Each control is accompanied by an example image of how it appears in Mari, as well as a description of what it does and how to use it.

Controls	What it looks like	What it does	How to use it	Notes
Add attribute		Lets you add an attribute to a property.	Click the icon and fill in the **Add User Attribute** dialog box.	For more information, see the *Extending Mari* chapter in the *Mari User Guide*.
Buttons		Lets you select a function.	Click it.	May show an icon or a word.
Checkbox	unselected, selected	Lets you select an option.	Click to select.	An **x** displays when selected.
Control sphere		Lets you move a light around the scene.	Click and drag to pull the light around.	
Curve editor	The Curve Editor button displays the Curve Editor in a pop-up window.	Maps between input and output values (45° = the same). Bar along the bottom displays the	Add points by clicking on the curve or bar. Click and drag to move a point on the graph. **Shift**+click and drag to move multiple points	You can move, edit, and delete multiple points at once.

Controls	What it looks like	What it does	How to use it	Notes
		effect at each point.	on the graph. To select and deselect multiple points: • **Shift**+click the points. • Click outside the curve. To edit points: • Double-click the selected points and change their values. • Right-click the points and select **Edit Points**. You can also edit a single point using the same methods. To delete a point: • **Ctrl** /**Cmd**+click it. • Right-click it and select **Delete**	

Controls	What it looks like	What it does	How to use it	Notes
			Point. To delete multiple points: • Right-click the selected points and select **Delete Points**.	
Dropdown list		Lets you select from a list of valid values.	Click the down arrow on the right, then click to select an option.	
text field		Lets you type an entry.	Click inside the box and type a valid entry.	Often combined with a slider and reset button. When an text field has been combined with a slider to change values, click in the text field and press the middle-

Controls	What it looks like	What it does	How to use it	Notes
				mouse button+drag up or down to increase or decrease the value.
Horizontal slider		Slides along a range of valid values.	Click and drag left (more) or right (less).	Often combined with an text field (that shows the value) and reset button.
Incrementers		Lets you increase or decrease a value.	Click the up or down arrows.	If you press **Ctrl** (on Windows and Linux) or **fn** (Mac) while pressing the up or down arrows, the number is increased or decreased by whole numbers rather than individual decimal points.
List		Lets you select from a list of valid values.	Click to select an option.	
On/Off	on, off	Shows	Look at it.	Often

Controls	What it looks like	What it does	How to use it	Notes
Indicator		whether an option is on or off		combined with a switch.
Option buttons		Lets you select mutually exclusive options.	Click to select which option you want.	Circle is filled when selected. Sometimes called “radio buttons”.
Remove attribute		Lets you remove an attribute from a property.	Click the icon, then click **Remove** to confirm.	
Reset button		Reset to the default.	Click the icon.	Often combined with an text field and slider
Swatch		Displays the currently selected color, and lets you select a different color.	Click to select a color from the Colors Palette	
Switch		Turns an option on or off.	Click and drag right (on) or left (off).	Often combined with an On/Off indicator.
Tabs		Displays several groups of the same kind of information,	Click: • a tab to view it • to view the previous or next tab	

Controls	What it looks like	What it does	How to use it	Notes
		such as different categories of brush tips in the Shelf Palette.	• to add a tab • to delete a tab • to open the current tab in a separate window	
Thumbnail		Displays a thumbnail preview of a selected feature or function.	Look at it.	

Changing Numerical Values

You can key values directly into a field, press the arrow keys to increment and decrement values, use the middle-mouse button, or click and drag on the value.

To key in field values:

1. Double-click in the field to select the whole value.
2. Type the value you want to replace the selection.

You can increment or decrement values by hundreds, tens, tenths, hundredths, and so on. The magnitude of change depends on the initial position of your cursor. For example if you wanted to increment the initial value of 20.51 by ones, you would insert your cursor before the 0.

To increment or decrement a field value:

1. Click to insert the cursor just prior to the digit you want to increment or decrement.
2. Press the up arrow to increment by one unit, or the down arrow to decrement by one unit.

Tip: You can also increment and decrement values using the mouse wheel (if available) or by clicking and dragging on the value. The latter method is particularly useful for tablet users.

Palettes

Learn about Mari's palettes and their controls. When controls also bring up additional dialogs, these are referenced for more information. You can access palettes through the menu bar by navigating to **View** > **Palettes** or by using the Palettes Toolbar.You can also right-click outside of the canvas and palettes to select palettes from the dropdown menu.

Use the left menu to see the details for Mari's palettes.

Channels Palette

The **Channels** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Channels** palette displays:

- a list of channels in the project,
- buttons to add, convert, or remove channels,
- options for creating quick channels, and
- channel attribute information.

Note: Channels provide a foundation that layers and layered shaders are built on. These channels can be diffuse, bump, or specular channels, to name only a few. Channels are assigned to lighting shaders, and sliders in that palette give you fine control over aspects of each channel.

Channels Palette Fields

Control	Type	What it does	Dialog	Notes
Quick Channel				
■	swatch	Creates a channel filled in black. The created channel is named **Quick Channel**. If there are multiple Quick Channels, then the name is followed with a number.		
□	swatch	Creates a channel filled in white. The created channel is named **Quick Channel**. If there are multiple Quick Channels, then the name is followed with a number.		
■	swatch	Creates a channel filled in mid-gray. The created channel is named **Quick Channel**. If there are multiple Quick Channels, then the name is followed with a number.		
▩	swatch	Creates a channel filled in transparent. The created channel is named **Quick Channel**. If there are multiple Quick Channels, then the name is followed with a number.		
<channel size>	dropdown	Sets the channel size used when creating a Quick Channel, from the options: • **256x256** • **512x512** • **1024x1024** • **2048x2048**		Ensure this is set to the size you want before creating Quick Channels.

Control	Type	What it does	Dialog	Notes
		• **4096x4096** • **8192x8192** • **16384x16384** • **32768x32768**		
<channel depth>	dropdown	Sets the channel depth used when creating a Quick Channel, from the options: • **8bit (Byte)** • **32bit (Float)** • **16bit (Half)**		Ensure this is set to the depth you want before creating Quick Channels.
Channels panel				
	button	Opens an individual palette that displays the layers of that channel only.		
	icon	Indicates whether the channel is tagged for Channels Palette. - The data in the image-set/channel is considered as color data and is being color managed. - The data in the image-set/channel is considered as non-color data, specifically scalar data, and has no color management applied.		
8 16 32	icon	Indicates the color depth of the channel: 8 - 8bit (Byte) 16 - 16bit (Half) 32 - 32bit (Float)		
	button	Indicates an unlocked state. Clicking the icon changes the channel state to locked.		

Control	Type	What it does	Dialog	Notes
	button	Indicates a locked state. Clicking the icon changes the channel state to unlocked.		
	button	Adds a new channel to the channels list.	Add Channel Dialog	
	button	Converts the channel into either **8bit (Byte)**, **16bit (Half)**, **32bit (Float)**.	Convert Type Dialog	
	button	Channels can be shared into the layer stack of another channel in the **Channels** palette. This button shows with which channel it is shared.	Sharing Channels	
	button	Removes the channel.		
General				
File Space	dropdown	Set whether the channel is **Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**.		
Size	information	Lists the size of the patches in the channel.		
Depth	information	Lists the color depth of the channel.		
User Attributes				
MriImport Path	information	Lists the import path of the channel.		
MriImport Template	information	Lists the import file name of the channel.		
Color Data				
Data Colorspace (advanced color management mode only) or	dropdown	The colorspace of the channel. **Automatic** is determined by the file name, size, and type of data in the channel. Changing the colorspace of the channel changes all the color pickers and swatches when working on that channel.		

Control	Type	What it does	Dialog	Notes
Colorspace (intermediate color management mode only)				
Working Colorspace (advanced color management mode only)	dropdown	Select the colorspace to use for painting, lighting, applying filters, and similar operations.		
Output Colorspace (advanced color management mode only)	dropdown	Select the colorspace to use for the output.		
Raw Data (advanced and intermediate color management modes only)	checkbox	The colorspace of the channel masks. **Automatic** is determined by the file name, size, and type of data in the channel. Changing the colorspace of the channel changes all the color pickers and swatches when working on that channel.		
Scalar Data	checkbox	When checked, the data in the image-set/channel is considered as non-color data such as masks, heights, normals, and depths.		

Layers Palette

The **Layers** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

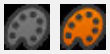
The Layers palette lets you view, create and edit layers. You can also add the following to individual layers, or layer groups:

- masks
- adjustments
- procedurals
- shader layers

> **Note:** For shader layers, the fields that are available to each shader are the same as those in the Shaders Palette section.

Layers Palette Fields

Control	Type	What it does	Notes
Name	dropdown, switch	Filter bar function that gives the choices: **Name**, **Type**, **Attribute**, and **Color Tag**. Depending on which filter option you select, the filter bar shows one of the following: • - filter for the type of layers by clicking on the associated icons. • - filter for the attributes of layers by clicking on the associated icons. • - filter for the color tags of the selected color button.	Both **Type** and **Attribute** provide icons that allow you to set the parameters for your filter. Quickly clear the text field by pressing the button.
Normal	dropdown	Click to select from the blend mode options available for individual layers. To adjust the blend amount, enter the value	This list does not include the advanced blend mode options.

Control	Type	What it does	Notes
		into the text field, or adjust the slider located to the right of the blend mode menu. • - adjusts the blend mode amount.	
	button	Opens the **Advanced Blending** menu, which gives options for blend modes and components, as well as blend amount.	
	button	Click to toggle the selected layers' visibility on the canvas.	
	button	Denotes whether the layer is a paint layer and, when lit, whether it is selected as the current paint target.	
	button	Denotes whether the layer is a mask layer and, when lit, whether the mask is selected as the current paint target.	**Shift**+click on the mask to disable it, or right-click on the layer and select **Layer Mask** > **Disable Mask**. The disabled mask button indicates the disabled state. To enable it, **Shift**+click on the mask again or select **Layer Mask** > **Enable Mask** from the right-click menu.
	button	Denotes whether the layer has a mask stack and, when lit, whether the mask stack is selected as the current layer.	Clicking on the button opens the mask stack

Control	Type	What it does	Notes
			palette.
	button	Specifies whether the layer is an adjustment layer or if the layer contains an adjustment, also called adjustment stack .	
	button	Denotes whether the layer is a procedural layer and, when lit, whether the procedural is selected as the current layer.	
	button	Specifies the layer is a material layer. When lit, the material is selected and when clicked, the node graph opens with the material's sub-graph.	
	button	Denotes whether the layer is a graph layer and, when lit, whether the graph is selected as the current layer.	
	button	Denotes whether the layer is the parent layer of a group.	
	button	Locks or unlocks the selected layer. If the layer is the parent of a group of layers, all the layers in the group are also locked or unlocked. Clicking the padlock button when a layer is cached also unlocks it.	
	button	Specifies that the layer is shared with another layer.	
	button	Specifies that the layer is shared into the layer stack of another channel.	
	button	Adds a new paint layer to the layer stack.	
	button	Adds a mask to the selected layer or, if	If a mask has

Control	Type	What it does	Notes
		none is selected, the top layer in the layer stack.	already been added to a layer, you cannot add a second mask using this method. You must create a mask stack using the right-click menu.
	button	Adds a mask stack to the selected layer, or if none is selected, the top layer in the layer stack.	Clicking on the mask stack button opens the mask stack palette.
	button	Adds an adjustment layer to the layer stack.	Select adjustments from the dropdown menu when you click the button.
	button	Adds an adjustment stack to the selected layer.	Select adjustments from the dropdown menu when you click the button. Clicking on the adjustment stack button opens the adjustment stack palette.
	button	Adds a procedural layer to the layer stack.	Select procedurals from the dropdown menu when you click the button.

Control	Type	What it does	Notes
	button	Adds a parent layer for a group to the layer stack. Additional layers can be added after the parent layer is created.	
	button	Adds a graph layer to the layer stack and Node Graph.	See Adding Graph Layers.
	button	Duplicates the selected layer in the layer stack.	The duplicate has "copy" included in the name so that it can be differentiated from the original.
	button	Removes the selected layer from the layer stack.	
	button	Adds a Multi-Channel layer to the layer stack.	
	button	Assigns a color to the selected layers. To clear an already assigned color, click the button and select the gray color in the center of the palette. This effectively removes the color and returns it to the default color state.	
Adjustment/Brightness			
Brightness	text field, slider	Controls the brightness of the paint on the model.	
Adjustment/Brightness Lookup			
Map	curve editor	Adjusts the brightness to a specific lookup value using a curve map.	
Adjustment/Clamp			
Min	text field, slider	Clamps the minimum RGB value for the selected checkboxes below.	

Control	Type	What it does	Notes
Max	text field, slider	Clamps the maximum RGB value for the selected checkboxes below.	
Components | **R**	checkbox	If enabled, specifies that the red channel should be affected by the **Min** and **Max** sliders.	
Components | **G**	checkbox	If enabled, specifies that the green channel should be affected by the **Min** and **Max** sliders.	
Components | **B**	checkbox	If enabled, specifies that the blue channel should be affected by the **Min** and **Max** sliders.	
Components | **A**	checkbox	If enabled, specifies that the alpha channel should be affected by the **Min** and **Max** sliders.	
Adjustment/Color Balance			
Preserve Luminosity	checkbox	Ensures that the luminosity value is maintained when other values are adjusted.	
Highlights | **Cyan/Red**	text field, slider	Adjusts the scale of the Red color component. Dragging the slider to the left pulls the scale towards cyan, whereas dragging the slider to the right pulls it towards red. The modification applies to the texture's highlights.	
Highlights | **Magenta/Green**	text field, slider	Adjusts the Green color component. Dragging the slider to the left pulls the scale towards magenta, whereas dragging the slider to the right pulls it towards green. The modification applies to the texture's highlights.	
Highlights |	text field,	Adjusts the Blue color component.	

Control	Type	What it does	Notes
Yellow/Blue	slider	Dragging the slider to the left pulls the scale towards yellow, whereas dragging the slider to the right pulls it towards blue. The modification applies to the texture's highlights.	
Shadows | **Cyan/Red**	text field, slider	Adjusts the scale of the Red color component. Dragging the slider to the left pulls the scale towards cyan, whereas dragging the slider to the right pulls it towards red. The modification applies to the texture's shadows.	
Shadows | **Magenta/Green**	text field, slider	Adjusts the Green color component. Dragging the slider to the left pulls the scale towards magenta, whereas dragging the slider to the right pulls it towards green. The modification applies to the texture's shadows.	
Shadows | **Yellow/Blue**	text field, slider	Adjusts the Blue color component. Dragging the slider to the left pulls the scale towards yellow, whereas dragging the slider to the right pulls it towards blue. The modification applies to the texture's shadows.	
Midtones | **Cyan/Red**	text field, slider	Adjusts the scale of the Red color component. Dragging the slider to the left pulls the scale towards cyan, whereas dragging the slider to the right pulls it towards red. The modification applies to the texture's midtones	
Midtones | **Magenta/Green**	text field, slider	Adjusts the Green color component. Dragging the slider to the left pulls the scale towards magenta, whereas	

Control	Type	What it does	Notes
		dragging the slider to the right pulls it towards green. The modification applies to the texture's midtones	
Midtones | **Yellow/Blue**	text field, slider	Adjusts the Blue color component. Dragging the slider to the left pulls the scale towards yellow, whereas dragging the slider to the right pulls it towards blue. The modification applies to the texture's midtones.	
Adjustment/Color Lookup			
Red	curve editor	Adjusts the value of the red channel for the overall layer stack, using the curve editor.	
Green	curve editor	Adjusts the value of the green channel for the overall layer stack, using the curve editor.	
Blue	curve editor	Adjusts the value of the blue channel for the overall layer stack, using the curve editor.	
Adjustment/Color Switch			
Red	checkbox	Turns the red channel on or off. Disabling the red channel removes it from the painting.	
Green	checkbox	Turns the green channel on or off. Disabling the green channel removes it from the painting.	
Blue	checkbox	Turns the blue channel on or off. Disabling the blue channel removes it from the	

Control	Type	What it does	Notes
		painting.	
Alpha	checkbox	Turns the alpha channel on or off. Disabling the alpha channel removes it from the painting.	
Adjustment/Color to Mask			
Color	swatch	Select a color from the color picker to convert that color to a mask. This allows a multicolored channel to be used for multiple masks.	
Error	text field, slider	The tolerance level of the mask to the selected color. The lower the number, the closer the RGB value has to be to the selected color in order for it to be treated as a mask; the higher the number, the more variation that's allowed.	
Adjustment/Contrast			
Contrast	text field, slider	Adjusts the contrast for the layer stack.	
Contrast Pivot	text field, slider	Sets the point around which **Contrast** is adjusted.	
Adjustment/Copy Channel			
Channel	dropdown	Copies the value from one RGB color channel to the other two. The result is a grayscale image with the intensity values from the selected channel.	
Adjustment/Flow			
Image| **Tile Image**	file browser	Selects the image to use on the model for flow.	

Control	Type	What it does	Notes
Image| **Repeat**	text field, slider	Sets the repetition value of the tiled image.	
Animation | **Animated**	checkbox	Enables or disables the animation.	
Animation| **Time Offset**	text field, slider	Adjusts the flow over time to view the effect of vector paint.	
Animation| **Speed**	text field, slider	Adjusts the sensitivity of **Time Offset** to affect the speed of flow.	
Adjustment/Gamma			
Gamma	text field, slider	Adjusts the gamma levels in the layer stack.	
Invert	text field, slider	Reverses the gamma conversion. That is, a higher setting for **Gamma** results in a more washed-out looking image.	
Adjustment/Grade			
Blackpoint	text field, slider	Sets the blackpoint of the RGB values.	Blackpoint is the color value at which the input is considered to be 100% black.
Whitepoint	text field, slider	Sets the whitepoint of the RGB values.	Whitepoint is the color value at which the input is considered to be 100% white.
Lift	text field, slider	Lifts the blackpoint, while keeping the whitepoint the same.	
Gain	text field, slider	Adjusts the whitepoint, while keeping the blackpoint the same	
Multiply	text field,	Multiplies the value to lighten the texture	

Control	Type	What it does	Notes
	slider	while preserving the blackpoint.	
Offset	text field, slider	Specifies a fixed value to add in order to lighten the texture.	Adding negative values essentially darkens the texture.
Gamma	text field, slider	Adjusts the midtones, while keeping the whitepoint and blackpoint the same.	
Adjustment/HSL			
Hue Shift	text field, slider	Adjusts the hue, as tied to the HSL, for the layer stack.	
Saturation Scale	text field, slider	Adjusts the saturation level, as tied to the HSL, for the layer stack.	
Lightness Scale	text field, slider	Adjusts the lightness level, as tied to the HSL, for the layer stack.	
Adjustment/HSV			
Hue Shift	text field, slider	Adjusts the hue level, as tied to the HSV, for the layer stack.	
Saturation Scale	text field, slider	Adjusts the saturation level, as tied to the HSV, for the layer stack.	
Value Scale	text field, slider	Adjusts the value level, as tied to the HSV, for the layer stack.	
Adjustment/Height As Normal			
Bump Weight	text field, slider	Adjusts the sensitivity of the bump weight when calculating height as a normal value.	
Bump Mode	dropdown	Displays the bump quickly (**Fast**) or accurately (**Accurate**).	By default, Bump Mode is set to **Fast**.

Control	Type	What it does	Notes
Bump Space	dropdown	**UV** - Normals are calculated relative to the UV coordinates space. The **UV Bump Space** mode is the more traditional approach but can lead to noticeable seams and color variations. **World** - Normals are calculated relative to the World coordinates space. Using the **World Bump Space** mode gives a more seamless result.	By default, **Bump Space** is set to **UV**.
Adjustment/Hue Shift			
Hue	text field, slider	Changes the RGB hue value of the paint in the layer stack.	
Adjustment/Invert			
Red	checkbox	Inverts the contents of the red color component when checked. Uncheck the parameter to leave the component unchanged. If there is a significant amount of red paint in your layer stack the inverse is a significant absence of red.	It is uncommon that only one component would be selected, however if the need arises, you can choose to invert only one, or a subset, of the four components.
Green	checkbox	Inverts the contents of the green color component when checked. Uncheck the parameter to leave the component unchanged. If there is a significant amount of green paint in your layer stack the inverse is a significant absence of green.	It is uncommon that only one component would be selected, however if the need arises, you can choose to invert only one, or a

Control	Type	What it does	Notes
			subset, of the four components.
Blue	checkbox	Inverts the contents of the blue color component when checked. Uncheck the parameter to leave the component unchanged. If there is a significant amount of blue paint in your layer stack the inverse is a significant absence of blue.	It is uncommon that only one component would be selected, however if the need arises, you can choose to invert only one, or a subset, of the four components.
Alpha	checkbox	Inverts the contents of the alpha color component when checked. Uncheck the parameter to leave the component unchanged.	It is uncommon that only one component would be selected, however if the need arises, you can choose to invert only one, or a subset, of the four components.
Adjustment/Levels			
Clamp		Set whether to: • **Clamp Output Levels** - to clamp the output values to between the white and blackpoints, • **Clamp LDR** - to clamp the output values to the 0-1 range, which is Byte black to white, or • **Disable** - to prevent Mari from clamping any of the levels.	

Control	Type	What it does	Notes
Red	checkbox	Determines whether to apply the adjustment to the red component.	
Green	checkbox	Determines whether to apply the adjustment to the green component.	
Blue	checkbox	Determines whether to apply the adjustment to the blue component.	
Output Level | **White**	text field, slider	Adjusts the white level of the output in the paint for the selected color components.	
Output Level | **Black**	text field, slider	Adjusts the black level of the output in the paint for the selected color component.	
Input Level | **Mid Mode**	dropdown	Choose whether levels adjustments are made under **Absolute** or **Relative** mode. **Absolute** is the default mode for Mari; changing the mode to **Relative** changes the *Input* | **Mid** point to be relative to the black and white points.	
Input Level | **White**	text field, slider	Adjusts the white level of the input for the selected color component.	
Input Level | **Mid**	text field, slider	Adjusts the midtone level of the input for the selected color components.	
Input Level | **Black**	text field, slider	Adjusts the black level of the input for the selected color component.	
Adjustment/Luminosity			
Adjustment/Premultiply Alpha			
Unpremultiply	checkbox	Either pre- or post-multiplies the alpha in the selected image. If you are painting using an image without premultiplied	

Control	Type	What it does	Notes
		alpha onto one with it, use this filter to perform the premultiplication, so the images match and you avoid lines around the outside of the patch. Post-multiply works the same, but in reverse: it removes premultiplication to match images that do not have premultiplied alpha.	
Adjustment/Saturation			
Saturation	text field, slider	Adjusts the color saturation of the paint in the layer stack.	
Adjustment/Scale			
R	text field, slider	Adjusts the scale of the Red color component. Dragging the slider to the left pulls the scale towards cyan, whereas dragging the slider to the right pulls it towards red.	
G	text field, slider	Adjusts the Green color component. Dragging the slider to the left pulls the scale towards magenta, whereas dragging the slider to the right pulls it towards green.	
B	text field, slider	Adjusts the Blue color component. Dragging the slider to the left pulls the scale towards yellow, whereas dragging the slider to the right pulls it towards blue.	
A	text field, slider	Adjusts the Alpha component.	
Adjustment/Set Value			
Channel	dropdown	The options available in the dropdown	

Control	Type	What it does	Notes
		menu are Red, Green, Blue, Alpha.	
Value	text field, slider	Adjusting the slider modifies the value of the component selected from the dropdown menu above.	
Adjustment/Shuffle			
R	dropdown	Select a color component from the options **r**, **g**, **b**, and **a** in the dropdown menu to replace the Red component with another color. Setting **r** in the dropdown menu keeps the red color in the Red component.	
G	dropdown	Select a color component from the options **r**, **g**, **b**, and **a** in the dropdown menu to replace the Green component with another color. Setting **g** in the dropdown menu keeps the green color in the Green component.	
B	dropdown	Select a color component from the options **r**, **g**, **b**, and **a** in the dropdown menu to replace the Blue component with another color. Setting **b** in the dropdown menu keeps the blue color in the Blue component.	
A	dropdown	Select a color component from the options **r**, **g**, **b**, and **a** in the dropdown menu to replace the Alpha component with another color. Setting **a** in the dropdown menu keeps the alpha color in the Alpha component.	
Adjustment/Tangent To Screen			
Suppress Blue	checkbox	Suppresses the blue value from the color-encoded vectors in screen space.	

Control	Type	What it does	Notes
Adjustment/Tangent To World			
Suppress Blue	checkbox	Suppresses the blue value from the color-encoded vectors in world space.	
Adjustment/World To Tangent			
Suppress Blue	checkbox	Suppresses the blue value from the color-encoded vectors in world space.	
Flip Y	dropdown	Choose whether to **Flip** the Y axis normals or leave them as they are with **normal**.	
Adjustment/sRGB2Linear			
Invert	checkbox	Applies an sRGB to linear colorspace conversion. Checking **Invert** applies a linear to sRGB colorspace conversion.	
Procedural/Basic/Color			
Color	swatch	Applies the specified color from the color swatch to the entire model.	
Procedural/Basic/Constant			
Constant	text field, slider	Generates a constant number across the **RGB** or **RGBA** components that may be relied upon for mathematical operations.	This number ranges from 0 to 10,000, and defaults to 1.000.
Components	dropdown	Choose whether the contact applies to the **RGB** or **RGBA** components.	
Procedural/Basic/Vector			
X	text field, slider	Adjusts the values of the X axis, which correspond to the Red color component, to modify the color-encoded vectors.	
Y	text field,	Adjusts the values of the Y axis, which	

Control	Type	What it does	Notes
	slider	correspond to the Green color component, to modify the color-encoded vectors.	
Z	text field, slider	Adjusts the values of the Z axis, which correspond to the Blue color component, to modify the color-encoded vectors.	
W	text field, slider	Adjusts the values of the W axis, which correspond to the Alpha component, to modify the color-encoded vectors.	
Procedural/Projection/Cubemap			
Cubic Image	text field, file browser	The environment image you want to be reflected.	This procedural cannot be cached or baked and is for display purposes only.
Falloff Start	text field, slider	Specifies where the cubic image starts.	
Falloff End	text field, slider	Specifies where the cubic image ends.	
Procedural/Projection/Cubemap Projector			
Cubic Image	text field, file browser	Select or input the filepath to the **.dds** cubic image to project as a texture.	This procedural cannot be cached or baked and is for display purposes only. Note: When exporting **.dds** files, they should be encoded as 8.8.8.8 ARGB files to avoid any

Control	Type	What it does	Notes
			complications when bringing them into Mari.
Cull Backfaces	checkbox	When the checkbox is ticked, this setting ensures that projection does not affect areas facing away from the camera when factoring edge falloff.	
Offset | **X**	text field, incrementer	The offset of the projected image along the **X** axis. This affects the position of where the camera sits in the scene.	
Offset | **Y**	text field, incrementer	The offset of the projected image along the **Y** axis. This affects the position of where the camera sits in the scene.	
Offset | **Z**	text field, incrementer	The offset of the projected image along the **Z** axis. This affects the position of where the camera sits in the scene.	
Rotation | **X**	text field, slider	The rotation of the projected image along the **X** axis.	This rotates the camera for the entire scene, not just one part.
Rotation | **Y**	text field, slider	The rotation of the projected image along the **Y** axis.	This rotates the camera for the entire scene, not just one part.
Rotation | **Z**	text field, slider	The rotation of the projected image along the **Z** axis.	This rotates the camera for the entire scene, not just one part.
Edge Falloff | **Start**	text field, slider	Modifies how far away the projection falloff starts on the model.	
Edge Falloff | **End**	text field,	Modifies how far away the projection	

Control	Type	What it does	Notes
	slider	falloff ends on the model.	
Edge Falloff | **Curve**	curve editor	Modifies the otherwise linear shape of the edge falloff to your desired shape.	
Distance Falloff | **Start**	text field, incrementer	Modifies the falloff start distance. From 0-100; 100 represents straight-on projection.	
Distance Falloff | **End**	text field, incrementer	Modifies the falloff end distance. From 0-100; 100 represents straight-on projection.	
Distance Falloff | **Curve**	curve editor	Modifies the otherwise linear shape of the distance falloff to your desired shape.	
Procedural/Projection/Sphere Map			
Sphere Map	file browser	The environment image you want to be reflected.	This procedural cannot be cached or baked and is for display purposes only.
Falloff Start	text field, slider	Specifies where the spherical image starts.	
Falloff End	text field, file browser	Specifies where the spherical image ends.	
Procedural/Projection/Sphere Map Projector			
Spherical Image	file browser	Uses a spherical image of any file type as a texture to project onto the model. Select the filepath to the spherical image to project as a texture.	
Cull Backfaces	checkbox	When the checkbox is ticked, this setting ensures that projection does not affect	

Control	Type	What it does	Notes
		areas facing away from the camera when factoring edge falloff.	
Offset | **X**	text field, incrementer	The offset of the projected image along the **X** axis. This affects the position of where the camera sits in the scene.	
Offset | **Y**	text field, incrementer	The offset of the projected image along the **Y** axis. This affects the position of where the camera sits in the scene.	
Offset | **Z**	text field, incrementer	The offset of the projected image along the **Z** axis. This affects the position of where the camera sits in the scene.	
Rotation | **X**	text field, slider	The rotation of the projected image along the **X** axis.	This rotates the camera for the entire scene, not just one part.
Rotation | **Y**	text field, slider	The rotation of the projected image along the **Y** axis.	This rotates the camera for the entire scene, not just one part.
Rotation | **Z**	text field, slider	The rotation of the projected image along the **Z** axis.	This rotates the camera for the entire scene, not just one part.
Edge Falloff | **Start**	text field, slider	Modifies how far away the projection falloff starts on the model.	
Edge Falloff | **End**	text field, slider	Modifies how far away the projection falloff ends on the model.	
Edge Falloff | **Curve**	curve editor	Modifies the otherwise linear shape of the edge falloff to your desired shape.	
Distance Falloff | **Start**	text field, incrementer	Modifies the falloff start distance. From 0-100; 100 represents straight-on	

Control	Type	What it does	Notes
		projection.	
Distance Falloff | **End**	text field, incrementer	Modifies the falloff end distance. From 0-100; 100 represents straight-on projection.	
Distance Falloff | **Curve**	curve editor	Modifies the otherwise linear shape of the distance falloff to your desired shape.	
Procedural/Projection/Triplanar Projection			
World Scale	text field	Gives overall scale control for the image projection.	
Top | **Top Image**	file browser	Specifies the location of the image you want to project onto the top of your model.	
Top | **Top Repeat**	text field, slider	Specifies the frequency at which the image is repeated across the top of your model.	
Top | **Top Angle**	text field, slider	Specifies the rotation angle of the image on the top of your model.	
Top | **Top U Offset**	text field, slider	Specifies how much the image on the top of your model is offset by on the U axis.	
Top | **Top V Offset**	text field, slider	Specifies how much the image on the top of your model is offset by on the V axis.	
Top | **Top U Scale**	text field, slider	Specifies how much the image on the top of your model is stretched or contracted on the U axis.	
Top | **Top V Scale**	text field, slider	Specifies how much the image on the top of your model is stretched or contracted on the V axis.	
Top | **Top Falloff Start**	text field,	Controls where the image projection	

Control	Type	What it does	Notes
	slider	starts on the top of the model.	
Top | **Top Falloff End**	text field, slider	Controls where the image projection ends on the top of the model.	
Top | **Top Falloff**	curve editor	Controls the falloff of the image projection between the start and end values on the top of the model.	
Front | **Front Image**	file browser	Specifies the location of the image you want to project onto the front of your model.	
Front | **Front Repeat**	text field, slider	Specifies the frequency at which the image is repeated across the front of your model.	
Front | **Front Angle**	text field, slider	Specifies the rotation angle of the image on the front of your model.	
Front | **Front U Offset**	text field, slider	Specifies how much the image on the front of your model is offset by on the U axis.	
Front | **Front V Offset**	text field, slider	Specifies how much the image on the front of your model is offset by on the V axis.	
Front | **Front U Scale**	text field, slider	Specifies how much the image on the front of your model is stretched or contracted on the U axis.	
Front | **Front V Scale**	text field, slider	Specifies how much the image on the front of your model is stretched or contracted on the V axis.	
Front | **Front Falloff Start**	text field, slider	Controls where the image projection starts on the front of the model.	
Front | **Front Falloff End**	text field, slider	Controls where the image projection ends on the front of the model.	

Control	Type	What it does	Notes
Front | **Front Falloff**	curve editor	Controls the falloff of the image projection between the start and end values on the front of the model.	
Right | **Right Image**	file browser	Specifies the location of the image you want to project onto the side of your model.	
Right | **Right Repeat**	text field, slider	Specifies the frequency at which the image is repeated across the side of your model.	
Right | **Right Angle**	text field, slider	Specifies the rotation angle of the image on the side of your model.	
Right | **Right U Offset**	text field, slider	Specifies how much the image on the side of your model is offset by on the U axis.	
Right | **Right V Offset**	text field, slider	Specifies how much the image on the side of your model is offset by on the V axis.	
Right | **Right U Scale**	text field, slider	Specifies how much the image on the side of your model is stretched or contracted on the U axis.	
Right | **Right V Scale**	text field, slider	Specifies how much the image on the side of your model is stretched or contracted on the V axis.	
Right | **Right Falloff Start**	text field, slider	Controls where the image projection starts on the side of the model.	
Right | **Right Falloff End**	text field, slider	Controls where the image projection ends on the side of the model.	
Right | **Right Falloff**	curve editor	Controls the falloff of the image projection between the start and end values on the side of the model.	

Control	Type	What it does	Notes
Procedural/Lighting/Environment Light			
Sharpness	text field, slider	Adjusts the sharpness of the reflections on the model, with higher values increasing the sharpness and lower values decreasing it.	This procedural cannot be cached or baked and is for display purposes only.
Procedural/Geometry/Ambient Occlusion			
Procedural/Geometry/BiTangent			
Procedural/Geometry/Geo-Channel			
Channels	dropdown box	Selects a geo-channel for the selected Geo-channel layer.	
Background Color	swatch	Adds a background color to the Geo-channel.	This is useful when you have a geometry that has UDIMS 1001-1003, but your Geo-channel only has 1001-1002. 1003 is filled with the background color in this instance.
Procedural/Geometry/Position			
Procedural/Geometry/Selection Mask			
Procedural/Geometry/Surface Normal			
Procedural/Geometry/Tangent			
Procedural/Geometry/UDIM Mask			
UDIM	text field	Enter the UDIM number of the UDIM you want to unmask.	

Control	Type	What it does	Notes
Procedural/Geometry/UV			
Procedural/Geometry/UV Mask			
Procedural/Procedural/Fractal/Cloud			
Size	text field, slider	Adjusts the size of the cloud pattern.	
Roughness	text field, slider	Adjusts the roughness of the cloud pattern. Dragging the slider to lower values smooths out the transition from color A to color B.	
Offsets | **X Offset**	text field, slider	Moves the pattern across the model on the x axis.	
Offsets | **Y Offset**	text field, slider	Moves the pattern across the model on the y axis.	
Offsets | **Z Offset**	text field, slider	Moves the pattern across the model on the z axis.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's B input.	
Procedural/Procedural/Fractal/Turbulence			
Size	text field, slider	Adjusts the size of the turbulence pattern.	
Roughness	text field, slider	Adjusts the roughness of the turbulence pattern. Dragging the slider to lower values smooths out the transition from color A to color B.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's A input.	

Control	Type	What it does	Notes
Color | **Color B**	swatch	Sets the color to be used for the pattern's B input.	
Procedural/Procedural/Misc/Oil			
Size	text field, slider	Adjusts the size of the oil pattern.	
Organicness	text field, slider	Sets the overall organic quality of the oil pattern.	
Roughness	text field, slider	Adjusts the roughness of the oil pattern. Dragging the slider to lower values smooths out the transition from color A to color B.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's B input.	
Procedural/Procedural/Noise/Cellular			
Size	text field, slider	Adjusts the size of the cellular pattern.	
Type	dropdown	Sets how color A and color B are used in the pattern.	
Distance Method	dropdown	Changes the shape of the cells to affect the distance the cells are from each other.	
Color | **Color A**	swatch	Sets the color to be used to the pattern's color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	
Procedural/Procedural/Noise/Perlin			
Size	text field,	Adjusts the size of the noise pattern.	

Control	Type	What it does	Notes
	slider		
Color | **Color A**	swatch	Sets the color to be used for the pattern's color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	
Procedural/Procedural/Noise/Squiggle			
Size	text field, slider	Adjusts the size of the squiggle pattern.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	
Procedural/Procedural/ Pattern/Cube			
Repeat	text field, slider	Specifies the repetition frequency of the cube pattern.	
Gap	text field, slider	Sets the spacing between the cubes in the pattern.	
Fall Off	text field, slider	Adjusts the falloff amount between the cubes.	
Roundness	text field, slider	Adjusts how much the cubes are rounded off.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	
Procedural/Procedural/Pattern/Object Space Linear Gradient			
Start	text field, slider	Sets the start point on the object for Color B of the gradient.	

Control	Type	What it does	Notes
End	text field, slider	Sets the end point of Color B, where Color A begins, on the object of the gradient.	
Map	curve editor	Set the color A and color B transitions for the gradient using points on a curve.	
Rotation | **X**	text field, slider	Adjusts how the gradient appears on the model by rotating the gradient on the X axis.	
Rotation | **Y**	text field, slider	Adjusts how the gradient appears on the model by rotating the gradient on the Y axis.	
Rotation | **Z**	text field, slider	Adjusts how the gradient appears on the model by rotating the gradient on the Z axis.	
Color | **Color A**	swatch	Sets the color to be used for the gradient's A input.	
Color | **Color B**	swatch	Sets the color to be used for the gradient's B input.	
Procedural/Procedural/Pattern/Sphere			
Size	text field, slider	Adjusts the size of the spherical pattern.	
Falloff	curve editor	Adjusts the falloff amount between the spheres.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	
Procedural/Procedural/Pattern/Tiled			
Tile Image	file browser	The image you want to have tiled across the model.	

Control	Type	What it does	Notes
Rotation Angle (Degrees)	text field, slider	Changes the angle of rotation for the tiled image.	
Offset | **U Offset**	text field, slider	Specifies how much the image is offset by on the U axis.	
Offset | **V Offset**	text field, slider	Specifies how much the image is offset by on the V axis.	
Repeat | **U Repeat**	text field, slider	Specifies how much the image is repeated on the U axis.	
Repeat | **V Repeat**	text field, slider	Specifies how much the image is repeated on the V axis.	
Mirror | **Mirror U**	text field, slider	Sets whether the tiled image is mirrored on the U axis.	
Mirror | **Mirror V**	text field, slider	Sets whether the tiled image is mirrored on the V axis.	
Procedural/Procedural/Pattern/UV Grid			
Color	swatch	Sets the color to be used for the grid lines.	
Repeat	text field, slider	Specifies the repeat rate for the grid on the model's surface.	
Thickness	text field, slider	Sets the thickness of the grid lines.	
Procedural/Procedural/Pattern/UV Linear Gradient			
Start	text field, slider	Sets the start point on the UV patches for Color B of the gradient.	
End	text field, slider	Sets the end point of Color B, where Color A begins, on the UV patches of the gradient.	
Map	curve editor	Set the color A and color B transitions for	

Control	Type	What it does	Notes
		the gradient in UV space using points on a curve.	
Angle	text field, slider	Specifies the angle of the linear gradient on the UV patches.	
Repeat	text field, slider	Sets how much the linear gradient is repeated across the UV patches.	
Color | **Color A**	swatch	Sets the color to be used for the gradient's A input.	
Color | **Color B**	swatch	Sets the color to be used for the gradient's B input.	
Procedural/Procedural/Plant/Wood			
Size	text field, slider	Adjusts the size of the wood pattern across the entire model.	
Bands	text field, slider	Adjusts the size of the individual bands in the wood pattern.	
Noise | **Size**	text field, slider	Adjusts the amount of the noise in the bands of the wood pattern.	
Noise | **Roughness**	text field, slider	Adjusts the roughness of the wood pattern. Dragging the slider to lower values makes the transition from color A to color B more distinct.	
Stretch | **X**	text field, slider	Stretches the wood pattern across the model on the x axis.	
Stretch | **Y**	text field, slider	Stretches the wood pattern across the model on the y axis.	
Stretch | **Z**	text field, slider	Stretches the wood pattern across the model on the z axis.	
Color | **Color A**	swatch	Sets the color to be used for the pattern's	

Control	Type	What it does	Notes
		color A input.	
Color | **Color B**	swatch	Sets the color to be used for the pattern's color B input.	

Multi-Paint Palette

The **Multi-Paint** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

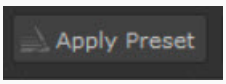
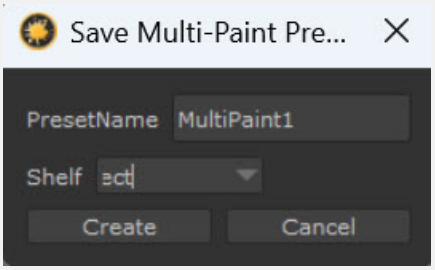
What It Does

The Multi-Paint palette enables you to define the stream settings for a Multi-Channel Paint node.

To learn more, go to Multi-Paint and Multi-Channel Paint Node.

Multi-Paint Palette Fields

Control	Type	What it does	Notes
Toggle Multi-Paint	button	Toggles Multi-paint on or off.	
Show all / active streams	button	Toggles visibility between all of the available shader streams and only the active shader streams. Active shader streams are defined by the tick button ( ) selected.	

Control	Type	What it does	Notes
Search streams	Text input	Search streams in the currently-selected shader model.	
Apply preset 	button	Apply pre-defined preset for the shader model. See more at Multi-Paint Presets.	Hover over available presets to see details.
Save as Preset	button	Saves the current settings as a Preset. Opens up pop up window to define the **Preset Name** and **Shelf** to store the preset on. 	
Reset Shader	button	Resets shader to default values, clearing all inputs.	
Active Stream	button	Defines whether the stream is active in the Multi-Channel Paint node. When selected, the stream will be painted into when the Multi-Channel Paint node is the active painting target.	
Primary Stream	button	When selected (icon is white), the primary stream is the only visible stream while multi-painting. Icon is gray when stream is not selected ( ).	
Target	text	The shader stream to assign an image/ color to. Stream names depend on the shader model chosen when the Multi-Channel pain node is created.	
Image	Image input	Defines the image for the current stream. When clicked, opens an Image Manager pop up to	

Control	Type	What it does	Notes
		select the image for the stream. To choose an image for the stream, select it and press **Ok**.	
Color	Color selector/ button	Define a foreground and background color for the stream. Swap the colors using the swap button ( ).	
Blend mode	Drop down	Define the blend node for the stream. For all available blend options, see Paint Blending Modes.	

Node Graph Palette

The **Node Graph** palette doesn't have any controls as such, rather it's a dedicated area designed to hold nodes.

What It Does

The **Node Graph** palette lets you create, edit, and view node trees and graph layers.

Node Properties Palette

The **Node Properties** palette doesn't have any controls as such, rather it's a dedicated area designed to hold the controls associated with nodes in the Node Graph.

What It Does

The **Node Properties** palette lets you view and modify node properties. To view a node's properties, double-click the node in the **Node Graph** palette. See the *Mari User Guide, Node Graph* chapter for more information.

Patches Palette

The **Patches** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Patches** palette displays a selectable list of patches in the project, with icons and buttons to toggle whether they are visible, or whether they are locked. You can group patches into sets.

Patches Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
Sorted by	dropdown	Lets you specify whether to list patches by: UDIM, or a custom sort (by dragging patches up and down the list).		
	icon	Click to toggle whether a patch is visible or not.		
	icon	Click to toggle whether a patch is locked or not.		

Control	Type	What it does	Opens this dialog box	Notes
	button	Hide selected patches.		
	button	Show selected patches.		
	button	Select all patches.		
	button	Lock selected patches.		
	button	Unlock selected patches.		
	button	Link selected patches. This allows you to replicate changes made to one patch onto all linked patches quickly and easily.		
	button	Unlink selected patches. **Note:** Unlinking does not set the patches back to their original texture.		

Objects Palette

The **Objects** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Objects** palette displays the current objects in the project, and allows you to:

- Show and hide the objects.
- Add and remove objects and child objects from the project.

- Lock and unlock the objects.
- Select the object to edit.
- Add versions to existing objects.
- Add locators to objects.
- Add geo-channels to objects.

When opening the Objects Palette, two tabs are visible - **Main** and **2D**.

When clicking **Main**, it displays the objects in 3D mode as normal.

Under **2D**, any 2D images added to the Image Manager appear as a plane here. If you right click on an object in 2D mode, there are two additional options - **Export 2D Paint** and **Import 2D Paint**. This allows you to save and import the node graph as a **.mrimg** file type, enabling you to save the 2D node tree / layer stack to disk separate to the main Mari cache.

Objects Palette Fields

Control	Type	What it does	Notes
Sorted by	dropdown	Lets you specify whether to list the objects by name , version number, or none.	
/	icon	Click to toggle whether an object is visible or not.	
/	icon	Click to toggle whether an object is locked or not.	
	button	Add another object to the project.	
	button	Add a locator to the object to rotate, scale, or move it.	
	button	Removes the selected object or locator from the project.	
	button	Allows you to add user attributes to the selected object.	
<object name> > Geometry |	dropdown	List of versions of the object, listed	

Control	Type	What it does	Notes
Version		by the name of the source file. When you select a version from this list, Mari updates to show that version in the display.	
<object name> > Geometry | **Path**	text	The filepath to the geometry.	
<object name> > Geometry | **User Attributes**	expand/collapse	Any custom attributes that are active on the object. For more information, see **About Custom Attributes** in Extending Mari.	
<object name> > Info | **Name**	text	The name of the object file.	You can edit this if required.
<object name> > Rendering | **Cast Shadows**	checkbox	Specify whether or not the object is set to cast shadows when **Shadow Maps** are enabled and toggle shadows is set to show them on the canvas.	
<object name> > State | **Hidden**	checkbox	Whether the object is hidden.	
<object name> > State | **Locked**	checkbox	Whether the object is locked or editable.	
<object name> > Subdivision | **Level**	dropdown	Indicates how dense the mesh is, once divided, with 3 being the highest level of subdivision. The number of subdivided quads or triangles for each level depends on the Subdivision Scheme used.	
<object name> > Subdivision | **Scheme**	text	Specifies the subdivision model used when subdividing the object: • **Catmull Clark** - Mari uses the	

Control	Type	What it does	Notes
		Catmull-Clark subdivision model when subdividing a mesh. • **Loop** - Mari uses the Loop subdivision model when subdividing a mesh. Loop subdivision supports meshes consisting of only triangles. • **Bilinear** - Mari uses the Bilinear subdivision model when subdividing a mesh.	
<object name> **>** *Subdivision* **| Triangle**	text	Rule added to the Catmull-Clarck scheme that can be applied to all triangular faces: • **Tri sub catmark** - default Catmark scheme weights. • **Tri sub smooth** - smooths triangle weights.	
<object name> **>** *Subdivision* **| Creasing**	text	Interpolates the sharpness of incident edges: • **Uniform** - applies regular semi-sharp crease rules. • **Chaikin** - applies Chaikin semi-sharp crease rules. Chaikin's curve subdivision algorithm improves the appearance of multi-edge semi-sharp creases with varying weights.	
<object name> **>** *Subdivision* **| Face-Varying**	text	Specifies the face varying interpolation rule: • **None** - Smooths everywhere the mesh is smooth. • **Corners only** - sharpens (linearly interpolate) corners only.	

Control	Type	What it does	Notes
		• **Corners plus1** - **Corners only** + sharpening of junctions of 3 or more regions. • **Corners plus2** - **Corners plus1** + sharpening of darts and concave corners. • **Boundaries** - linear interpolation along the boundary edges and corners • **All** - linear interpolation of everywhere (boundaries and interior).	
<object name> > *Subdivision* | **Boundary**	text	Specifies how the boundary edges and vertices are interpolated. • **Bilinear (no smoothing)** - no smoothing on interior or boundaries. • **Sharp Edges & Smooth Corners** - smooth interior, corners, and corners with more than two edges, but sharp edges. • **Sharp Edges & Sharp Corners** - smooth interior and corners with more than two edges, but sharp edges and corners. • **Sharp Boundaries & Smooth Interiors** - smooth interior, but sharp edges, corners, and corners with more than two edges.	
<object name> > *Subdivision* | **Subdivide**	button	Subdivides the object and display a subdivision progress bar. The subdivision **Level** chosen is applied in the Viewer when the progress bar is finished.	See Subdivide Dialog.

Control	Type	What it does	Notes
<object name> > User Attributes	info	Information about the object, including creation date, modification date, and the owner name.	
<object name> > Geo-channel Properties > **Channels**	button	Adds a new geo-channel to your object.	
*<object name> > Geo-channel Properties	<channel name>* >	button	Opens a dialog and allows you to choose an image set to add to the specified geo-channel.	
*<object name> > Geo-channel Properties	<channel name>* >	button	Removes the current image set from the specified geo-channel.	
*<object name> > Geo-channel Properties	<channel name>* >	button	Removes the specified geo-channel.	
2D			
<object name> > Info	**Name**	text	The name of the object file.	
<object name> > State	**Hidden**	checkbox	Whether the object is hidden.	
<object name> > State	**Locked**	checkbox	Whether the object is locked or editable.	
<object name> > User Attributes	**Owner**	text	Specifies the user who is the owner of the image.	
Create New 2D Paint	button	Opens a blank image to start painting in 2D.	
Export 2D Paint	(right click) option	This allows you to save the node	

Control	Type	What it does	Notes
		graph as a .mrimg file type.	
Import 2D Paint	(right click) option	This allows you to import a node graph as a .mrimg file type.	

Shaders Palette

The **Shaders** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Shaders** palette displays the lighting shaders and allows you to toggle between the shader setup for the current channel, layer stack, layer, material, and paint target, as well as custom-made shaders.

Shaders Palette Fields

Control	Type	What it does	Dialog	Notes
Current Channel	shader	Default shader that displays only the paint in the selected channel.		This shader can also be found on the **Project Controls** toolbar.
Current Layer and Below	shader	Default shader that displays only the paint in your selected sub-stack (such as a mask or adjustment stack) up to the selected layer. If you don't have a sub-stack selected, it shows the parent stack up to the selected layer.		This shader can also be found on the **Project Controls** toolbar.
Current Layer	shader	Default shader that displays only the paintable paint and masks in your		This shader can

Control	Type	What it does	Dialog	Notes
		selected layer.		also be found on the **Project Controls** toolbar.
Current Paint Target	shader	Default shader that displays only the paint in your selected layer, including any mask or mask stack, adjustment, filter, or groups associated with the layer.		This shader can also be found on the **Project Controls** toolbar.
Current Material	shader	Default shader that displays the currently selected material, node, or group. This allows you to display your materials in isolation of the rest of your channel data, allowing you to make tweaks with a more lightweight shader.		This shader can also be found on the **Project Controls** toolbar.
	button	Adds a new shader or layered shader to the shader list in the palette from a dropdown menu list of: • **Unreal**, • **Principled BRDF**, • **VRayMtl**, • **Arnold Standard Surface**, • **3Delight Principled**, • **Layered**, and • **Choose Diffuse And Specular**.	Create Shader Dialog when selecting **Choose Diffuse and Specular**. Masks Formats Dialog when selecting **Layered**.	The **Choose Diffuse and Specular** option allows you to create a custom shader. Refer to the Create Shader Dialog for more information**.** The **Layered** option creates a layered shader. When selected, its layers can be accessed using the Shader Layers Tab Fields.
	button	Makes a copy of the selected shader.		

Control	Type	What it does	Dialog	Notes
	button	Removes the shader from the shader list in the palette.		Default shaders cannot be removed from the **Shaders** palette.
	button	Selects the default **Current Channel** shader from the shader list and displays only the paint in the current channel.		
	button	Selects the default **Current Layer and Below** shader from the shader list and displays only the paint in the current layer stack.		
	button	Selects the default **Current Layer** shader from the shader list and displays only the paint in the current layer.		
	button	Selects the default **Current Paint Target** shader from the shader list and displays only the paint of the currently selected target.		

Shader Parameters

For each individual shader's parameters, refer to the specific shader in this section.

3Delight Principled

The 3Delight Principled material is 3Delight's interpretation of the Principled BRDF shader. The 3Delight Principled material adheres to the following principles:

• Intuitive rather than physical parameters should be used.

- There should be as few parameters as possible.
- Parameters should be zero to one over their plausible range.
- Parameters should be allowed to be pushed beyond their plausible range where it makes sense.
- All combinations of parameters should be as robust and plausible as possible.

This page is based on the 3Delight Principled shader documentation, for more information, see https://www.3delight.com/documentation/display/sfrp/3delight+principled.

Inputs

Control	Type	What it does	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input. The button to the right of the toggle turns green when a channel is selected on the specific input.	**Displacement Map** is toggled off by default.
	button	Creates a new Channel.	
Base Color	dropdown	The channel controlling the base color effect of the shader.	
Base Roughness	dropdown	The channel controlling the base roughness effect of the shader.	
Base Specular Level	dropdown	The channel controlling the base specular level effect of the shader.	
Metallic	dropdown	The channel controlling the metallic effect of the shader.	
Anisotropy	dropdown	The channel controlling the anisotropy effect of the shader.	
Anisotropy Direction	dropdown	The channel controlling the anisotropy direction effect of the shader.	
Opacity	dropdown	The channel controlling the opacity effect of the shader.	

Coat Thickness	dropdown	The channel controlling the coat thickness effect of the shader.	Has no perceivable effect in the Mari viewport.
Coat Color	dropdown	The channel controlling the coat color effect of the shader.	Has no perceivable effect in the Mari viewport.
Coat Roughness	dropdown	The channel controlling the coat roughness effect of the shader.	Has no perceivable effect in the Mari viewport.
Coat Specular Level	dropdown	The channel controlling the coat specular level effect of the shader.	Has no perceivable effect in the Mari viewport.
Subsurface Weight	dropdown	The channel controlling the subsurface weight effect of the shader.	Has no perceivable effect in the Mari viewport.
Subsurface Color	dropdown	The channel controlling the subsurface color effect of the shader.	Has no perceivable effect in the Mari viewport.
Subsurface Scale	dropdown	The channel controlling the subsurface scale effect of the shader.	Has no perceivable effect in the Mari viewport.
Incandescence Color	dropdown	The channel controlling the incandescence color effect of the shader.	
Incandescence Intensity	dropdown	The channel controlling the incandescence intensity effect of the shader.	
Normal	dropdown	The channel controlling the normal effect of the shader.	
Bump	dropdown	The channel controlling the bump effect of the shader.	
Normal Intensity	dropdown	The channel controlling the normal intensity effect of the shader.	

Vector	dropdown	The channel controlling the vector effect of the shader.	See Vector Brush for more information.
Displacement	dropdown	The channel controlling the displacement effect of the shader.	

Shader

Control	Type	What it does	Notes
R	button	Resets the control back to its default value.	
Base Layer | **Color**	swatch	Adjusts the color of the base layer.	The toggle picks whether this input is read in color or scalar data.
Base Layer | **Roughness**	text field, slider	Controls the roughness of the base layer.	Higher values will result in a rougher surface look more suitable for materials like concrete, plaster, or sand.
Base Layer | **Specular Level**	text field, slider	The specular level influences the strength of the specular reflections on the base layer.	
Base Layer | **Metallic**	text field, slider	Mainly controls the color of the reflections. Reflection from metals has the color of the light, whilst non-metals reflect the color of the surface.	
Base Layer | **Anisotropy**	text field, slider	Anisotropy reflects and transmits light with a directional bias and causes materials to appear rougher or glossier in certain directions. The default value for Anisotropy is 0, which means isotropic. As you move the control towards 1.0, the surface is made more anisotropic along the	

		anisotropic direction.	
Base Layer | **Anisotropy Direction**	text field, slider	Specifies the anisotropy direction in local tangents space.	
Base Layer | **Opacity**	text field, slider	Controls the transparency of the surface, 1.000 is fully opaque and 0.000 is fully transparent.	
Coating Layer | **Thickness**	text field, slider	The thickness of the coating layer in scene units (world units). The thicker the layer, the more light is absorbed and less visibility of the base layer.	
Coating Layer | **Color**	swatch	The color of the coating.	The color is more prominent in thicker coating layers.
Coating Layer | **Roughness**	text field, slider	The roughness of the coating layer. The smaller the value, the smoother the surface.	
Coating Layer | **Specular Level**	text field, slider	The specularity of the coating layer. The lower the value, the less specularity the coating has.	
Subsurface | **Weight**	text field, slider	The 'blend' between diffuse and subsurface scattering. When set to 1.0, there is only SSS, and when set to 0 it is only Lambert. In most cases, you want this to be 1.0 (full SSS).	
Subsurface | **Color**	swatch	Changes the color of the subsurface. For example, replicating a skin material would mean setting this to a fleshy color.	
Subsurface | **Scale**	text field, slider	A global scale that applies to the subsurface of this shader. Larger values make the object appear more translucent. Lower values decrease the subsurface scattering effect and make the surface appear more solid.	
Incandescence | **Color**	swatch	The color of the emitted light.	
Incandescence |	text field,	Intensity of the emitted light.	Final color of the

Intensity	slider		incandescence is color * intensity.
Normal | **Layers Affected**	dropdown	Choose whether to apply the Normal effect to the **Coating**, **Base**, or **Both**.	
Normal | **Intensity**	text field, slider	Specifies the intensity of the Normal effect. Lower values decrease the bump effect, and higher values intensify the bump effect.	
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.	
Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shader. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.	
Displacement | **Displacement Range**	text field, slider	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.	
Displacement | **Max Tessellation**	text field, slider	The maximum amount of texels the surface is tessellated to.	
Displacement | **Perturb Normals**	dropdown	Selecting Yes changes the displacement so that it moves the surface but leaves the surface normals as they are.	
Bump | **Bump Weight**	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.	
Bump | **Bump Mode**	dropdown	Displays bump quickly (**Fast**), accurately (**Accurate**), or less accurately (**Fastest**).	
Bump | **Bump Space**	dropdown	• **UV** - Normals are calculated relative to the UV coordinates space. The **UV** Bump Space mode is the more traditional approach but can lead to noticeable seams and	

		color variations. • **World** - Normals are calculated relative to the World coordinates space. Using the **World** Bump Space mode gives a more seamless result.		

BRDF

Control	Type	What it does	Dialog	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.		Displacement is toggled off by default.
Inputs | **Diffuse Color**	dropdown	The channel controlling the diffuse color effect of the shaded surface.		
Inputs | **Specular Color**	dropdown	The channel controlling the specular color effect of the shaded surface.		
Inputs | **Glossiness**	dropdown	The channel controlling the glossy areas for highlights in the specular shaded surface.		
	button	Inverts the channel values of the selected channel.		
Inputs | **Reflectance**	dropdown	The channel controlling the reflective effect of the shaded surface.		
Inputs | **Ambient Occlusion**	dropdown	The channel controlling the ambient occlusion effect of the shaded surface. Using this input overrides the ambient occlusion		

Control	Type	What it does	Dialog	Notes
		values stored on the object (**Objects** > **Ambient Occlusion**).		
Inputs | **Emissive Color**	dropdown	The channel controlling the color of the emissive (glow) qualities on the shaded surface.		
Inputs | **Normal**	dropdown	The channel used for input as a representation of the surface normals, added to the shaded surface. You must select either a **Bump** or **Normal** shader component in the same shader. If you attempt to use both, **Normal** overrides the **Bump** shader component.		
Inputs | **Bump**	dropdown	The channel controlling the bump map to show as a perturbed lighting.		
Inputs | **Vector**	dropdown	The channel controlling the vector field and resulting vector data from paint on the shaded surface.		This data is used by the vector inspector.
Inputs | **Displacement**	dropdown	The channel controlling the displacement map and dynamic tessellation to show a more advanced preview of displacement.		This component is disabled for Ptex channels.
	button	Adds a new channel for the shader component.	Add Channel Dialog	
Diffuse	text field, slider	How much of the diffuse input appears on the shaded surface.		From 0 to 1; defaults to 1.
Specular	text field, slider	How much of the specular light appears on the shaded surface.		From 0 to 1; defaults to 1.
Glossiness	text field,	How glossy the highlights appear		From 0 to 1;

Control	Type	What it does	Dialog	Notes
	slider	on the shaded surface. The closer to 1 this is set, the more mirror-like the shaded surface appears.		defaults to .200.
Ambient Occlusion	text field, slider	How much ambient occlusion occurs on the shaded surface.		From 0 to 2; defaults to 1.
Reflectance	text field, slider	How much of the reflection appears on the shaded surface. The closer to 1 this is set, the higher the reflectiveness.		From 0 to 1; defaults to .200.
Emissive	text field, slider	How much glow the emissive channel appears to have.		From 0 to 100; defaults to 1.000.
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.		From 0 to 1; defaults to 0.500.
Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shaded surface. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.		From 0 to 1; defaults to 0.500.
Displacement | **Displacement Range**	text field	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.		
Displacement | **Max Tessellation**	text field, slider	How many texels the surface is tessellated to.		From 1 to 64; defaults to 10.
Displacement | **Perturb Normals**	dropdown	Selecting **yes** changes the displacement so that displacement moves the surface but leaves the surface normals as they are.		

Control	Type	What it does	Dialog	Notes
Bump | **Bump Weight**	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.		From 0 to 10; defaults to 1.000.
Bump| **Bump Mode**	dropdown	Displays bump quickly (**Fast**), accurately (**Accurate**), or less accurately (**Fastest**).		While **Fastest** is very fast, it works better with live procedural layers rather than painted or cached layers.
Bump |**Bump Space**	dropdown	**UV** - Normals are calculated relative to the UV coordinates space. The **UV Bump Space** mode is the more traditional approach but can lead to noticeable seams and color variations. **World** - Normals are calculated relative to the World coordinates space. Using the **World Bump Space** mode gives a more seamless result.		By default, **Bump Space** is set to **UV**.

VRayMtl

The VRayMtl is a versatile material that allows for better physically correct illumination (energy distribution) in the scene. This material can be easily set up to simulate a variety of surfaces from plastics, metals, and glass. Furthermore, you can define the BRDF to decide how you want light to interact with the surface material.

This page is based on the VRayMtl shader documentation, for more information see docs.chaosgroup.com/display/VRAY4MODO/V-Ray+Material.

Inputs

Control	Type	What it does	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.	Displacement is toggled off by default.
	button	Creates a new Channel.	
Inputs | **Diffuse Color**	dropdown	The channel controlling the diffuse color of the material.	The actual diffuse color of the surface might also be affected by the reflection and refraction colors.
Inputs | **Diffuse Amount**	dropdown	The channel controlling the diffuse amount of the material.	
Inputs | **Opacity Map**	dropdown	The channel containing the opacity map (to determine which parts of the material are opaque or translucent) for the material.	
Inputs | **Roughness Amount**	dropdown	The channel specifying the roughness amount on the surface.	
Inputs | **Self-Illumination**	dropdown	The channel specifying the areas on the material that are self-illuminated.	
Inputs | **Reflection Color**	dropdown	The channel controlling the color on the areas of the material where light is reflected.	
Inputs | **Reflection Amount**	dropdown	The channel controlling the reflection amount on the material.	
Inputs | **Reflection**	dropdown	The channel specifying the areas on the	

Control	Type	What it does	Notes
Glossiness		material where there is glossiness on a highlight.	
Inputs | **Fresnel IOR**	dropdown	The channel controlling the Fresnel IOR of the material.	
Inputs | **GGX Tail Falloff**	dropdown	The channel controlling the GGX Tail Falloff of the material.	
Inputs | **Metalness**	dropdown	The channel controlling the Metalness of the material.	
Inputs | **Anisotropy**	dropdown	The channel specifying the direction of the anisotropic rotation.	
Inputs | **Anisotropy Rotation**	dropdown	The channel controlling the color of the areas on the material where light is refracted.	
Inputs | **Refraction Color**	dropdown	The channel specifying the areas on the material where there is glossiness on a refractive surface.	
Inputs | **Refraction Amount**	dropdown	The channel controlling the color of the areas on the material where fog (the attenuation of light as it passes through a material) appears, so that thick objects look less transparent than thinner objects.	
Inputs | **Refraction Glossiness**	dropdown	The channel controlling the color of the areas of translucency (or sub-surface scattering) in a material.	
Inputs | **Refraction IOR**	dropdown	The channel containing the bump map.	
Inputs | **Fog Color**	dropdown	The channel controlling the fog color of the material.	
Inputs | **Translucency**	dropdown	The channel controlling the translucency color of the material.	

Control	Type	What it does	Notes
Color			
Inputs | **Normal**	dropdown	The channel controlling the normal map of the material.	
Inputs | **Bump**	dropdown	The channel controlling the bump map of the material.	
I	button	Inverts the channel values of the selected **Bump** channel.	
Inputs | **Vector**	dropdown	The channel controlling the vector map of the material.	
Inputs | **Displacement**	dropdown	The channel controlling the displacement map of the material.	

Shader

Control	Type	What it does	Notes
Basic Properties | **Diffuse Color**	swatch	The base diffuse color on the object for the **Diffuse Color** channel.	
Basic Properties | **Diffuse Amount**	text field, slider	The base diffuse color amount on the object for the **Diffuse Amount** channel.	
Basic Properties | **Opacity Map**	swatch	Sets the color for the **Opacity Map** channel.	
Basic Properties | **Roughness Amount**	text field, slider	The level of roughness of the diffuse component in the **Diffuse Roughness** channel. Higher values result in a rougher surface.	
Basic Properties | **Self-Illumination**	swatch	The self illumination color of the material.	

Reflection | **Reflection Color**	swatch	The reflection color.	The reflection color dims the diffuse surface color.
Reflection | **Reflection Amount**	text field, slider	Controls the objects overall reflectivity.	The more reflective the material becomes, the less diffuse color it emits.
Reflection | **Reflection Glossiness**	text field, slider	Controls the sharpness of reflections. A value of 1.0 means mirror-like reflection, and lower values produce blurry or glossy reflections.	
Reflection | **Metalness**	text field, slider	Controls the metalness of the material. A value of 0 means the shader doesn't look metallic at all, a value of 1 means the material takes on a metal appearance.	
Reflection | **GGX Tail Falloff**	text field, slider	Defines the transition from highlighted areas to non-highlighted areas when **BRDF Type** is set to **GGX**.	
Reflection | **Use Fresnel**	toggle	When toggled on, makes the reflection strength dependent on the viewing angle of the surface.	Fresnel also depends on the **Fresnel IOR** value.
Reflection | **Glossy Fresnel**	toggle	When toggled on, uses glossy Fresnel to interpolate glossy reflections and refractions.	This control is permanently locked, since the calculations involved result in a more realistic appearance for edges with lower glossiness.
Reflection | **Lock Fresnel IOR to Refract...**	toggle	Locks the Fresnel effect to the **Refraction IOR** value.	Unlock this for finer control when needed.
Reflection | **Fresnel IOR**	text field, slider	The Index of Refraction used when calculating Fresnel reflections.	Normally this is locked to **Refraction IOR** parameter, but when unlocked provides you

			with finer control.
Reflection | **Use Roughness**	toggle	Determines whether **Roughness Amount** is used in the reflection of the material.	
Reflection | **BRDF Type**	dropdown	BRDF is an equation that defines how light is reflected off a surface. The BRDF Type determines the highlight's general shape: • **Phong** - Phong highlights/reflections. • **Blinn** - Blinn highlights/reflections. • **Ward** - Ward highlights/reflections. • **GGX** - GGX highlights/reflections.	GGX is the most modern and flexible BRDF type, and is able to represent a broader range of materials due to its ability to control the shape of the specular lobe.
Reflection | **Trace Reflections**	toggle	When disabled, reflections are not traced even if **Reflection Color** is other than black.	You can disable this option to produce only highlights. When this option is disabled, the **Diffuse Color** is not dimmed by the **Reflection Color**, as would normally happen.
Anisotropy | **Anisotropy**	text field, slider	Determines the roundness of the highlights. A value of 0.0 means isotropic (round) highlights, while other values elongate the highlight.	Negative and positive values simulate brushed metal surfaces.
Anisotropy | **Anisotropy Rotation**	text field, slider	Determines the orientation of the anisotropic effect. A value of 0 is 0 degrees, and a value of 1.000 is 360 degrees.	
Anisotropy | **Local Object Axis**	dropdown	Determines the direction in the local object axis of the anisotropic effect.	
Refraction |	swatch	Determines the refraction color.	Refraction color depends

Refraction Color			on the **Reflection Color** as well.
Refraction | **Refraction Amount**	text field, slider	The amount of refraction in the material.	
Refraction | **Refraction Glossiness**	text field, slider	Controls the sharpness of the refractions. A value of 1.0 means glass-like refraction, and lower values produce blurry or glossy refractions.	
Refraction | **Refraction IOR**	text field, slider	The Index of Refraction for the material, which describes the way light bends when crossing the material surface. A value of 1.0 means the light will not change direction.	
Refraction | **Trace Refractions**	toggle	Enables refractions for the material.	
Refraction | **Fog Color**	swatch	The attenuation of light as it passes through the material. It simulates the fact that thick objects look less transparent than thin objects.	The effect of the fog color depends on the absolute size of the objects and is therefore scene-dependent.
Refraction | **Fog Multiplier**	text field, slider	The strength of the fog effect. Smaller values make the material appear more transparent, larger values make the material appear more opaque.	
Refraction | **Fog Bias**	text field, slider	Changes the way the **Fog Color** is applied. Negative values make the thin parts of the object more transparent, and thicker parts more opaque. Positive values make the thin parts more opaque, and the thicker parts more transparent.	

Subsurface Scattering | **SSS On**	toggle	Turns subsurface scattering on or off.	Has no perceivable effect in the Mari viewport.
Subsurface Scattering | **Translucency Color**	swatch	Normally the color of the sub-surface scattering effect depends on the **Fog Color**. The **Translucency Color** adds tint to the SSS effect.	Has no perceivable effect in the Mari viewport.
Subsurface Scattering | **Fwd/back Coefficient**	text field, slider	Controls the scatter direction of a ray. A value of 0.0 means a ray moves away from the surface. A value of 1.0 means a ray will move towards the surface.	Has no perceivable effect in the Mari viewport.
Subsurface Scattering | **Scatter Coefficient**	text field, slider	Defines the amount of scattering inside the object. A value of 0.0 means rays are scattered in all directions. A value of 1.0 means a ray cannot change its direction inside the sub-surface volume.	Has no perceivable effect in the Mari viewport.
Subsurface Scattering | **Thickness**	text field, slider	Limits the rays that are traced below the surface.	This option is useful if you don't need to trace the whole sub-surface volume. Has no perceivable effect in the Mari viewport.
Misc | **Double Sided**	toggle	When enabled, VRay flips the normals for back-facing surfaces with this material applied.	
Misc | **Approximate Environment**	toggle		
Misc | **Environment Samples**	text field, slider		
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.	

Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shader. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.	
Displacement | **Displacement Range**	text field, slider	What the range of displacement is. This setting is multiplied by the Displacement Scale to give the displacement.	
Displacement | **Max Tessellation**	text field, slider	The maximum amount of texels the surface is tessellated to.	
Displacement | **Perturb Normals**	dropdown	Selecting Yes changes the displacement so that it moves the surface but leaves the surface normals as they are.	
Bump | **Bump Weight**	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.	
Bump | **Bump Mode**	dropdown	Displays bump quickly (Fast), accurately (Accurate), or less accurately (Fastest).	
Bump | **Bump Space**	dropdown	• **UV** - Normals are calculated relative to the UV coordinates space. The **UV** Bump Space mode is the more traditional approach but can lead to noticeable seams and color variations. • **World** - Normals are calculated relative to the World coordinates space. Using the **World** Bump Space mode gives a more seamless result.	

Principled BRDF

Control	Type	What it does	Dialog	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.		Displacement is toggled off by default.
Inputs | **Base Color**	dropdown	The surface color, usually supplied by texture maps.		
Inputs | **Metallic**	dropdown	The metallic-ness (**0** = dielectric, **1** = metallic). This is a linear blend between two different models. The metallic model has no diffuse component and also has a tinted incident specular, equal to the base color.		
Inputs | **Subsurface**	dropdown	Controls diffuse shape using a subsurface approximation.		
Inputs | **Specular**	dropdown	Incident specular amount. This is in lieu of an explicit index-of-refraction.		
Inputs | **Roughness**	dropdown	Surface roughness, controls both diffuse and specular response.		
Inputs | **Specular Tint**	dropdown	A concession for artistic control that tints incident specular towards the base color. Grazing specular is still achromatic.		
Inputs | **Anisotropic**	dropdown	Degree of anisotropy. This controls the aspect ratio of the specular highlight. (**0** = isotropic, **1** = maximally anisotropic).		

Control	Type	What it does	Dialog	Notes
Inputs | **Sheen**	dropdown	An additional grazing component, primarily intended for cloth.		
Inputs | **Sheen Tint**	dropdown	Amount to tint sheen towards base color.		
Inputs | **Clearcoat**	dropdown	A second, special-purpose specular lobe.		This data is used by the vector inspector.
Inputs | **Clearcoat Gloss**	dropdown	Controls clearcoat glossiness (**0** = a "satin" appearance, **1** = a "gloss" appearance).		This component is disabled for Ptex channels.
Inputs | **Ambient Occlusion**	dropdown	The channel controlling the ambient occlusion effect of the shaded surface. Using this input overrides the ambient occlusion values stored on the object (**Objects** > **Ambient Occlusion**).		
Inputs | **Emissive Color**	dropdown	The channel controlling the color of the emissive (glow) qualities on the shaded surface.		
Inputs | **Normal**	dropdown	The channel used for input as a representation of the surface normals, added to the shaded surface. You must select either a **Bump** or **Normal** shader component in the same shader. If you attempt to use both, **Normal** overrides the **Bump** shader component.		
Inputs | **Bump**	dropdown	The channel controlling the bump map to show as a perturbed lighting.		

Control	Type	What it does	Dialog	Notes
Inputs | **Vector**	dropdown	The channel controlling the vector field and resulting vector data from paint on the shaded surface.		
Inputs | **Displacement**	dropdown	The channel controlling the displacement map and dynamic tessellation to show a more advanced preview of displacement.		
	button	Adds a new channel for the shader component.	Add Channel Dialog	
Metallic	text field, slider	The metallic-ness (**0** = dielectric, **1** = metallic). This is a linear blend between two different models. The metallic model has no diffuse component and also has a tinted incident specular, equal to the base color.		
Subsurface	text field, slider	Controls diffuse shape using a subsurface approximation.		From 0 to 1; defaults to 1.
Specular	text field, slider	Incident specular amount. This is in lieu of an explicit index-of-refraction.		From 0 to 1; defaults to 1.
Roughness	text field, slider	Surface roughness, controls both diffuse and specular response.		From 0 to 1; defaults to .200.
Specular Tint	text field, slider	A concession for artistic control that tints incident specular towards the base color. Grazing specular is still achromatic.		From 0 to 2; defaults to 1.
Anisotropic	text field, slider	Degree of anisotropy. This controls the aspect ratio of the specular highlight. (**0** = isotropic, **1** = maximally anisotropic).		From 0 to 1; defaults to .200.

Control	Type	What it does	Dialog	Notes
Sheen	text field, slider	An additional grazing component, primarily intended for cloth.		From 0 to 100; defaults to 1.000.
Sheen Tint	text field, slider	Amount to tint sheen towards base color.		
Clearcoat	text field, slider	A second, special-purpose specular lobe.		
Clearcoat Gloss	text field, slider	Controls clearcoat glossiness (0 = a "satin" appearance, 1 = a "gloss" appearance).		
Ambient Occlusion	text field, slider	How much ambient occlusion occurs on the shaded surface.		
Emissive	text field, slider	How much glow the emissive channel appears to have.		
IBL Quality	dropdown	Performs image-based lighting on the surface of the object. The lower the setting the lower the visible quality but the faster the frame rate. The higher the setting the better the rendered quality, look, and accuracy but the lower the frame rate.		
Bump **| Bump Weight**	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.		From 0 to 10; defaults to 1.000.
Bump **| Bump Mode**	dropdown	Displays bump quickly (**Fast**), accurately (**Accurate**), or less accurately (**Fastest**).		While **Fastest** is very fast, it works better with live procedural layers rather than painted or cached layers.

Control	Type	What it does	Dialog	Notes
Bump **|Bump Space**	dropdown	**UV** - Normals are calculated relative to the UV coordinates space. The **UV Bump Space** mode is the more traditional approach but can lead to noticeable seams and color variations. **World** - Normals are calculated relative to the World coordinates space. Using the **World Bump Space** mode gives a more seamless result.		By default, **Bump Space** is set to **UV**.
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.		From 0 to 1; defaults to 0.500.
Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shaded surface. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.		From 0 to 1; defaults to 0.500.
Displacement **|Displacement Range**	text field	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.		
Displacement | **Max Tessellation**	text field, slider	How many texels the surface is tessellated to.		From 1 to 64; defaults to 10.
Displacement **|Perturb Normals**	dropdown	Selecting **yes** changes the displacement so that displacement moves the surface but leaves the surface normals as they are.		

Arnold Standard Surface

Note: The non-active inputs (highlighted in gray) help drive the shaders but have no effects on the canvas. This page is based on the Arnold Standard Surface shader documentation, for more information, see https://docs.arnoldrenderer.com/display/A5AFMUG/Standard+Surface.

Control	Type	What it does	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.	Displacement Map is toggled off by default.
Inputs | **Diffuse Weight**	dropdown	The channel controlling the diffuse weight effect of the shaded surface.	Default is 0.8
Inputs | **Diffuse Color**	dropdown	The channel controlling the diffuse color effect of the shaded surface.	
Inputs | **Diffuse Roughness**	dropdown	The channel controlling the diffuse roughness effect of the shaded surface.	
Inputs | **Metalness**	dropdown	The channel controlling the metalness effect of the shaded surface.	
Inputs | **Specular Weight**	dropdown	The channel controlling the specular weight effect of the shaded surface.	
Inputs | **Specular Color**	dropdown	The channel controlling the specular color effect of the shaded surface.	
Inputs | **Specular Roughness**	dropdown	The channel controlling the specular roughness effect of the shaded surface.	
Inputs | **Specular IOR**	dropdown	The channel controlling the specular IOR effect of the shaded surface.	

Control	Type	What it does	Notes
Inputs | **Anisotropy**	dropdown	The channel controlling the anisotropy effect of the shaded surface.	
Inputs | **Anisotropy Rotation**	dropdown	The channel controlling the anisotropy rotation effect of the shaded surface.	
Inputs | **Transmission Weight**	dropdown	The channel controlling the transmission weight effect of the shaded surface.	
Inputs | **Transmission Color**	dropdown	The channel controlling the transmission color effect of the shaded surface.	
Inputs | **Transmission Depth**	dropdown		
Inputs | **Transmission Scatter**	dropdown		
Inputs | **Transmission Scatter Anisotropy**	dropdown		
Inputs | **Transmission Dispersion**	dropdown		
Inputs | **Transmission Extra Roughness**	dropdown		
Inputs | **Subsurface Weight**	dropdown	The channel controlling the subsurface weight effect of the shaded surface.	

Control	Type	What it does	Notes
Inputs | **Subsurface Color**	dropdown	The channel controlling the subsurface color effect of the shaded surface.	
Inputs | **Subsurface Radius**	dropdown	The channel controlling the subsurface radius effect of the shaded surface.	
Inputs | **Subsurface Scale**	dropdown	The channel controlling the subsurface scale effect of the shaded surface.	
Inputs | **Clearcoat Weight**	dropdown	The channel controlling the clearcoat weight effect of the shaded surface.	
Inputs | **Clearcoat Color**	dropdown	The channel controlling the clearcoat color effect of the shaded surface.	
Inputs | **Clearcoat Roughness**	dropdown	The channel controlling the clearcoat roughness effect of the shaded surface.	
Inputs | **Clearcoat IOR**	dropdown	The channel controlling the clearcoat IOR effect of the shaded surface.	
Inputs | **Sheen Weight**	dropdown		
Inputs | **Sheen Color**	dropdown		
Inputs | **Sheen Roughness**	dropdown		
Inputs | **Thin Film Thickness**	dropdown		
Inputs | **Thin Film IOR**	dropdown		

Control	Type	What it does	Notes
Inputs | **Emission Weight**	dropdown	The channel controlling the emission weight effect of the shaded surface.	
Inputs | **Emission Color**	dropdown	The channel controlling the emission color of the shaded surface.	
Inputs | **Opacity**	dropdown	The channel controlling the opacity effect of the shaded surface.	
Inputs | **Normal Map**	dropdown	The channel controlling the normal map effect of the shaded surface.	
Inputs | **Bump Map**	dropdown	The channel controlling the bump map effect of the shaded surface.	
Inputs | **Vector Map**	dropdown	The channel controlling the vector map effect of the shaded surface.	
Inputs | **Displacement Map**	dropdown	The channel controlling the displacement map effect of the shaded surface.	Displacement Map is toggled off by default.
	button	Adds a new channel for the shader component.	Add Channel Dialog

Base

Control	Type	What it does
Weight	text field, slider	The base color weight.
Color	swatch, Color/Scalar icon	The base color sets how bright the surface is when lit directly with a white light source (intensity at 100%). It defines which percentage for each component of the RGB spectrum that does not get absorbed when light scatters beneath the surface. Metal normally has a black or very dark base color, however, rusty metals need some base color. A base color map is

Control	Type	What it does
		usually required.
Roughness	text field, slider	The base component follows an Oren-Nayar reflection model with surface roughness. A value of 0.0 is comparable to a Lambert reflection. Higher values will result in a rougher surface look more suitable for materials like concrete, plaster, or sand.
Metalness	text field, slider	With metalness 1.0 the surface behaves like a metal, using fully specular reflection and complex Fresnel.

Specular

Control	Type	What it does
Weight	text field, slider	The specular weight influences the brightness of the specular highlight.
Color	swatch, /Color/Scalar icon	The color the specular reflection is modulated with. Use this color to 'tint' the specular highlight. You should only use colored specular for certain metals, whereas non-metallic surfaces usually have a monochromatic specular color. Non-metallic surfaces normally do not have a colored specular.
Roughness	text field, slider	Controls the glossiness of the specular reflections. The lower the value, the sharper the reflection. A value of 0 gives you a perfectly sharp mirror reflection, while 1.0 creates reflections that are close to a diffuse reflection. You should connect a map here to get variation in the specular highlight.
IOR	text field	The IOR parameter (Index of Refraction) defines the material's Fresnel reflectivity and is by default the angular function used. Effectively the IOR defines the balance between reflections on surfaces facing the viewer and on surface edges. You can see the reflection intensity remains unchanged, but the reflection intensity on the front side changes a lot.
Anisotropy	text field, slider	Anisotropy reflects and transmits light with a directional bias and causes materials to appear rougher or glossier in certain directions. The default value for Anisotropy is 0, which means 'isotropic.' As you move the control towards 1.0, the surface is made more anisotropic in the U axis.

Control	Type	What it does
Anisotropy Rotation	text field, slider	The rotation value changes the orientation of the anisotropic reflectance in UV space. At 0.0, there is no rotation, while at 1.0 the effect is rotated by 180 degrees. For a surface with brushed metal, this controls the angle at which the material was brushed. For metallic surfaces, the anisotropic highlight should stretch out in a direction perpendicular to the brushing direction.

Transmission

Control	Type	What it does
Weight	text field, slider	Allows light to scatter through the surface, for materials such as glass or water.
Color	swatch, Color/Scalar icon	This filters the refraction according to the distance traveled by the refracted ray. The longer light travels inside a mesh, the more it is affected by the Transmission Color. Therefore green glass gets a deeper green as rays travel through thicker parts. The effect is exponential and computed with Beer's Law. It is recommended to use light, subtle color values.
Depth	text field	Controls the depth into the volume at which the transmission color is realized. Increasing this value makes the volume thinner, which means less absorption and scattering. It is a scale factor so that you can set a transmission color and then tweak the Depth to be appropriate for the size of your object.
Scatter	swatch, Color/Scalar icon	Transmission Scatter is suitable for any liquid that is fairly thick or where there is enough of it for scattering to be visible, such as a deep body of water or honey. If you have a glass of water, there is not that much scattering, however, for an ocean, it is required. Other examples include materials like ice, opalescent glass or milky glass.
Scatter Anisotropy	text field, slider	The directional bias, or anisotropy, of the scattering. The default value of zero gives isotropic scattering so that light is scattered evenly in all directions. Positive values bias the scattering effect forwards, in the direction of the light, while negative values bias the scattering backward, toward the light.

Control	Type	What it does
Dispersion	text field, slider	Specifies the Abbe number of the material, which describes how much the index of refraction varies across wavelengths. For glass and diamonds, this is typically in the range of 10 to 70, with lower numbers giving more dispersion. The default value is 0, which turns off dispersion.
Extra Roughness	text field, slider	Adds some additional blurriness of a refraction computed with an isotropic microfacet BTDF (Bidirectional Transmittance Distribution Function). The range goes from 0 (no roughness) to 1.

Subsurface

Control	Type	What it does
Weight	text field, slider	The 'blend' between diffuse and subsurface scattering. When set to 1.0, there is only SSS, and when set to 0 it is only Lambert. In most cases, you want this to be 1.0 (full SSS).
Color	swatch, /Color/Scalar icon	The color used to determine the subsurface scattering effect. For example, replicating a skin material would mean setting this to a fleshy color.
Radius	swatch, /Color/Scalar icon	The approximate distance up to which light can scatter below the surface, also known as “mean free path” (MFP). This parameter affects the average distance that light might propagate below the surface before scattering back out. This effect on the distance can be specified for each color component separately. Higher values smooth the appearance of the subsurface scattering, while lower values result in a more opaque look.
Scale	text field	Controls the distance that the light is likely to travel under the surface before reflecting back out. It scales the scattering **Radius** and multiplies the SSS Radius **Color**.

Clearcoat

Control	Type	What it does
Weight	text field, slider	This attribute is used to coat the material. It acts as a clear-coat layer on top of all other shading effects. The coating is always reflective (with the given roughness) and is assumed to be dielectric, such as the clear-coat layer for car paint or the sheen layer for a skin material. For example, for an extra oily layer or wet skin. Other examples include objects that have been laminated or a protective film over an aluminum cell phone.
Color	swatch, /Color/Scalar icon	This is the color of the coating layer's transparency.
Roughness	text field, slider	Controls the glossiness of the specular reflections. The lower the value, the sharper the reflection. In the limit, a value of 0 gives you a perfectly sharp mirror reflection, while 1.0 creates reflections that are close to a diffuse reflection. You should connect a map here to get variation in the coat highlight.
IOR	text field	The IOR parameter (Index of Refraction) defines the material's Fresnel reflectivity and is by default the angular function used. Effectively the IOR defines the balance between reflections on surfaces facing the viewer and on surface edges. You can see the reflection intensity remains unchanged, but the reflection intensity on the front side changes a lot.

Sheen

Control	Type
Weight	text field, slider
Color	swatch, /Color/Scalar icon
Roughness	text field, slider

Thin Film

Control	Type
Thickness	text field, slider
IOR	text field

Emission

Control	Type	What it does
Weight	text field, slider	Controls the amount of emitted light. It can create noise, especially if the source of indirect illumination is very small (for example, light bulb geometry).
Color	swatch, Color/Scalar icon	The emitted light color.

Geometry

Control	Type	What it does
Thin Walled	checkbox	When enabled, provides the effect of a translucent object being lit from behind (the shading point is 'lit' by the specified fraction of the light hitting the reverse of the object at that point). It is recommended that this only be used with thin objects (single sided geometry) as objects with thickness may render incorrectly.
Opacity	swatch, Color/Scalar icon	Controls the transparency of the surface, white is fully opaque and black is fully transparent.

Advanced

Control	Type	What it does
Caustics	checkbox	
Internal Reflections	checkbox	
Exit to Background	checkbox	When enabled, this causes the Standard Surface shader to trace a ray against the background/environment when the maximum GI reflection/refraction depth is met and return the color that is visible in the background/environment in that direction. When the option is disabled, the path is terminated instead and returns black when the maximum depth is reached.
Indirect Diffuse Weight	text field, slider	
Indirect Specular Weight	text field, slider	

Displacement

Control	Type	What it does	Notes
Displacement			
Displacement Bias	text field, slider	How much the white or black values are pushed or pulled from the surface.	Ranges from 0 to 1; defaults to 0.500.
Displacement Scale	text field, slider	How much displacement is applied to the shaded surface. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.	Ranges from 0 to 1; defaults to

Control	Type	What it does	Notes
			0.500.
Displacement Range	text field	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.	
Max Tesselation	text field, slider	How many texels the surface is tessellated to.	Ranges from 1 to 64; defaults to 10.
Perturb Normals	dropdown	Selecting **Yes** changes the displacement so that it moves the surface but leaves the surface normals as they are.	

Bump

Control	Type	What it does	Notes
Bump			
Bump Weight	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.	Ranges from 0 to 10; defaults to 1.000.
Bump Mode	dropdown	Displays bump quickly (**Fast**), accurately (**Accurate**), or less accurately (**Fastest**).	While **Fastest** is very fast, it works better with live procedural layers rather than painted or cached layers.
Bump Space	dropdown	• **UV** - Normals are calculated relative to the UV coordinates space. The **UV** Bump Space mode is the more traditional approach but can lead to noticeable seams and color variations. • **World** - Normals are calculated relative to the World coordinates space. Using the **World** Bump Space mode gives a more seamless result.	By default, **Bump Space** is set to **UV**.

Unreal

Note: Results may not be as expected if you do not use the **Environment Light** in the **Lights** palette when using this shader.

Control	Type	What it does	Dialog	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.		Displacement is toggled off by default.
Inputs | **Base Color**	dropdown	The channel controlling the overall diffuse color of the material.		
Inputs | **Roughness**	dropdown	The channel controlling how rough the material is. Rough materials scatter reflected light in more directions than smooth materials.		
Inputs | **Metallic**	dropdown	The channel specifying the areas on the material that have a metallic effect, with results being determined by values between 0 and 1.		
Inputs | **Specular**	dropdown	The channel specifying the non-metallic areas on the material where the specularity is visible.		
Inputs | **Ambient Occlusion**	dropdown	The channel used to specify the self-shadowing that occurs on an occluded object.		
Inputs | **Normal**	dropdown	The channel containing the Normal map, used to give additional		

Control	Type	What it does	Dialog	Notes
		physical details to a surface.		
Inputs | **Bump**	dropdown	The channel containing the bump map to show as a perturbed lighting.		
	button	Inverts the channel values of the selected channel.		
Inputs | **Emissive Color**	dropdown	The channel controlling the color of the emissive (glow) qualities on the shaded surface.		
Inputs | **Vector**	dropdown	The channel controlling the vector field and resulting vector data from paint on the shaded surface.		This data is used by the vector inspector.
Inputs | **Displacement**	dropdown	The channel containing the displacement map and dynamic tessellation to show a more advanced preview of displacement.		This component is disabled for Ptex channels.
	button	Adds a new channel for the shader component.	Add Channel Dialog	
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.		From 0.000 to 1.000; defaults to 0.500.
Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shaded surface. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.		From 0.000 to 1.000; defaults to 0.500.
Displacement | **Displacement Range**	text field	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.		

Control	Type	What it does	Dialog	Notes
Displacement | **Max Tessellation**	text field, slider	How many texels the surface is tessellated to.		From 1.000 to 64.000; defaults to 10.000.
Displacement | **Perturb Normals**	dropdown	Selecting **yes** changes the displacement so that displacement moves the surface but leaves the surface normals as they are.		
Bump | **Bump Weight**	text field, slider	How much weight the bump map has. Lower values are smaller bumps, higher values are larger and more obvious displacements.		From 0.000 to 10.000; defaults to 0.100.
Bump| **Bump Mode**	dropdown	Displays bump quickly (**Fast**), accurately (**Accurate**), or less accurately (**Fastest**).		By default, Bump Mode is set to **Fast**. While **Fastest** is very fast, it works better with live procedural layers rather than painted or cached layers.
Bump |**Bump Space**	dropdown	**UV** - Normals are calculated relative to the UV coordinates space. The **UV Bump Space** mode is the more traditional approach but can lead to noticeable seams and color variations. **World** - Normals are calculated relative to the World coordinates space. Using the **World Bump Space** mode gives a more seamless result.		By default, **Bump Space** is set to **UV**.

Layered

Control	Type	What it does	Dialog	Notes
	toggle	Toggles the influence of channels connected to the shader without disconnecting them from the shader input.		Displacement is toggled off by default.
Inputs |**Vector**	dropdown	The channel controlling the vector field and resulting vector data from paint on the shaded surface.		This data is used by the vector inspector.
Inputs| **Displacement**	dropdown	The channel controlling the displacement map and dynamic tessellation to show a more advanced preview of displacement.		This component is disabled for Ptex channels.
	button	Adds a new channel for the shader component.	Add Channel Dialog	
Displacement | **Displacement Bias**	text field, slider	How much the white or black values are pushed or pulled from the surface.		From 0 to 1; defaults to 0.500.
Displacement | **Displacement Scale**	text field, slider	How much displacement is applied to the shaded surface. Lower values equal smaller displacement; higher values equal larger and more obvious displacement.		From 0 to 1; defaults to 0.500.
Displacement | **Displacement Range**	text field	What the range of displacement is. This setting is multiplied by the **Displacement Scale** to give the displacement.		
Displacement | **Max Tessellation**	text field, slider	How many texels the surface is tessellated to.		From 1 to 64; defaults to 10.

Control	Type	What it does	Dialog	Notes
Displacement | **Perturb Normals**	dropdown	Selecting **yes** changes the displacement so that displacement moves the surface but leaves the surface normals as they are.		

Shader Layers Tab Fields

Control	Type	What it does	Dialog	Notes
Name	dropdown, switch	Filter bar function that gives the choices: **Name**, **Type**, **Attribute**, and **Color Tag**. Depending on which filter option you select, the filter bar shows one of the following: • - filter by the type of layers by clicking on the associated icons. • - filter by the attributes of layers by clicking on the associated icons. • - filter by the color tags of the selected color icon.		Both **Type** and **Attribute** provide icons that allow you to set the parameters for your filter. Quickly clear the text field by pressing the icon.
Normal	dropdown	Click to select from the blend mode options available for individual layers. To adjust the blend amount, enter the value into the text field, or adjust the slider located to the right of the blend mode menu. • 1.00 R - adjusts the blend mode		This list does not include the advanced blend mode options.

Control	Type	What it does	Dialog	Notes
		amount.		
	button	Opens the **Advanced Blending** menu, which gives options for blend modes and components, as well as blend amount.		
	button	Adds a new shader to the layer stack from a dropdown menu list of: • **Unreal**, • **BRDF**, • **Principled BRDF**, • **3Delight**, • **Arnold Standard Surface**, • **VRayMtl**, and • **Choose Diffuse And Specular**.	Create Shader Dialog when selecting **Choose Diffuse and Specular**.	The **Choose Diffuse and Specular** option allows you to create a custom shader. Refer to the Create Shader Dialog for more information**.**
	button	Adds a mask to the selected layer or, if none is selected, the top layer in the layer stack.		If a mask has already been added to a layer, you cannot add a second mask using this method. You must create a mask stack using the right-click menu.
	button	Adds a mask stack to the selected layer, or if none is selected, the top layer in the layer stack.		Clicking on the mask stack icon opens the mask stack palette.
	button	Removes the shader from the layer stack.		Default shaders cannot be

Control	Type	What it does	Dialog	Notes
				removed from the **Shaders** palette.
	button	Assigns a color to the selected layers. To clear an already assigned color, click the icon and select the gray color in the center of the palette. This effectively removes the color and returns it to the default color state.		

Lights Palette

The **Lights** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Lights** palette displays the lights in your project, and allows you to:

- Customize the lights.
- Turn each light on or off.
- Move the lights around.
- Reposition a light to the current camera position.
- Render shadows for individual lights.

Lights Palette Fields

Control	Type	What it does	Notes
Sorted by	dropdown	Lets you specify whether to list lights by their	

Control	Type	What it does	Notes
		name or in a custom order. To use a custom order, drag the lights up and down the list.	
	button	Lets you reposition the selected light to the current camera position.	
Environment			
Color | **Intensity**	text field, slider	Sets the intensity (brightness) of the selected light.	0.010 to 100.000; defaults to 1.000.
General | **Fixed To**	dropdown	Sets what the light is fixed to: • **Scene** (default) - the light is fixed to the model, for example if a light is set at the model's back, it always points at the model's back, no matter how you move the view. • **Camera** - the light is fixed to a camera view, for example you can have a light always shining from above the camera, allowing you to move the model around and cast light on different areas.	
General | **On**	checkbox	Whether the light is on or off.	
Texture | **Up Axis**	dropdown	Sets the image being used for the environment light to either **Axis X**, **Axis Y** (default), or **Axis Z**. This adjusts both the background (canvas) image and the lighting on the model.	
Texture | **Rotation**	text field, slider	Rotates the background image and the lighting on the Y axis.	0.000 to 360.000 degree rotation; defaults to 0.000.
Texture | **Background**	dropdown	Toggles the visibility of the background (canvas)	

Control	Type	What it does	Notes
		image by selecting **Show** or **Hide**. This does not turning on or off the lighting, only the image on the canvas.	
Texture | **Resolution**	dropdown	Sets the resolution of the lighting to either **64**, **128**, or **256**. This does not affect the resolution of the background (canvas) image.	
Texture | **Cubemap Type**	dropdown	Sets how the cubemap is handled: either **None**, **LatLong**, or **Cross**. If a 2D image is being used for the environment light, this option specifies the method used for interpreting the image into a cubemap. When a 2D image is brought into the **Image** field, Mari attempts to set the **Cubemap Type** automatically, but this can be changed at any time.	
Texture | **Image**	arrows, button, file browser	Click the blank image space to choose an environment from the Mari Environment library. Clicking either of the arrow keys cycles backwards or forwards through the Mari Environment library, loading the image as it does. Clicking the file browser opens the **Open an Image** dialog. Select an image of any of the file formats listed in the **File of Type** field. You can also drag-and-drop an image from the **Image Manager** or a file browser onto the blank image space to load the image as an environment light.	For more about the options in the dialog, refer to Open an Image Dialog.
Texture | **Blur**	text field, slider	Blurs the background (canvas) image. If the image is a low resolution, it may not blur evenly.	0.000 to 1.000; defaults to 0.000.
Texture Animation | **Animation**	dropdown	Sets the background (canvas) image animation to either be **Static**, where it does not move, or **Rotate**, where it rotates around the object on the canvas at the speed set in the **Speed** field. If the	

Control	Type	What it does	Notes
		control is changed from **Rotate** to **Static**, the background (canvas) image is reset back to its default position.	
Texture Animation | **Speed**	text field, slider	Sets the rotation speed of the background (canvas) image. If the **Animation** field is set to **Static**, this control has no affect.	0.000 to 1.000; defaults to 0.050.
Light			
Color | **Specular**	swatch	Sets the specular color of the light (the "shiny" color of the light).	Defaults to white.
Color | **Diffuse**	swatch	Sets the diffuse color of the light (the "surface" color of the light).	Defaults to white.
Color | **Intensity**	text field, slider	Sets the intensity (brightness) of the selected light.	0.010 to 100; defaults to 1.000.
General | **Fixed To**	dropdown	Sets what the light is fixed to: • **Scene** (default) - the light is fixed to the model, for example if a light is set at the model's back, it always points at the model's back, no matter how you move the view. • **Camera** - the light is fixed to a camera view, for example you can have a light always shining from above the camera, allowing you to move the model around and cast light on different areas.	
General | **On**	checkbox	Whether the light is on or off.	
Position | **Position**	control sphere	Lets you move the light around your project. Click and drag to move the light.	
Rendering | **Render Shadows**	checkbox	Renders and displays the shadows cast on the model by the selected light in its current position.	

Projectors Palette

The **Projectors** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Projectors** palette displays the projectors in your project, and allows you to:

- Create, load, save, and delete projectors.
- Use a projector to take a snapshot of the current view.
- Use a projector to project an edited snapshot back onto the model's surface.

Projectors Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
	button	Creates a new projector.		
	button	Saves the selected projector to disk.	**Save Projector**	
	button	Loads a saved projector from disk.	**Load Projector**	
	button	Deletes the selected projector.		
<projector name> > *General* | **Name**	text field	Enter the name of the projector.		This identifies the projector at the top of the palette.
<projector name> > *Action* | **Frame Offset**	text field, slider	Set by how many frames to offset the loading of an animated object. You can		

Control	Type	What it does	Opens this dialog box	Notes
		specify a frame range, or simply a number of frames.		
<projector name> > *Action* | **Import**	button	Click to load the image from the file set in the Input File Path and project it as paint onto the model (without baking).	Pick Path (if **Input File Path** has not been specified)	
<projector name> > *Action* | **Project**	button	Click to load the image from the file set in the **Input File Path** and project and bake it onto the model.	Pick Path (if **Input File Path** has not been specified)	
<projector name> > *Action* | **Input File Path**	text field	Enter the filename and path where the projector loads updated images from.		
<projector name> > *Action* | **Input Alpha**	dropdown	Choose how to handle pre-multiplication of imported images by controlling how to process the alpha channel. • **Premultiply** - Premultiply the RGB values by the alpha channel of the imported image. • **Leave** - Leave the alpha channel as it is when the image is imported. • **Remove** - Remove the alpha channel from the imported image.		
<projector name> > *Action* | **Unproject**	button	Click to save the current view buffer out to the file set in the **Output File Path**.	Pick Path (if **Output File Path** has not been specified)	
<projector name> > *Action* | **Output File Path**	text field	Enter the filename and path where the projector saves images to disk.		

Control	Type	What it does	Opens this dialog box	Notes
<projector name> > *Input Colorspace* | **Data Colorspace** (advanced color management mode only) or **Colorspace** (intermediate color management mode only)	dropdown	Select the colorspace that is displayed on your screen.		
<projector name> > *Input Colorspace* | **Working Colorspace** (advanced color management mode only)	dropdown	Select the colorspace to use for painting, lighting, applying filters, and similar operations.		
<projector name> > *Input Colorspace* | **Raw Data** (advanced and intermediate color management modes only)	checkbox	When enabled, prevents the data from being converted to another colorspace.		
<projector name> > *Input Colorspace* | **Scalar Data**	checkbox	When checked, the data is considered as non-color data such as masks, heights, normals, and depths.		
<projector name> > *Output Colorspace* | **Data Colorspace** (advanced color management mode only)	dropdown	Select the colorspace that is displayed on your screen.		

Control	Type	What it does	Opens this dialog box	Notes
<projector name> > *Output Colorspace* | **Output Colorspace** (advanced color management mode only) or **Colorspace** (intermediate color management mode only)	dropdown	Select the colorspace to use for the output.		
<projector name> > *Output Colorspace* | **Raw Data** (advanced and intermediate color management modes only)	checkbox	When enabled, prevents the data from being converted to another colorspace.		
<projector name> > *Output Colorspace* | **Scalar Data**	checkbox	When checked, the data is considered as non-color data such as masks, heights, normals, and depths.		
<projector name> > *Unprojection* | **Clamp**	checkbox	If this is checked, Mari restricts the range of the paint buffer to values between 0 and 1.		
<projector name> > *Unprojection* | **Shader Used**	dropdown	Select the shader you want the projector to take a screen snapshot of. By default, this is set to **Current Paint Target**.		**Current Shader**, **Current Channel**, **Current Layer**, **Current Paint Target**, and **Current Layer and Below**.
<projector name> > *Unprojection* | **Lighting Mode**	dropdown	The lighting mode to use when taking a screen snapshot with this projector.		Flat, Basic, Full.
<projector name> >	dropdown	The color depth of the screen		8bit (Byte), 16bit

Control	Type	What it does	Opens this dialog box	Notes
Unprojection | **Color Depth**		snapshots taken by this projector.		(Half), 32bit (Float).
<projector name> > *Unprojection* | **Size**	dropdown	The size (in pixels) of the screen snapshots taken by this projector.		
<projector name> > *Painting* | **Painting Mode**	dropdown	Sets the paint blending mode.		The paint blending modes available are similar to many other paint packages. If you aren't familiar with the various modes, see *Paint Blending Modes* in the *Mari User Guide* for descriptions and examples.
<projector name> > *Projection* | **Projection**	dropdown	Sets whether Mari projects only onto the **Front** of the model (as you're seeing it) or whether paint goes straight **Through** the model (appearing on the back as well as the front).		
<projector name> > *Transformation* | **Scale**	text fieldes	Sets the scale of the projector.		
<projector name> > *Transformation* | **Translation**	text fieldes	Sets the translated position of the projector.		
<projector name> >	text field	Sets the rotation, in degrees, of		

Control	Type	What it does	Opens this dialog box	Notes
Transformation | **Rotation**		the projector.		
<projector name> > Update | **Update Global Settings From Projector**	button	If you have adjusted any projection properties, click this button to see the effect of your changes in the canvas. When you do, the equivalent global projection properties in the **Painting** palette are also updated.		Updates all global projection properties.
<projector name> > Update | **Update Only Masks From Projector**	button	If you have adjusted any projection mask properties, click this button to see the effect of your changes in the canvas. When you do, the equivalent global projection mask properties in the **Palette** palette are also updated.		Updates global projection mask properties only.
<projector name> > <projector name> > General | **Name**	information	Name of the camera used for the projector.		
<projector name> > <projector name> > General | **Animated**	information	Whether the camera used for the projector is animated.		True, False.
<projector name> > <projector name> > Perspective | **FoV**	text field, slider	Enter the value for the field of vision. This controls how much Mari distorts the view when applying perspective. At 0, the perspective camera gives exactly the same view as the ortho camera. As the value increases,		

Control	Type	What it does	Opens this dialog box	Notes
		the distortion increases.		
<projector name> > *<projector name>* > *Perspective* | **Near plane**	text field, slider	Enter the value for the near clipping plane. Mari doesn't display parts of the scene that are closer than the Near plane.		
<projector name> > *<projector name>* > *Perspective* | **Far plane**	text field, slider	Enter the value for the far clipping plane. Mari doesn't display parts of the scene that are further than the Far plane.		
<projector name> > *Edge Mask* > *General* | **Enabled**	checkbox	Whether the edge mask is turned on or not.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Edge Mask* > *General* | **Falloff Start**	text field, slider	Controls where the edge mask starts on the model. The higher the value, the more forward facing the masking starts. If this is higher than the **FalloffEnd**, areas facing the view are paintable, and areas oblique to the view are masked. If this is lower, facing areas are masked and oblique areas are paintable.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.

Control	Type	What it does	Opens this dialog box	Notes
<projector name> > Edge Mask > General | **Falloff End**	text field, slider	Controls where the edge mask ends on the model. The higher the value, the closer, more forward facing the mask finishes.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Edge Mask > General | **Falloff Curve**	curve editor	Adjusts the falloff, using a curve editor to modify the falloff start and end point. To fine tun control over the falloff, add more points to the curve.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Channel Mask > General | **Invert**	checkbox	If this is selected, the mask data is black on white rather than white on black.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette

Control	Type	What it does	Opens this dialog box	Notes
				are also updated.
<projector name> > *Channel Mask* > *General* | **Mask Curve**	curve editor	Adjusts the amount of masking on the channel, where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Channel Mask* > *General* | **Mask Contrast**	text field, slider	The contrast applied to the original channel to derive the mask. At 1.0, the mask comes directly from the original channel. Lower values make the mask fuzzier than the input channel, higher values make it sharper than the input channel.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Channel Mask* > *General* | **Mask Amount**	text field, slider	The strength of the mask effect.1.0 means that the mask absolutely controls where you can paint; the effects of the mask decrease as the value gets lower.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global

Control	Type	What it does	Opens this dialog box	Notes
				properties in the **Painting** palette are also updated.
<projector name> > Channel Mask > General | **Mask Channel**	dropdown	Sets the channel holding the mask from a list of available channels in your project.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Channel Mask > General | **Enabled**	checkbox	Whether the channel mask is turned on or not.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Ambient Occlusion Mask > General | **Invert**	checkbox	If this is selected, the mask data is black on white rather than white on black.		
<projector name> > Ambient Occlusion	curve editor	Controls how the ambient occlusion pixel values relate to		

Control	Type	What it does	Opens this dialog box	Notes
Mask > General | **Mask Curve**		the degree of masking. The horizontal axis is the darkness of the occluded pixels (black to the left, white to the right), the vertical axis is the degree of visibility. By default, this is a direct line, where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.		
<projector name> > Ambient Occlusion Mask > General | **Mask Contrast**	text field, slider	The contrast applied to the ambient occlusion to derive the mask. At 1.0, the mask comes directly from the calculated ambient occlusion. Lower values make the mask fuzzier than the ambient occlusion, higher values make it sharper.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Ambient Occlusion Mask > General | **Mask Amount**	text field, slider	The amount of ambient occlusion masking to apply.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.

Control	Type	What it does	Opens this dialog box	Notes
<projector name> > Ambient Occlusion Mask > General | **Enabled**	checkbox	Whether the ambient occlusion mask is turned on or not.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Depth Mask > General | **Enabled**	checkbox	Whether the depth mask is turned on or not.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > Depth Mask > General | **Falloff**	text field, slider	How fast the depth masking applies. Lower values make the mask apply slowly, higher values make it apply more quickly.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette

Control	Type	What it does	Opens this dialog box	Notes
				are also updated.
<projector name> > *Depth Mask* > *General* | **End**	text field, slider	The depth in the scene for the depth masking to finish.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Depth Mask* > *General* | **Start**	text field, slider	The depth in the scene for the depth masking to start.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Backface Mask* > *General* | **Enabled**	checkbox	Whether the backface mask is turned on or not.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent

Control	Type	What it does	Opens this dialog box	Notes
				global properties in the **Painting** palette are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Invert**	checkbox	If this is selected, the mask data is black on white rather than white on black.		
<projector name> > *Fractal Noise Mask* > *General* | **Mask Curve**	curve editor	Adjusts the fractal noise masking where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Enabled**	checkbox	Whether the fractal noise mask is turned on or not.		When enabled, a icon displays in the status bar.
<projector name> > *Fractal Noise Mask* > *General* | **Mask Amount**	text field, slider	The amount that the mask affects the paint buffer. 1.0 means that the mask absolutely controls where you can paint; the effects of the mask decrease as the value gets lower.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette

Control	Type	What it does	Opens this dialog box	Notes
				are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Size**	text field, slider	Determines the size of the fractal noise features.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Roughness**	text field, slider	Determines the roughness of the fractal noise features.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Contrast**	text field, slider	Controls the level of contrast applied to the fractal noise features to derive the mask.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global

Control	Type	What it does	Opens this dialog box	Notes
				properties in the **Painting** palette are also updated.
<projector name> > *Fractal Noise Mask* > *General* | **Seed**	text field, slider	Changes the pattern of the fractal noise by moving the pattern start point through 3D space.		You need to click either of the update buttons under **Update** to see the effect of your changes in the canvas. When you do, the equivalent global properties in the **Painting** palette are also updated.

Selection Groups Palette

The **Selection Groups** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Selection Groups** palette lets you switch between saved selection groups. You can also create new selection groups, lock and unlock groups, and show or hide groups.

Selection Groups Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
	button	Adds a new selection group, based on the current selection.		
	button	Removes the highlighted selection group.		
	button	Reloads the highlighted selection group.		
	button	Locks the selection group.		
	button	Unlocks the selection group.		
	button	Hides the selection group.		
	button	Shows the selection group.		

Color Manager (View Transform) Palette

The **Color Manager (View Transform)** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

> **Note:** The **Color Manager (View Transform)** palette is being deprecated and will be removed in a future release. You can still display it using the following environment variable: **MARI__ENABLE_LEGACY_COLOR_MANAGER_PALETTE=1**
> See Environment Variables That Mari Recognizes for more information.

What It Does

The **Color Manager (View Transform) palette** lets you manage how colors are displayed on your monitor. It includes many options for color correction, including a host of preset filters. Managing colors using the preset filters in this palette does not affect the paint on your model. For information on the filters used to apply transformations to the paint on your model, refer to the Filter Functions section.

Color Manager (View Transform) Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
Options				
Enable	switch, on/off indicator	Turns color correction on or off. When on, the Color Management icon displays in the status bar, and a histogram displays a count of frequencies of colors in RGBA channels in the current view.		
Active	on/off indicator	Displays green when color correction is on.		Not active for scalar channel.
Histogram	switch, on/off indicator	Turns histogram view on or off. If it's on, Mari displays a color histogram in the background of the canvas.		
Split Point	text field, slider	Splits the display between showing corrected and uncorrected color. Enter a value or slide to specify what proportion of the model (split vertically) displays with color correction.		**Color Correction** is on. Range: 0.00 to 1.00
Current Color Profile				
Colorspace	dropdown	Select a saved or preset color profile.		

Control	Type	What it does	Opens this dialog box	Notes
New	button	Prompts you to name a new profile, select available filters to include in a particular order, and save.	Edit	
Edit	button	Rename and change the current profile, including adding filters to or deleting filters from the profile.	Edit	The default filters **Gain**, **LUT**, and **Display** cannot be deleted from the profile
Delete	button	Deletes the current profile (after confirming).	Really Delete?	
Current Color Profile | LUT				
LUT	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.		
General | **File Path**	text field, button	Enter the filepath of the LUT, or select from the file browser.		
Current Color Profile | Composite Filter				
Composite Filter	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.	Add User Attribute	
Current Color Profile | Levels				
Levels	switch, on/off indicator, add attribute	Changes the color levels in the paint, by setting the white point, midtone, and black point.		
Channel | **Color Component**	dropdown	Whether the filter affects all components, or a specific one (**Red**, **Green**, or **Blue**).		
Input Levels | **White Point**	text field, slider	Specifies the upper limit of displayed color intensity. Areas with a value		

Control	Type	What it does	Opens this dialog box	Notes
		higher than this are mapped to 1 (pure white).		
Input Levels | **Midtone**	text field, slider	Sets the middle point between white and black. Mari remaps the values so that this is the middle of the range between the white point and black point. Moving this pushes the values towards that end of the spectrum.		
Input Levels | **Black Point**	text field, slider	Specifies the lower limit of displayed color intensity. Areas with a value lower than this are mapped to 0 (pure black).		
Output Levels | **White Output Levels**	text field, slider	Sets how much white is output in the filtered image. Higher values show the entire white output, lower values show decreasing amounts.		
Output Levels | **Black Output Levels**	text field, slider	Sets how much black is output in the filtered image. Higher values show the entire black output, lower values show decreasing amounts.		
Current Color Profile | sRGB To Linear				
sRGB To Linear	switch, on/off indicator, add attribute	Applies an sRGB to linear colorspace conversion.		
General | **Invert**	checkbox	Applies a linear to sRGB colorspace conversion.		
Current Color Profile | Color Curves				
Color Curves	switch, on/off indicator, add attribute	Changes the color curve for each color channel (RGB) in the painting.	Add User Attribute	Presets can be loaded from **.mgf** (Mari

Control	Type	What it does	Opens this dialog box	Notes
				Gradient Files).
Channels | **Red**	curve editor	Controls how the filter translates current Red values to new values.		
Channels | **Green**	curve editor	Controls how the filter translates current Green values to new values.		
Channels | **Blue**	curve editor	Controls how the filter translates current Blue values to new values.		
Current Color Profile | Color Switches				
Color Switches	switch, on/off indicator, add attribute	Turns individual color channels on or off. (When Mari applies the filter, it removes any unchecked color channels from the painting.)	Add User Attribute	
Channels | **Red**	checkbox	Click to turn the Red channel display on or off.		
Channels | **Green**	checkbox	Click to turn the Green channel display on or off.		
Channels | **Blue**	checkbox	Click to turn the Blue channel display on or off.		
Current Color Profile | Brightness				
Brightness	switch, on/off indicator, add attribute	Changes the brightness of the paint on the model, either overall or for specific channels.	Add User Attribute	
General | **Brightness**	text field, slider, reset	Sets the brightness value for all channels (overwrites changes to individual channels).		Range: 0.00 to 2.00. Reset: 1.00
General | **Contrast**	text field, slider	Sets the contrast value for all channels (overwrites changes to individual channels).		Range: 0.00 to 2.00. Reset: 1.00

Control	Type	What it does	Opens this dialog box	Notes
Channels | **Blue**	text field, slider, reset	Sets the brightness value for the blue channel.		Range: 0.00 to 2.00. Reset: 1.00
Channels | **Green**	text field, slider, reset	Sets the brightness value for the green channel.		Range: 0.00 to 2.00. Reset: 1.00
Channels | **Red**	text field, slider, reset	Sets the brightness value for the red channel.		Range: 0.00 to 2.00. Reset: 1.00
Current Color Profile | Blur				
Blur	switch, on/off indicator, add attribute	Applies a standard blur to the selection.	Add User Attribute	
General| **Blur Radius**	text field, slider, reset	Sets the blur radius - the number of pixels Mari uses when calculating the blur value for each pixel. The higher the radius, the more blurred the results. If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.		Range: 0 to 10. Reset: 4 If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.
Current Color Profile | Tone Mapping				
Tone Mapping	switch, on/off indicator, add attribute	Varies the "exposure" of the painting.	Add User Attribute	
General | **White Point**	text field, slider, reset	Sets the upper limit of displayed color intensity - any colors of greater intensity map to white.		Range: 0.00 to 1.00. Reset: 1.00

Control	Type	What it does	Opens this dialog box	Notes
General | **Exposure**	text field, slider, reset	Sets the exposure - lower exposures give darker images; higher exposures give lighter ones.		Range: 0.00 to 10.00. Reset: 1.00
Current Color Profile | Gamma				
Gamma	switch, on/off indicator, add attribute	Changes the gamma levels in the selection.	Add User Attribute	A higher setting for **Gamma** results in a more washed-out looking image.
General | **Do Inverse**	checkbox	Reverses the gamma conversion.		
General | **Gamma**	text field, slider, reset	Sets the gamma level.		Range: 0.00 to 3.00. Reset: 2.20
Current Color Profile | Linear To Log				
Lin 2 Log	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.	Add User Attribute	
General | **Gamma**	text field, slider, reset	Sets the gamma level.		Range: 0.00 to 1.00. Reset: 0.600
General | **White**	text field, slider, reset	Sets the upper limit of displayed color intensity - any colors of greater intensity map to white.		Range: 0.000 to 1023.000. Reset: 685.000
General | **Ignore Black**	checkbox	Ignores the threshold set by the **Black** control.		
General | **Black**	text field, slider, reset	Sets the lower limit of displayed color intensity - any colors of lesser intensity map to black.		Range 0.000 to 1023.000. Reset: 95.000
Current Color Profile | Hue				

Control	Type	What it does	Opens this dialog box	Notes
Hue	switch, on/off indicator, add attribute	Changes the hue, saturation, and value of the colors on the model.	Add User Attribute	
Color | **Calculation**	dropdown	Determines whether the colorspace is set to use the Mari default **Original (HSV)**, or **Photoshop (HSL)**.		
Hue | **Hue**		Change the hue by setting the values. Adjusting the slider, moves the colors around the color wheel. The rotation value is the degree around the color wheel that each color shifts, between 0 and 360 (which both correspond to the original colors).		
Saturation | **Saturation**	curve editor	Controls how the filter translates current saturation to new saturation.		
Saturation | **Offset**	text field, slider, reset	Adds an offset value to the initial saturation, before applying the **Scale**.		Range: 0.00 to 1.00. Reset: 0.00
Value | **Value**	curve editor	Controls how the filter translates current values to new values.		
Value | **Offset**	text field, slider, icon	Adds an offset value to the initial value, before applying the **Scale**.		Range: 0.00 to 1.00. Reset: 0.00
Current Color Profile | Invert				
Invert	switch, on/off indicator, add attribute	Inverts the colors in the selection. Replaces colors with their "opposite" in the color chart. For example, replaces a color with a float value of 0.3 with a color with float value 0.7.	Add User Attribute	

Control	Type	What it does	Opens this dialog box	Notes
General | **Invert Alpha**	checkbox	Inverts the alpha channel.		
Current Color Profile | Premultiply Alpha				
Premultiply Alpha	switch, on/off indicator, add attribute	Either pre- or post-multiplies the alpha in the selected image. If you are painting onto an image with a premultiplied alpha, using an image without, use this filter to perform the premultiplication, so the images match and you avoid lines around the outside of the patch.	Add User Attribute	
General | **Unpremultiply**	checkbox	If the **Unpremultiply** box is checked, this acts in reverse of premultiplication (that is, it removes the premultiplied alpha)		
Current Color Profile | Log To Linear				
Log To Linear	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.	Add User Attribute	
General | **Gamma**	text field, slider, reset	Sets the gamma level.		Range: 0.00 to 1.00. Reset: 0.600
General | **White**	text field, slider, reset	Sets the upper limit of displayed color intensity - any colors of greater intensity map to white.		Range: 0.000 to 1023.000. Reset: 685.000
General | **Ignore Black**	checkbox	Ignores the threshold set by the **Black** control.		

Control	Type	What it does	Opens this dialog box	Notes
General | **Black**	text field, slider, reset	Sets the lower limit of displayed color intensity - any colors of lesser intensity map to black.		Range 0.000 to 1023.000. Reset: 95.000
Current Color Profile | Luminosity				
Luminosity	switch, on/off indicator, add attribute	Outputs just the luminosity value of each pixel in the selection. That is, it outputs a grayscale image based on the brightness of each of the pixels in the original image.	Add User Attribute	
Current Color Profile | Contrast				
Contrast	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.	Add User Attribute	
General | **Amount**	text field, slider, reset	Changes the contrast of the paint.		Range: 0.00 to 2.00. Reset: 1.00
Current Color Profile | Clamp				
Clamp	switch, on/off indicator, add attribute	Clamps color values to lie within the specified upper and lower values. Set the **Upper Value** and **Lower Value** by adjusting the sliders or entering values in the text fieldes. Select the individual **Color Component** to clamp in the dropdown menu.		
General | **Upper Value**	text field, slider	Set the upper value at which color values are clamped. All values higher than this number are set to this value.		
General | **Lower**	text field,	Set the lower value at which color		

Control	Type	What it does	Opens this dialog box	Notes
Value	slider	values are clamped. All values lower than this number are set to this value.		
Channel | **Color Component**	dropdown	Select the individual color component to clamp.		
Current Color Profile | Copy Channel				
Copy Channel	switch, on/off indicator, add attribute	Copies the value from one RGB color channel to the other two. The result is a grayscale image with the intensity values from the selected channel.	Add User Attribute	
General | **Copy Channel**	dropdown	Select the source channel from the list (**Red**, **Green**, or **Blue**). When you apply the filter, Mari copies the selected channel over the other two channels.		
Current Color Profile | Soften				
Soften	switch, on/off indicator, add attribute	Applies a subtle softening blur. This is a quick, predefined blur filter. If you need more control over the degree or type of blurring, use either the standard **Blur** or **Gaussian Blur** filters.	Add User Attribute	
Current Color Profile | Edge Detect				
Edge Detect	switch, on/off indicator, add attribute	Finds "edges" (transitions between colors) in the image and heightens them.	Add User Attribute	
Current Color Profile | Emboss				
Emboss	switch, on/off	Applies an "emboss" effect, so that	Add User	

Control	Type	What it does	Opens this dialog box	Notes
	indicator, add attribute	textures look “raised” on the surface.	Attribute	
Current Color Profile | Sharpen				
Sharpen	switch, on/off indicator, add attribute	Removes 'fuzziness' and clarifies detail in the image.	Add User Attribute	
General | **Amount**	text field, slider, reset	Sets the amount of sharpening. Higher values mean more sharpening, while a value of 0.00 is the picture without the filter applied.		Range: 0.00 to 1.00
Current Color Profile | Dilate				
Dilate	switch, on/off indicator, add attribute	Switches the filter on or off, or lets you add attributes to it.	Add User Attribute	
Current Color Profile | Gaussian1D				
Gaussian1D	switch, on/off indicator, add attribute	Applies a Gaussian blur. Compared to the standard **Blur** filter, this gives you much finer control over the degree of blurring, and the option of using much higher blur values.	Add User Attribute	
General | **Axis**	text field, incrementer	Apply the blur along the X or Y axis. If the field is set to 0, the blur is along the X axis; if it is anything other than 0, the blur is applied along the Y axis.		
General | **Radius**	text field, slider	Sets the blur radius. The higher the radius, the more blurred the results. If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is		Range: 0.00 to 50.00

Control	Type	What it does	Opens this dialog box	Notes
		an artifact of the way that Mari handles the edges of patches.		
Current Color Profile | Gaussian				
Gaussian	switch, on/off indicator, add attribute	Applies a Gaussian blur. Compared to the standard **Blur** filter, this gives you much finer control over the degree of blurring, and the option of using much higher blur values.	Add User Attribute	
General | **Amount**	text field, slider	Sets how much blur is applied.		
General | **Radius**	text field, slider	Sets the blur radius. The higher the radius, the more blurred the results. If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.		Range: 0.00 to 50.00
Current Color Profile | High Pass				
High Pass	switch, on/off indicator, add attribute	Removes low frequency information from the image while maintaining higher frequency detail.		
General | **Amount**	text field, slider	Sets how much low frequency information is removed by the filter.		
General | **Radius**	text field, slider	Controls the number of pixels around the image highlights that are affected by the filter.		
Current Color Profile | Color Correction				
Color Correction	switch, on/off indicator, add	Switches the filter on or off, or lets you add attributes to it.		

Control	Type	What it does	Opens this dialog box	Notes
	attribute			
User Attributes | **OCIO Config**	text field, button	Specifies the configuration file to use.		
User Attributes | **Input Colorspace**	dropdown	The colorspace of the current channel or image values.		
User Attributes | **Output Colorspace**	dropdown	The colorspace used to convert from the channel or image values to the values written to file.		
Current Color Profile | Add Noise				
Add Noise	switch, on/off indicator, add attribute	Applies noise to the current channel or image values.		
User Attributes | **Amount**	text field, slider	Adjusts how much noise is applied.		
User Attributes | **Color Component**	dropdown	Sets the channel you want to affect.		
User Attributes | **Grayscale**	checkbox	Applies grayscale noise with only the intensity values from the selected channel.		
User Attributes | **Seed**	text field, slider	Adds variation to the noise.		
User Attributes | **Size**	text field, slider	Adjusts the size of the noise applied.		

Image Manager Palette

The **Image Manager** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Image Manager** lets you store and work with images in your project, including using them for brushes and painting through them onto your model. For organization purposes, you can also add custom tabs to the palette by clicking the add tab button.

You can also create Image Groups to organize your project's images.

Image Manager Palette Fields

Control	Type	What it does	Opens this dialog box
Image Info > Info			
Depth	information	The color depth of the currently selected image in bytes.	
Channels	information	The color components of the currently selected image.	
File Space	dropdown	File space of the currently selected image from the options: • **NORMAL** • **VECTOR** • **VECTOR_Y_FLIP** • **NORMAL_Y_FLIP**	
Tiled Image	information	Whether the image is tiled - true or false.	
Height	information	The height of the currently selected image in	

Control	Type	What it does	Opens this dialog box
		pixels.	
Width	information	The width of the currently selected image in pixels.	
Path	information	The file path to the image that Mari is accessing. If the original image was saved to another location this is the file path shown, not the path to the original file.	
Image Info > User Attributes			
Comment	text field	Assign a user-defined comment to the image.	
MriImportFilePath	information	The file path to the original image.	
MriLastImportDateTime	information	The date and time that the file was first loaded into Mari.	
Colorspace			
Colorspace	dropdown	The colorspace to which the output values are converted. **Automatic** is determined by the file name, size, and type of data in the image.	
Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.			
Raw Data	checkbox	If this is enabled, the image data is not converted.	

Control	Type	What it does	Opens this dialog box
		Note: As the raw colorspace nullifies the corresponding colorspace transform when either used as the input or output, there is a discrepancy between what is shown in the **Image Manage** palette thumbnail and the image viewer. This is the expected behavior.	
	button	Adds an image to the Image Manager.	Open an Image Dialog
	button	Removes the selected image from the Image Manager.	
	button	Opens the selected image in another window, where you can crop it.	<image>
	button	Saves the selected image to a file.	Save As Dialog
Send Image to 2D Paint	button	Opens your image in the **2D** tab. Mari will add a new object and import the image as a Paint Node. The node graph will show Bottom Transparency + Paint node (with the image imported) merged and connected to a channel node.	Using 2D Paint Mode
<image> window			
	button	Crops the current image. Mari copies the cropped area as a separate image in the **Image Manager**. The cropped image is part of the project, but won't be saved as a separate image file unless you use the **Save As** option.	

Control	Type	What it does	Opens this dialog box
		Select one of the following crop modes: • **Arbitrary** - click-and-drag on an arbitrary area of the image. • **Fixed** - enter a fixed size for the crop box (in pixels) and drag it to the area you want to crop. • **Aspect** - enter a fixed aspect ratio for the crop box (in pixels) and drag it over the area you want to crop. Once you have selected the crop area, click the mouse button to crop.	
Colorspace	dropdown	The colorspace to which the output values are converted. **Automatic** is determined by the file name, size, and type of data in the image.	
Raw Data	checkbox	If this is enabled, the image data is not converted. **Note:** As the raw colorspace nullifies the corresponding colorspace transform when either used as the input or output, there is a discrepancy between what is shown in the **Image Manage** palette thumbnail and the image viewer. This is the expected behavior.	
Image Group window			
Shader Model	dropdown	Defines the available shader models you can assign images to. The available options are: • 3Delight Principled • Arnold Standard Surface	

Control	Type	What it does	Opens this dialog box
		• Autodesk Standard Surface (1.0.2.x) • BRDF • Principled BRDF • USD Preview Surface • Unreal • VRayMtl • VRayMtl (V-Ray 6)	
Ingest Template	submenu	Apply preset configurations for specific texture vendors to the shader model. There are two options: • **Apply to Current Shader Model** - matches images to streams based on filenames in the selected preset. This will only apply for the Shader model selected in the dropdown. • **Apply to All Shader Models** -matches images to streams based on filenames in the selected preset. This will be applied across all shader models. Available texture vendors to apply templates to are: • CCO_Textures Friendlyshade • GameTexture • Poliigon • Quixel • RealDisplacementTextures • Substance • TextureHaven • Textures_Com • TexturingXYZ	
Auto-Assign Images	Button	Auto-assigns the images in the Image group to the corresponding shader channels, based on	

Control	Type	What it does	Opens this dialog box
		naming conventions of images.	
Search	Text input	Search for a specific stream in the shader model.	
Lock Stream Unlocked: Locked:	Button	Locks the specific stream. When a stream is locked, it will not be altered when the Auto-assign and Ingest template tools are used. You can still manually change the stream by using the image dropdown and picking a new image	
Stream	n/a	The shader stream that the image can be assigned to.	
Image	Dropdown	Defines the image assigned to the shader stream.	

Texture Sets Palette

The **Texture Sets** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Texture Sets** palette manages texture sets or images (for example Megascans assets) that you can use for painting. You can browse and import texture sets to the **Image Manager** palette to work with images in your project. See Managing Images for more information.

Texture Sets Palette Fields

Control	Type	What it does	Opens this dialog box
Megascans tab			
	icon, slider	Resizes the Megascans previews in the **Megascans** tab. Dragging the slider to the left makes the previews smaller, while dragging it to the right makes them larger.	
Megascans asset type	dropdown	Specifies which type of Megascans asset to display. You can choose between the following options: **All types** - displays all available scans. **Atlases** - displays vegetation atlas scans. **Surfaces** - displays tileable surface scans.	
Filters	dropdown	Specifies which Megascans categories to display. You can choose between the categories used in the Megascans imported in Mari. The more filter options you add, the more assets display in the tab.	
Refresh	button	Updates the available Megascans assets once you uploaded new ones to your machine. To load or update your Megascans assets, see Managing Megascans Assets or Mari Preferences Dialog.	

History View Palette

The **History View** palette is described below.

What It Does

The **History View** palette displays a list of actions performed in the current session. Click an action to step back to that version of the project.

> **Video:** To learn how Mari records actions and events into history, watch the Understanding History in Mari video.
> This video shows the workflow using Mari 3. Even though the Mari 4 workspace is different, the workflow remains the same. To have a look at the main UI differences, see Mari 3.3 vs 4.0.

Python Console Palette

The **Python Console** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Python Console** palette is divided into three sections:

- a **Script Path** text field that loads Python scripts from a defined location
- an input pane that is used to enter and execute Python statements in the lower half of the palette
- an output pane that displays statements and their outputs in the upper half of the palette.

Python Console Palette Fields

Control	Type	What it does	Notes
Output Pane	information	Displays Python statements and their outputs.	
Input Pane	text field	Click inside to enter Python statements.	
Script Path	text field	Enter the file path where the console loads scripts from.	
	button	Selects a Python file for evaluation in the console. Selecting a file fills in the **Script Path** but you need to click **Evaluate** to continue.	Opens the **Python Script Path** dialog.
Evaluate	button	Executes Python script or statement.	
Clear	button	Clears the input and output panes.	
Help	button	Opens the **Mari Python API Help** in a new window. This window is visible until you click **Help** again or close the window. If you have the help open while typing in the Input Pane with auto-complete on, commands that appear in auto-complete are automatically found in the help as well.	

Colors Palette

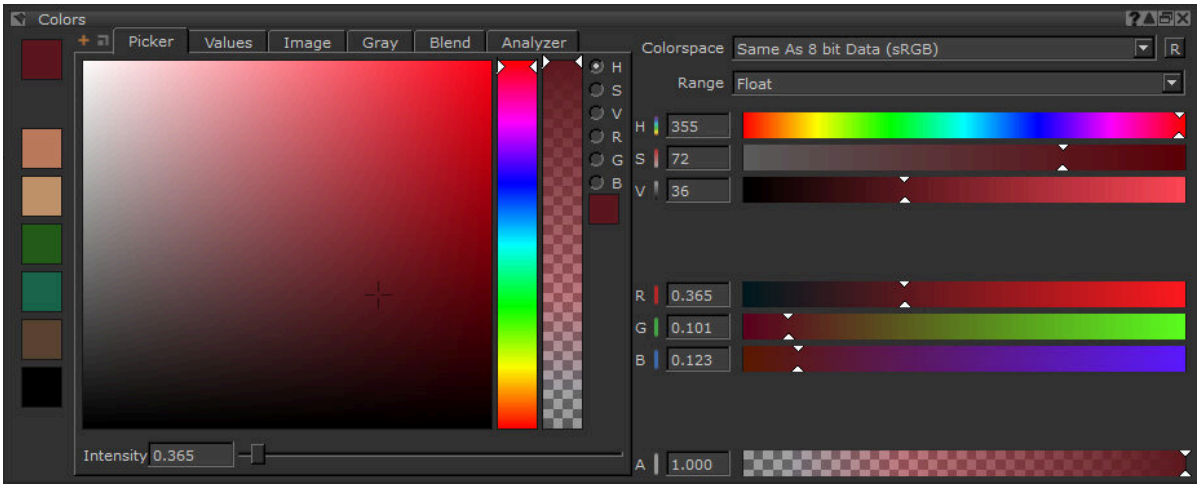
The **Colors** palette and the controls that can be found on it are described in the table below.

Note: **Value** can go beyond 100% if picking colors from an HDR image and **Saturation** can go beyond 100% if picking colors from a Channel in a colorspace with a wider gamut than the color picking colorspace.

What It Does

The **Colors** palette lets you view and change colors. The controls on this palette all work together - as you change the color through one control, the other controls update to show your change.

Colors Palette Fields



Note: Picking values for HDR images outside of the range as defined by the color picking OCIO colorspace, may result in clamped values.

Control	Type	What it does	Notes
Swatches	swatch	These swatches store the 6 most recently selected colors and the current foreground color (the one at the top). When a new color is selected, it is added to the top of the list. Clicking on a swatch	The current foreground color is the color that Mari uses when you paint using the **Paint** tool.

Control	Type	What it does	Notes
		selects it as the current color (foreground color).	
	button	Adds any of the following tabs: • **Color History** • **Picker** • **Colorspace** • **Sliders** • **Values** • **Image** • **Blend** • **Gray** • **Analyzer**	
	button	Opens tabs in separate windows.	
Picker			
Color field	field	Click to select a color.	
Color slider	vertical slider	Click and drag up and down to select a color.	
Transparency/Alpha slider	vertical slider	Click and drag up and down to select the level of transparency.	
H S V R G B	radio buttons	Select whether to display **Hue**, **Saturation**, **Values**, **Red**, **Green**, or **Blue** in the Color field and the Color slider.	
Color swatch	swatch	The current foreground color.	The current foreground color is the

Control	Type	What it does	Notes
			color that Mari uses when you paint using the **Paint** tool.
Intensity	input box, horizontal slider	Enter a value in the input field or click and drag from left to right to adjust the intensity of colors to select HDR colors. HDR colors are considered to be any color with an intensity that is greater than 1.	From 0 to 4096; defaults to 4096. The maximum range of the **Intensity** slider can be adjusted in the Mari Preferences Dialog under **Color** > **Color Management Defaults** | **Color Picker Maximum**.
Values			
Source (sRGB)			
If your channel has **Scalar** enabled, the **Source (sRGB)** sliders are disabled and are not directly editable. These sliders will, however, update to reflect any values present in the **Target** parameters.			
	text field, slider	Specify **Hue** values in degrees.	Range: 0 to 360°
	text field, slider	Specify **Saturation** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Value** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Red** values.	Range: 0 to 255
	text field, slider	Specify **Green** values.	Range: 0 to 255
	text field, slider	Specify **Blue** values.	Range: 0 to 255

Control	Type	What it does	Notes
	text field, slider	Specify **Alpha** values.	Range: 0.0 to 1.0
Target (linear)			
	text field, slider	Specify **Hue** values in degrees.	Range: 0 to 360°
	text field, slider	Specify **Saturation** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Value** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Red** values.	Range: 0 to 255
	text field, slider	Specify **Green** values.	Range: 0 to 255
	text field, slider	Specify **Blue** values.	Range: 0 to 255
	text field, slider	Specify **Alpha** values.	
Image			
Image field	field	Drag-and-drop an image in the image field then click the image to select a foreground color. You can also load an image by right-clicking and selecting **Load Image**, then selecting a file from the **Load** dialog box. To delete the image, right-click it and select **Clear Image**.	You can drag-and-drop an image from the Image Manager Palette or from a file browser.

Control	Type	What it does	Notes
Gray			
Gradient, presets	sliders, buttons	Use this to pick a gray value from the gradient, or one of the presets.	
Intensity	input box, slider	Enter a value in the input field or click and drag from left to right to adjust the intensity of colors to select HDR colors. HDR colors are considered to be any color with an intensity that is greater than 1.	From 0 to 4096; defaults to 4096. The maximum range of the **Intensity** slider can be adjusted in the Mari Preferences Dialog under **Color** > **Color Management Defaults** | **Color Picker Maximum**.
Blend			
Color field	field	This displays a field with four colors blending into each other. Click any of the color swatches (foreground color or recently used colors) and drag-and-drop them to any corners of the color field. Then click in the color field to pick from the blend.	
Analyzer			
Current	swatch	Displays the color of the pixel selected with the **Eye Dropper** tool.	
Minimum	swatch	Displays the minimum color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	

Control	Type	What it does	Notes
Average	swatch	Displays the average color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	
Maximum	swatch	Displays the maximum color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	
Accumulate	checkbox	When selected, keeps data on all pixels selected (until **Reset**), and calculates **Minimum**, **Average** and **Maximum** colors.	
Reset	button	Clears all accumulated pixel selections, to just the Current selection.	
Values | **R, G, B, A**	information	Values for Red, Green, Blue, and Alpha.	
Values | **H, S, V, L**	information	Values for Hue, Saturation, Value, and Luminosity.	
Values | **X, Y E, I**	information	Values for position of pixel on x and y axis, and the pixel Exposure and Intensity.	
	button	Deletes the currently displayed tab.	
Colorspace			
Colorspace	dropdown	The colorspace in which the colors are interpreted. **Same as 8 bit Data** is determined by the **Color Picking** control in the **Project**	This colorspace can be reset to the default option by clicking .

Control	Type	What it does	Notes
		Settings, or the same control set at project creation. OCIO colorspace roles are also displayed in this list.	The **Colorspace** menu is linked to the **Source** column in the **Values** tab. If **Scalar** is enabled when you create your channel, the **Colorspace** menu is grayed out and disabled. See **Color Swatches and Pickers** preference in the Mari Preferences Dialog.
Range	dropdown	Set color values to either **Byte** (8-Bit), **Half** (16-Bit), or **Float** (32-Bit) .	
Sliders			
H S V	input box, sliders	Enter a value in the input field or click and drag from left to right to select Hue, Saturation, and Value.	
R G B	input box, sliders	Enter a value in the input field or click and drag from left to right to select Red, Green, and Blue.	
Alpha	input box, slider	Enter a value in the input field or click and drag from left to right to select the level of transparency.	

Tool Properties Palette

The **Tool Properties** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Tool Properties** palette displays information about the selected tool.

Tool Properties Palette Fields

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Select > *Select* | **Facing**	dropdown	Sets whether the selection only applies to the **Front** of the model as you can see it, or if the selection goes **Through** the model and includes the other side.		
	Transform Selected Objects No properties.				
	Paint | **Paint**	add attribute	Click to add attributes	Add User Attribute	
	Paint > *Paint* |**Description**	information	Specifies the tool function.		
	Paint > *Paint* | **Name**	information	Specifies the tool name.		
	Paint >*Paint* | **Brush Settings**		Brush texture details. See Brush Properties.		
	Paint > *Paint* | **Brush**		Brush geometry details. See		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Tip		Brush Properties.		
	Blur | **Blur**	add attribute	Click to add attributes.	Add User Attribute	
	Blur > *Blur* | **Description**	information	Specifies the tool function.		
	Blur > *Blur* | **Name**	information	Specifies the tool name.		
	Blur > *Blur* | **Brush settings**	add attribute	Brush tip details. See Brush Properties.	Add User Attribute	
	Blur > *Blur* | **Brush Tip**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Blur > *Blur* | **Texture**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Paint Buffer Eraser > (Same as **Paint**)	add attribute	Click to add attributes.	Add User Attributes	
	Vector Paint | **Vector Paint**	add attribute	Click to add attributes.	Add User Attributes	
	Vector Paint>*Vector Paint* | **Description**	information	Specifies the tool function.		
	Vector Paint> *Vector Paint* | **Name**	information	Specifies the tool name.		
	Vector Paint > *Vector Paint* | **Brush settings**	add attribute	Brush tip details. See Brush Properties.	Add User Attribute	
	Vector Paint > *Vector Paint* | **Brush Tip**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Paint Through | **Paint Through**	add attribute	Click to add attributes.	Add User Attributes	

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Paint Through >Paint Through | **Description**	information	Specifies the tool function.		
	Paint Through > Paint Through| **Name**	information	Specifies the tool name.		
	Paint Through > Paint Through | **Brush settings**	add attribute	Brush tip details. See Brush Properties.	Add User Attribute	
	Paint Through> Paint Through| **Brush Tip**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Paint Through> Paint Through| **Texture**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Gradient | **Gradient**	add attribute	Click to add attributes.	Add User Attributes	
	Gradient > Gradient | **Description**	information	Specifies the tool function.		
	Gradient > Gradient | **Name**	information	Specifies the tool name.		
	Gradient > Gradient | **Brush settings**	add attribute	Brush tip details. See Brush Properties.	Add User Attribute	
	Gradient > Gradient | **Brush Tip**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Gradient > Gradient | **Texture**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Clone Stamp | **Clone Stamp**	add attribute	Click to add attributes.	Add User Attributes	

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Clone Stamp>*Clone Stamp*| **Description**	information	Specifies the tool function.		
	Clone Stamp>*Clone Stamp* | **Name**	information	Specifies the tool name.		
	Clone Stamp> *Clone Stamp* | **Brush settings**	add attribute	Brush tip details. See Brush Properties.	Add User Attribute	
	Clone Stamp> *Clone Stamp*| **Brush Tip**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Clone Stamp> *Clone Stamp*| **Texture**	add attribute	Brush texture details. See Brush Properties.	Add User Attribute	
	Warp | **Warp** No properties.	add attribute	Click to add attributes	Add User Attribute	
	Slerp | **Slerp** No properties.	add attribute	Click to add attributes.	Add User Attribute	
	Pinup | **Pinup** No properties.	add attribute	Click to add attributes.	Add User Attribute	
	Towbrush > *Towbrush* >**Mode**	dropdown	Selects the Towbrush mode: • Clone from the selected region to another part of the surface. • Overwrite the selected region with another part of the surface. • Swap the contents of the selected region with another area on the surface.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Towbrush > *Towbrush* >**Falloff**	text field, slider	Sets how sharply the texture blends in from the edges of the selected patch. Higher values have sharper transition to the surrounding texture, lower values blend more smoothly.		
	Towbrush > *Towbrush* >**Value**	text field, slider	Controls how much of the color of the original patch to preserve. Higher values keep more of the original color, lower values blend the whole patch in with the background as you move it.		
	Marquee Select > *Marquee Select*| **Crawling Ants**	checkbox	Controls whether the selection edges are animated. If on, the edges are animated.		
	Marquee Select > *Marquee Select* | **Type**	dropdown	The shape of the selection drawn: • **Lasso** - lets you draw a "free" selection of any shape. • **Rectangle** - lets you draw a rectangular selection. • **Polygonal** - lets you draw a polygonal selection of any shape by setting multiple points before closing the selection. • **Ellipse** - lets you draw an oval selection. • **MagicWand** - lets you		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			select by color. When you click, Mari creates a selection around the point you clicked. The selection is based on color – areas with a similar color to the original point are selected. You can change the color threshold for the selection.		
	Marquee Select > *Marquee Select*| **Mode**	dropdown	How drawing a new selection affects the current marquee selection: • **Replace** - the new selection replaces the existing selection. • **Transform** - this lets you move you selection area around (move, rotate, or scale the selection). • **Add** - the new selection adds onto the existing selection. • **Subtract** - the new selection removes areas of the existing selection. • **Intersect** - inverts the existing selection. That is, any areas that were selected become unselected, and any areas that were unselected become selected.		
	Marquee Select >	button	Fills the selected paint buffer		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Marquee Select| **Fill Selection with Foreground**		area with the current foreground color.		
	Marquee Select > *Marquee Select*| **Invert Marquee Selection**	button	Inverts the current selection. Unselected areas become selected, and the other way around.		
	Marquee Select > *Marquee Select* | **Lift Marquee Selection to Paint Buffer**	button	Lifts the selected texture and stores it in the paint buffer.		
	Marquee Select > *Marquee Select*| **Clear Marquee Selection**	button	Clears the current selection.		
	Marquee Select > *General* | **Enabled**	checkbox	Turns the marquee selection on and off.		
	Marquee Select > *General* | **Color Threshold**	text field, slider	Sets the color threshold for the marquee selection. Setting a lower value here results in a more accurate color selection. Higher values catch more of the surrounding color.		
	Marquee Select > *General* | **Amount**	text field, slider, reset	Sets how much the marquee selection prevents painting outside the selected area. This works as a multiplier to the alpha of paint applied outside the selection: • At 1, only the area inside the selection is paintable. • At 0.5, paint outside the		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			selection applies with an alpha of 0.5.		
	Marquee Select > Selection Edge | **Feathering**	text field, counter	Controls how sharp the edge of the selection area is. At 0, the selection has a completely sharp edge. As the value rises, the edges become softer.		
	Marquee Select > Transform | **Reset Marquee Transform**	button	Resets any transformations (moves, rotations, scales) applied to the marquee selection area.		
	Marquee Select > Transform | **Center**	text field	Sets the center of the marquee selection.		Use the break link button to break the link between the two fields.
	Marquee Select > Transform | **Rotation**	text field	Sets the rotation, in degrees, of the marquee selection.		
	Marquee Select > Transform | **Scale**	text field	Sets the scale of the marquee selection.		1 is the original scale.
	Marquee Select > Transform | **Translation**	text field	Sets the translated position onscreen of the marquee selection.		
	No properties.				

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Zoom Paint Buffer > Zoom Paint Buffer | **Paint Buffer Zoom**	information	This tool lets you zoom in and out of the scene with the paint buffer locked to the object and without baking the paint.	Add User Attribute	You can also enable this tool by pressing **Z**. Zooming can be set to be reset when paint is baked.
	Vector Inspector > *Vector Inspector* | **InspectorTool**	add attribute	Click to add attributes.	Add User Attribute	
	Vector Inspector > *Vector Inspector* |**Spacing**	text field, slider	Adjusts the spacing between the directional markers.		**Shift**+ drag the mouse when the vector inspector is active.
	Vector Inspector > *Vector Inspector* | **Length**	text field, slider	Adjust the length of the directional markers.		
	Vector Inspector > *Vector Inspector* | **Width**	text field, slider	Adjust the width of the directional markers.		
	Vector Inspector > *Vector Inspector* | **Start color**	swatch	Change the color at the base of the directional markers.		
	Vector Inspector > *Vector Inspector* | **End color**	swatch	Change the color at the tip of the directional markers.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Vector Inspector > *Vector Inspector* | **Vector Space**	dropdown	Set whether the vector inspector operates in **ScreenSpace**, **TangentSpace**, or **WorldSpace**.		
	Vector Inspector > *Vector Inspector* | **Rotation**	text field, slider	Adjusts the rotation of the directional markers. This can be changed using the slider, or by entering a specific value.		
	Vector Inspector > *Vector Inspector* | **Translation**	text fieldes	The exact location in XY space of the vector inspector. Entering values adjusts the position of the inspector too.		
	Eye Dropper > *General* | **Sample Size**	text field, slider	Radius in pixels that the eyedropper uses when sampling colors off the surface of the model.		Range: 1 to 15

Painting Palette

The **Painting** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Painting** palette displays information and controls for managing painting features.

Painting Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
Painter > *Filtering* | **Image**	dropdown	The image filtering to use when copying textures using the clone tools (**CloneStamp** and **PaintThrough**). One of: • **Nearest** - preserves edge detail, but gives quite "blocky" textures. • **Bilinear** - gives good results, but can produce square artifacts at extreme zoom. • **Bicubic** - provides more rounded results, slightly blurrier but without the square artifacts.		
Painter > *SourceGrade* | **Enabled**	checkbox	Enables source image grading. Used this to grade your source image when cloning from or painting through an image.		
Painter > *SourceGrade* | **Blackpoint**	text field, slider	Sets the blackpoint of the RGB values.		Blackpoint is the color value at which the input is considered to be 100% black.
Painter > *SourceGrade* | **Whitepoint**	text field, slider	Sets the whitepoint of the RGB values.		Whitepoint

Control	Type	What it does	Opens this dialog box	Notes
				is the color value at which the input is considered to be 100% white.
Painter > *SourceGrade* | **Lift**	text field, slider	Lifts the blackpoint, while keeping the whitepoint the same.		
Painter > *SourceGrade* | **Gain**	text field, slider	Adjusts the whitepoint, while keeping the blackpoint the same		
Painter > *SourceGrade* | **Multiply**	text field, slider	Multiplies the value to lighten the texture while preserving the blackpoint.		
Painter > *SourceGrade* | **Offset**	text field, slider	Specifies a fixed value to add in order to lighten the texture.		Adding negative values essentially darkens the texture.
Painter > *SourceGrade* | **Gamma**	text field, slider	Adjusts the midtones, while keeping the whitepoint and blackpoint the same.		
Painter > *SourceGrade* | **Rgb**	text field, slider	Changes the overall color saturation for all three color components of the source image. Moving this also moves		

Control	Type	What it does	Opens this dialog box	Notes
		the separate **Red**, **Green**, and **Blue** sliders.		
Painter > *SourceGrade* | **Red**	text field, slider	Changes the color saturation in the Red component of the source image.		
Painter > *SourceGrade* | **Green**	text field, slider	Changes the color saturation in the Green component of the source image.		
Painter > *SourceGrade* | **Blue**	text field, slider	Changes the color saturation in the Blue component of the source image.		
Painter > *SourceGrade* | **Contrast**	text field, slider	Changes the contrast of the source image.		
Painter > *SourceGrade* | **Saturation**	text field, slider	Changes the color saturation of the source image. Used in addition to the RGB sliders.		
Painter > *SourceGrade* | **Value**	curve editor	Allows you to edit the color value curve for the source image.		
PaintBuffer > *PaintBuffer* | **ColorDepth**	dropdown	Color depth of the paint buffer. 8, 16, or 32-bit color.		
PaintBuffer > *PaintBuffer* | **BufferSize**	dropdown	Onscreen resolution of the paint buffer. This controls the amount of detail you can use when painting, not the resolution of the patches in your project.		
PaintBuffer > *PaintBuffer* | **Clamp**	checkbox	If this is checked, Mari restricts the range of the image to values		

Control	Type	What it does	Opens this dialog box	Notes
		between 0 and 1.		
PaintBuffer > *Transform* | **Translation**	text fieldes	The amount that the buffer has been translated (moved onscreen) from the default.		
PaintBuffer > *Transform* | **Scale**	text fieldes	The amount that the buffer has been scaled (resized) from the default.		
PaintBuffer > *Transform* | **Rotation**	text field, slider	The amount that the buffer has been rotated from the default.		
PaintBuffer > *Transform* | **Reset**	button	Resets the paint buffer to the default values, removing any translations or distortions you have applied.		
PaintBuffer > *Transform* | **Resetonbake**	dropdown	Either: • **Enabled** - the paint buffer automatically resets to default settings every time you bake the paint. • **Disabled** - any changes you have made to the paint buffer stay until you manually click the **Reset** button.		
Projection Settings > *Masking* | **Masking Disabled**	checkbox	Disables all masking, regardless of channel, and overrides the **Masking Enabled** setting.		When enabled, a icon displays in the status bar.

Control	Type	What it does	Opens this dialog box	Notes
Projection Settings > *Mask*ing | **Mask Preview Color**	swatch	What color to display the masking on-screen.		
Projection Settings > *Mask*ing | **MaskPreviewEnabled**	checkbox	Shows the masking. Areas that are covered by the current defined mask(s) are shown colored. This includes all the masking currently in use – for example, if you have both the edge mask and depth mask turned on, Mari shows the effects of both simultaneously.		When enabled, a icon displays in the status bar.
Projection Settings > *Painting* | **PaintingMode**	dropdown	The blending mode used when the paint bakes down onto the channel surface. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.		
Projection Settings > *Painting* | **PaintingOpacity**	text field, slider, reset	How opaque the paint is. This is a multiplier on the paint buffer contents. At 1.0, the paint bakes down to the surface with the same opacity as it is in the buffer. At 0.5, the paint applied to the surface is half as transparent as when it's in the buffer.		
Projection Settings > *Projection* | **ProjectionFilter**	dropdown	The image filtering to use when calculating how to apply the paint buffer onto the		

Control	Type	What it does	Opens this dialog box	Notes
		surface. One of: • **Nearest** - preserves edge detail, but gives quite "blocky" textures. • **Bilinear** - gives good results, but can produce square artifacts at extreme zoom.		
Projection Settings > *Projection* | **BakeBehavior**	dropdown	One of: • **Manual** - you have to bake manually. After a bake, the paint stays in the paint buffer until you manually clear it. • **ClearOnly** - you have to bake manually. After a bake, the paint buffer clears automatically. • **AutoBakeAndClear** - Mari bakes, and then clears the paint buffer, whenever you move the model.		
Projection Settings > *Projection* | **Projection**	dropdown	One of: • **Front** - paint only applies to the front of the model, as it is visible in the current view. • **Through** - paint applies to all surfaces on the model under your brush, whether or not there is something in front of them.		
Projection Settings > *Projection*	dropdown	Either:		

Control	Type	What it does	Opens this dialog box	Notes
| **ProjectOn**		• **All** - you can paint on all areas on your model(s), selected or not. • **SelectedOnly** - you can only paint on the currently selected areas.		
Projection Settings > **EdgeMask**	toggle	Whether the edge mask is turned on or not. If this is on, but **MaskPreviewEnabled** is off, the model is masked but you won't see the masking. That is, you can see the mask's effects, but not the mask itself.		When enabled, a icon displays in the status bar.
Projection Settings > *EdgeMask* > *General* | **FalloffEnd**	text field, slider, reset	Controls where the edge mask ends on the model. The higher the value, the closer, more forward facing the mask finishes.		Range: 0.00 to 1.00 Reset: 0.00
Projection Settings > *EdgeMask* > *General* | **FalloffStart**	text field, slider, reset	Controls where the edge mask starts on the model. The higher the value, the more forward facing the masking starts. If this is higher than the **FalloffEnd**, areas facing the view are paintable, and areas oblique to the view are masked. If this is lower, facing areas are masked and oblique areas are paintable.		Range: 0.00 to 1.00 Reset: 0.00
Projection Settings > **ChannelMask**	toggle	Whether the channel mask is turned on or not. If this is on,		When enabled, a

Control	Type	What it does	Opens this dialog box	Notes
		but **MaskPreviewEnabled** is off, the model is masked but you won't see the masking. That is, you can see the mask's effects, but not the mask itself.		icon displays in the status bar.
Projection Settings > *ChannelMask* > *General* | **MaskChannel**	dropdown	Sets the channel holding the mask for the paint buffer. This mask sets the paintable area within the buffer.		
Projection Settings > *ChannelMask* > *General* | **MaskAmount**	text field, slider, reset	The amount that the mask affects the paint buffer. 1.0 means that the mask absolutely controls where you can paint; the effects of the mask decrease as the value gets lower.		
Projection Settings > *ChannelMask* > *General* | **MaskContrast**	text field, slider, reset	The contrast applied to the original channel to derive the mask. At 1.0, the mask comes directly from the original channel. Lower values make the mask fuzzier than the input channel, higher values make it sharper than the input channel.		
Projection Settings > *ChannelMask* > *General* | **Mask Curve**	curve editor, reset	Sets how the channel masking works. The x axis is the color of the source pixel (black to the left, white to the right). The y axis is how paintable the target pixel is.		

Control	Type	What it does	Opens this dialog box	Notes
		By default, the curve runs from the bottom at the left up to the top at the right - so black pixels are totally masked, white pixels are totally paintable, and there's a constant curve between the two. Use this if you want to change which pixel values are masked.		
Projection Settings > *ChannelMask* > *General* | **Invert**	check box	Whether to invert the input from the mask. If this is selected, the mask data is black on white rather than white on black.		
Projection Settings > **Ambient Occlusion Mask**	toggle	Whether the ambient occlusion mask is turned on or not. If this is on, but **MaskPreviewEnabled** is off, the model is masked but you won't see the masking. That is, you can see the mask's effects, but not the mask itself. Before using this, you should calculate the ambient occlusion for each object in the project (**Objects** menu, select **AmbientOcclusion**).		When ambient occlusion masking is enabled, a icon displays in the status bar.
Projection Settings > *AmbientOcclusionMask* > *General* | **Mask Amount**	text field, slider, reset	The amount of ambient occlusion masking to apply.		

Control	Type	What it does	Opens this dialog box	Notes
Projection Settings > *AmbientOcclusionMask* > *General* | **MaskContrast**	text field, slider, reset	The contrast applied to the ambient occlusion to derive the mask. At 1.0, the mask comes directly from the calculated ambient occlusion. Lower values make the mask fuzzier than the ambient occlusion, higher values make it sharper.		
Projection Settings > *AmbientOcclusionMask* > *General* | **MaskCurve**	curve editor, reset	Controls how the ambient occlusion pixel values relate to the degree of masking. The horizontal axis is the darkness of the occluded pixels (black to the left, white to the right), the vertical axis is the degree of visibility. By default, this is a direct line, where white pixels are totally masked, black pixels are totally unmasked, and 50% gray is 50% masked.		
Projection Settings > *AmbientOcclusionMask* > *General* | **Invert**	checkbox	If this is selected, the mask data is black on white rather than white on black.		
Projection Settings > **DepthMask**	toggle	Whether the depth mask is turned on or not.		When enabled, a icon displays in the status bar.

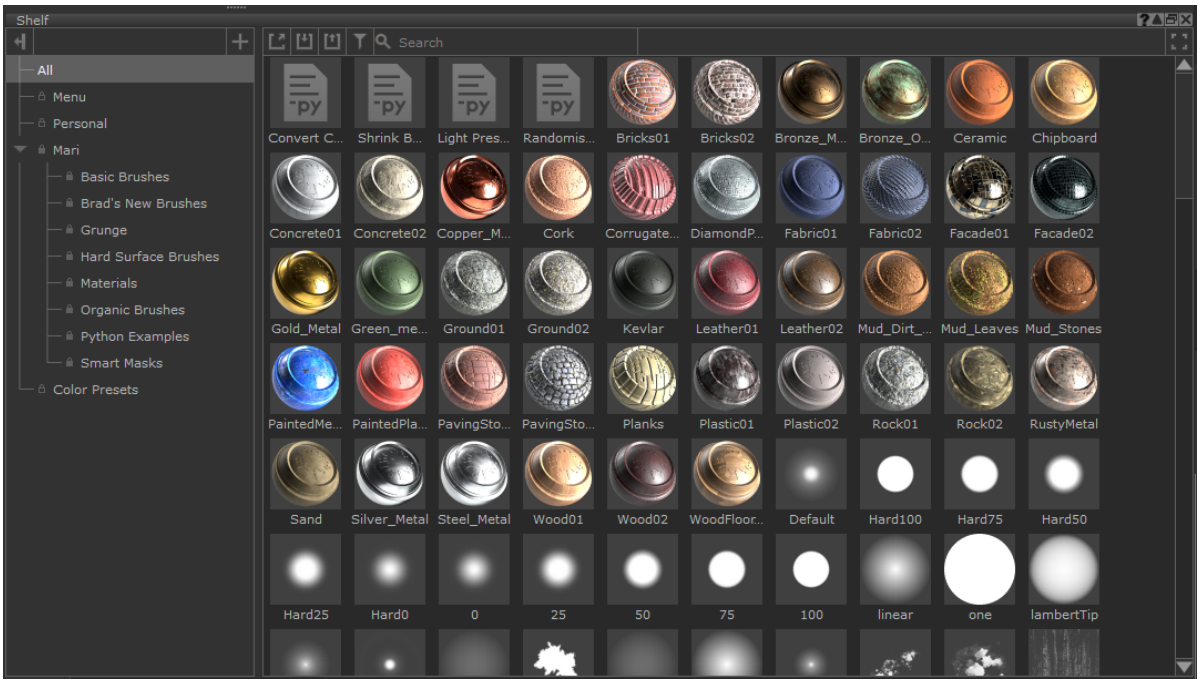
Control	Type	What it does	Opens this dialog box	Notes
Projection Settings > *DepthMask* > *General* | **Start**	text field, slider, reset	The depth in the scene for the depth masking to start.		
Projection Settings > *DepthMask* > *General* | **End**	text field, slider, reset	The depth in the scene for the depth masking to finish.		
Projection Settings > *DepthMask* > *General* | **Falloff**	text field, slider, reset	How fast the depth masking applies. Lower values make the mask apply slowly, higher values make it apply more quickly.		
Projection Settings > **BackfaceMask**	toggle	Whether the backface mask is turned on or not.		When enabled, a icon displays in the status bar.
Projection Settings > **Fractal Noise Mask**	toggle	Whether the fractal noise mask is turned on or not.		When enabled, a icon displays in the status bar.
Projection Settings > *Fractal Noise Mask* > *General* | **Seed**	text field, slider, reset	Changes the pattern of the noise by moving the pattern through 3D space.		
Projection Settings > *Fractal Noise Mask* > *General* | **Contrast**	text field, slider, reset	Controls the level of contrast applied to the fractal noise features to derive the mask.		
Projection Settings > *Fractal Noise Mask* > *General* | **Roughness**	text field, slider, reset	Determines the roughness of the fractal noise features.		

Control	Type	What it does	Opens this dialog box	Notes
Projection Settings > *Fractal Noise Mask* > *General* | **Size**	text field, slider, reset	Determines the size of the fractal noise features.		
Projection Settings > *Fractal Noise Mask* > *General* | **Mask Amount**	text field, slider, reset	The amount that the mask affects the paint buffer. 1.0 means that the mask absolutely controls where you can paint; the effects of the mask decrease as the value gets lower.		
ProjectionSettings > *Fractal Noise Mask* > *General* | **Mask Curve**	curve editor, reset	Controls how the fractal noise relates to the degree of masking, based on the above fields. This allows you to fine tune the contrast and pattern appearance of the fractal noise mask.		
Projection Settings > *Fractal Noise Mask* > *General* | **Invert**	checkbox	If this is selected, the mask data is black on white rather than white on black.		

Shelf Palette

The **Shelf** palette stores sets of brushes, colors, materials, smart masks, python scripts, and images. This allows you to quickly access your favorite presets, and store external presets from other artists for use in your own work.

There are many shelf presets that Mari comes built with, however you can also add your own items to the **Shelf** palette. To open the **Shelf** palette, click **Shelf** in the palettes toolbar.



The **Shelf** palette, with shelves listed in the left panel, and shelf items in the right panel.

How is the Shelf Palette Organised?

The **Shelf** palette contains the following default shelves:

- **All** - displays all the available items from all available shelves.
- **Menu** shelf - items that you can select from the **F9** pie selection control menu.
- **Personal** shelf - customized items you use regularly.
- **Project** shelf - stores any items you place there while the current project is in use. If you open another project, those items won't be available.

The **Shelf** palette also contains the following shelves under the **Mari** shelf, but you can also add your own:

- **Basic Brushes** shelf - a set of predefined brushes for painting basic shapes, such as the soft round and hard round brushes.
- **Brad's New Brushes** - a set of predefined brushes created by Bradford DeCaussin, which can be used to add small detail to your textures. For example, use the **moleAttack** brush to add spots and blemishes to skin.
- **Grunge** - a set of high-quality grunge textures created by industry professionals.
- **Hard Surface Brushes** shelf - a set of predefined brushes designed to paint hard surface textures such as rust, cracks, and scratches.
- **Materials** - a set of predefined materials based on the Principled BRDF shader model. See Managing Materials in Mari for more information.

- **Organic Brushes** shelf - a set of predefined brushes for painting organic textures such as skin, wrinkles, and clouds.
- **Python Examples** - predefined python scripts that perform certain useful functions.
- **Smart Masks** shelf - ready-to-use masks and grunge maps, and a set of utilities to edit them. See Smart Mask Node for more information.
  Note: this shelf includes items which used to be called Custom Procedurals.
- **Color Presets** shelf - stores colors that you have saved.

Tip: For an overview of all the ready-to-use content in the shelf, see Ready-to-use Shelf Content.

Note: To load Mari without the **Mari Materials**, **Basic Brushes**, **Hard Surface Brushes**, **Organic Brushes**, and **Brad's New Brushes** preset shelves, disable the **Edit** > **Preferences** > **Misc.** > **Shelf** > **Shelves Include Presets** preference. See Mari Preferences Dialog.

Tip: You can add any function to the Mari **Shelf** from the **Manage Keyboard Shortcuts** window. See Adding any Function to a Shelf.

Tip: Hover over any item in the shelf to see a larger thumbnail with more information about that item.

Shelf Palette Functions

Control	What it does
/ Hide shelf panel	Hide or expands the shelf panel.
Add shelf	Creates a new shelf. See Creating a Custom Shelf for more information.
Open shelf as floating window	Opens a floating window for the selected shelf. See Creating a Floating Shelf for more information.

Control	What it does
Import shelf item	Imports items into the selected shelf. See Importing and Exporting Shelves and Items for more information.
Export shelf item	Exports items from the shelf as an **.msi** file. See Importing and Exporting Shelves and Items for more information.
Resize shelf item thumbnail	Opens a slider to resize the thumbnails in the shelf.
Filter shelf items	Filter items in the shelf by type. The available types are **Materials**, **Colors**, **Brushes**, **Images**, **Color Procedurals**, **Scalar Procedurals**, **Procedural Masks**.

Shelf Context Menu

The **Shelf** palette includes a right-click menu with options to rename, delete items, and assign shortcut keys to items in the **Personal**, **Project** or custom shelves.

The following right-click options are available when hovering over the shelf panel of the **Shelf**:

Add New Shelf	Creates a new shelf. See Creating a Custom Shelf for more information.
Delete Shelf	Deletes the selected shelf.
Rename Shelf	Renames the selected shelf.
Group Shelf	Groups the selected shelves under a parent hierarchy of your choosing. See Grouping Shelves in the Shelf for more information.
Ungroup Shelves	Ungroups the selected shelf from its current group. See Grouping Shelves in the Shelf for more information.
Load Shelf	Allows you to import a previously exported Mari shelf file (**.msh**). See Importing and Exporting Shelves and Items for more information.
Save Shelf	Allows you to save the selected shelf as an **.msh** file. See Importing and Exporting Shelves and Items for more information.

The following right-click options are available when hovering over the item panel of the **Shelf**:

Load Items	Load items into the shelf with any of the following formats: **.mpc**, **.msi**, **.msh**, **.mma**. See Importing and Exporting Shelves and Items for more information. **Note:** Preset shelves are locked and you cannot import items into them. You receive a **Shelf is locked** message if you try to import items into preset shelves.
Save Items	Save shelf items as **.msi** files that can be imported to other user's shelves. See Importing and Exporting Shelves and Items for more information.
Delete Items	Removes the selected item from the shelf.
Assign Shortcuts	Assigns a keyboard shortcut of your choosing to the selected shelf item. See Customizing General Keyboard Shortcuts for more information.
Edit Tags	Edit the tags assigned to the shelf item. See Exporting a Material Node and Exporting Smart Masks for more information on tags.
Rename Item	Allows you to provide a new name for the shelf item.

Snapshots Palette

The **Snapshots** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

What It Does

The **Snapshots** palette displays:

- a list of snapshots for the entire project or currently selected channel
- buttons to take, revert, or delete channel snapshots
- information about snapshots, such as name, when they were created and by what user.

Snapshots Palette Fields

Control	Type	What it does	Opens this dialog box	Notes
Channels > **All**	button	Displays a list of all snapshots in the project.		
Channels > **Current**	button	Displays a list of snapshots for the currently selected channel.		
Created	info	The date and time when the snapshot was taken.		
Preview	thumbnail	A thumbnail preview of the snapshot.		
Name	info	The name of the snapshot.		
User	info	The name of the user that created the snapshot.		
New	button	Takes a new snapshot. Depending on whether Channels > **All**, or Channels > **Current** is selected, this button behaves differently. If **All** is selected, a snapshot is taken for all channels in a project. If **Current** is selected, a snapshot is	Snapshot All Channels. Snapshot Channel.	

Control	Type	What it does	Opens this dialog box	Notes
		only taken for the current channel.		
Extract	button	Extracts the snapshot into a new channel in the **Channels** palette and populates the **Layers** palette with the layers of the snapshot.	Extract Snapshot	This option is only available when the Channels > **Current** button is active.
Revert	button	Loads the selected snapshot.		
Delete	button	Deletes the selected snapshot.	Delete Snapshot	

Modo Render Palette

The **Modo Render** palette and the controls that can be found on it are described in the table below. When controls also bring up additional dialogs, these are referenced for more information.

Article: Modo's Command Line Render Engine is not bundled with Mari 4.6v3 on macOS. Please consult the following Knowledge Base Article on how to connect Mari with a separate Modo installation.

What It Does

The **Modo Render** palette allows you to render a final image of your scene in Modo with your Mari channels mapped to a Modo shader, bake into Mari channels, preview render and bake, and save out the Mari scene

as an **.lxo** file.

Modo Render Palette Fields

Control	Type	What it does	Dialog	Notes
Render Tab				
Render Preset	dropdown	The render preset that specifies the render settings used. If you don't yet have any presets, you can specify **None**, or choose **Basic** for Mari to adjust the render settings to a basic level.		
Channel Assignment | **+**	button	Set the channel in Mari and the specific Modo effect that you want to assign to it.	Assign Channel Dialog	
Channel Assignment | **-**	button	Remove the selected channel assignment.		
Mari Channel	information	The assigned Mari channel.		
Modo Effect	information	The assigned Modo effect.		
Preview	button	Preview the final render in the **Preview** portion of the **Modo Render** palette, in either a **Flattened** or **Not Flattened** state. The preview is sent from Modo.		
Flattened/Not Flattened	dropdown	When **Flattened**, the assigned channels are flattened and sent to Modo. Modo sends back the preview. If **Not Flattened**, the assigned channels are sent to Modo as they are and Modo then sends back the preview.		
	button	Prevents Mari from sending camera updates to Modo and, thus, the preview from updating until it is toggled again.		

Control	Type	What it does	Dialog	Notes
Render Dimensions	text field	The width and height of the exported image in pixels.		
Flattened/Not Flattened	dropdown	When **Flattened**, the assigned channels are flattened and sent to Modo. Modo sends back the render. If **Not Flattened**, the assigned channels are sent to Modo as they are and Modo then renders out the image.		
Save Render to	text field, file browser	The location to which the rendered image is saved. Modo saves the rendered image in a **.png** or **.exr** file		
Save LXO to	text field, file browser	The location to which the Modo scene representing your Mari scene is saved as an **.lxo** file.		
Render	button	Render the scene, in either a **Flattened** or **Not Flattened** state, with Mari channels assigned to Modo shader effects. Mari sends channels and other data to Modo, then Modo renders the scene to an image file specified in the **Save Render to** field. The Modo scene representing the Mari scene is saved out to an **.lxo** specified in **Save LXO to** field. Also, the preview sent from Modo is shown in the **Preview** pane of the **Modo Render** palette.		
Bake Tab				
Bake Preset	dropdown	Select the specific preset you want to bake out. Choose from the Mari defaults or custom presets defined in **.lxo** files. The built-in Modo bake presets are:		

Control	Type	What it does	Dialog	Notes
		• **AO** • **Corner Rust** • **Curvature** • **DownSlope** • **Reflection** • **RT Curvature** • **Thickness** • **Up** • **Wood**		
Name	information	The parameter name for the component.		
Value	information	The value of the corresponding parameter. This is editable within the table, and updates the same parameters within Modo.		
Preview |	button	Prevents Mari from sending camera updates to Modo and, thus, the preview from updating until it is toggled again. This is the same as the preview in the **Render** tab, and the toggled state in one tab matches that in the other.		
Bake Resolution	dropdown	Specify the resolution of the baked texture from the following options: • **256 x 256** • **512 x 512** • **1024 x 1024** • **2048 x 2048** • **4096 x 4096** • **8192 x 8192** • **1624 x 1634** • **32768 x 32768**		

Control	Type	What it does	Dialog	Notes
Bake Depth	dropdown	Specify the channel depth from the following options: • **Byte (8 bit)** • **Half (16 bit)** • **Float (32 bit)**		
Bake	button	Bakes the result (for example, ambient occlusion) out from an **.lxo** file to a channel in Mari to use in your project.		
Settings Tab				
Use	dropdown	Specify whether to use the **Default Modo** - a bundled version of Modo (command line) that ships with Mari - or a copy of a **Locally Installed Modo**.		
Modo Bin	text field	If you chose to use a local copy of Modo, specify the location file path for Mari to find the application.		
Command Port	text field	Sets the port number through which Mari sends commands to Modo. The default port number is 12377.		
Preview Port	text field	Sets the port number through which Mari receives previews from Modo. The default port number is 12377.		

Control Functions

Learn about Mari functions. These pages explain how functions work, what they do, and how to access them. Mari functions can be accessed from the menubar, the right-click menu, or the use of Appendix A: Shortcuts.

Use the left menu to see the details for Mari's control functions.

Project Functions

A list of the functions used on the project level, such as shortcuts and menu items to open or close a project, can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
New project	• *Menubar* | **File** > **New** • *Keyboard* | **Ctrl**/**Cmd**+**N** • *Projects tab* | **New** button • *Projects tab* | *Right-click* > **New** • *Toolbar* |	Creates a new project.	New Project Dialog	Name the project, select options, and add geometry.
Open a project	• *Projects tab* | *Right-click* > **Open** • *Projects tab* | *Double-click*	Opens the selected project (the project you right-clicked on).		
Open recent project	• *Menubar* | **File** > **Open** > <project name>	Opens a project, from a list of the last 10 projects opened.		
Export a session	• *Menubar* | **File** > **Session** > **Export Session**	Exports your current session of Mari, including specific objects, colorspace, channels, patches, lights, projectors, subdivision, Ambient Occlusion, and so on, within it to an **.msf** file that others can import into their Mari session.		

Function	How to access it	What it does	Opens this dialog box	Notes
Import a session	• *Menubar* | **File** >**Session** > **Import Session**	Imports a session, either as a new project or into an existing project, from an **.msf** file.		
Import channels	• *Menubar* | **File** > **Session** > **Import Channels**	Imports specific channels from an **.msf** file into your existing project.		
Import shaders	• *Menubar* | **File** > **Session** > **Import Shaders**	Imports specific shaders from an **.msf** file into your existing project.		
Import Node Graphs	*Menubar* | **File** > **Session** > **Import Node Graphs**	Imports specific Node Graphs from an **.msf** file into your existing project.		
Rename project	• *Menubar* | **File** > **Rename**	Renames the current project. This doesn't change the file name, only the name of the project in the **Projects** tab.	**Rename Current Project**	
Adjust project colorspace settings	• *Menubar* | **File** > **Settings**	Set the colorspace options for the project, changing it on a global level.	Project Settings Dialog	
Save project	• *Menubar* | **File** > **Save** • *Keyboard* | **Ctrl/Cmd+S** • *Toolbar* |	Saves the current project.		
Close project	• *Menubar* | **File** > **Close** • *Keyboard* | **Ctrl/Cmd+W**	Closes the current project.	**Save Changes Before Closing?**	

Function	How to access it	What it does	Opens this dialog box	Notes
	• *Toolbar* |		(if unsaved changes)	
Quit Mari	• *Menubar* | **File** > **Quit** • *Keyboard* | **Ctrl**/**Cmd**+**Q**	Closes the current project and Mari.	**Save Changes Before Closing?** (if unsaved changes)	
Delete project	• *Projects tab* | **Delete** button • *Projects tab* | *Right-click* > **Delete**	Deletes the selected project from disk.		
Open archive	• *Projects tab* | **Open Archive** button • *Projects tab* | *Right-click* > **Open Archive**	Opens a project from an archive file.	Import Archive	
Archive project	• *Projects tab* | **Archive** button • *Projects tab* | *Right-click* > **Archive**	Saves the selected project to disk as a Mari archive (**.mra**) file.		The archive includes all the supporting files required for the project.
Copy (duplicate) project	• *Projects tab* | **Duplicate** button • *Projects tab* | *Right-click* > **Duplicate**	Creates a copy of the selected project. The new copy appears in the **Projects** tab.		

Preference Functions

The functions for how to access and edit preferences, toolbars, and shortcuts can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Set preferences	• *Menubar* | **Edit** > **Preferences**	Lets you set general preferences for Mari - cache size, plug-ins, autosave frequency, and so on.	Mari Preferences Dialog	
Edit toolbars	• *Menubar* | **Edit** > **Toolbars**	Lets you create and edit custom toolbars.	Manage Toolbar Dialog	
Edit shortcuts	• *Menubar* | **Edit** > **Shortcuts**	Lets you add or edit keyboard shortcuts.	Manage Keyboard Shortcuts Dialog	
Configure Heads-Up Display	• *Menubar* | **Edit** > **HUD Manager**	Lets you set what information Mari shows in the Heads-Up Display (HUD) behind the model on the canvas.	HUD Manager Dialog	

Selection Functions

A list of functions for selecting items in Mari can be found in the table below, this includes the selection of objects, patches, and faces, as well as how to access selection modes.

Function	How to access it	What it does	Opens this dialog box	Notes
Object selection mode	• **Project Controls** toolbar | • *Menubar* | **Selection** >	Sets Mari to select whole objects.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Object Mode • Right-click | canvas > **Object Mode**			
Patch selection mode	• **Project Controls** toolbar | • *Menubar* | **Selection** > **Patch Mode** • *Right-click* | canvas > **Patch Mode**	Sets Mari to select patches on objects.		
Faces selection mode	• **Project Controls** toolbar | • *Menubar* | **Selection** > **Face Mode** • *Right-click* | canvas > **Face Mode**	Sets Mari to select areas on the objects.		Create a smart selection in **Face** mode by double-clicking on the face of the model. This creates a smart selection (to select faces connected in UV) based on the settings in the **Type** dropdown menu.
Hide unselected areas	• *Menubar* | **Selection** > **Hide Unselected** • *Right-click* | canvas > **Visibility** > **Hide Unselected** • *Right-click* | Patches Palette > **Hide Unselected** • *Keyboard* | **Shift**+**H** • *Mouse* |	Hides the unselected areas on the canvas.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gesture onscreen.)

Function	How to access it	What it does	Opens this dialog box	Notes
Hide selected areas	• *Menubar* | **Selection** > **Hide Selected** • *Right-click* | canvas > **Visibility** > **Hide Selected** • *Right-click* | Patches Palette > **Hide Selected Patches** • *Keyboard* | **H** • *Mouse* |	Hides the selected areas on the canvas.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)
Show selected areas	• *Menubar* | **Selection** > **Show Selected** • *Right-click* | canvas > **Visibility** > **Show Selected** • *Right-click* | Patches Palette > **Show Selected Patches**	Shows the selected areas on the canvas.		
Show all areas	• *Menubar* | **Selection** > **Show All** • *Right-click* | canvas > **Visibility** > **Show All** • *Right-click* | Patches Palette > **Show All Patches** • *Keyboard* | **Ctrl**/**Cmd**+**Shift**+**H** • *Mouse* |	Displays all areas, selected or not, on the canvas.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)
Show whole	• *Menubar* | **Selection** >	Shows all area,		

Function	How to access it	What it does	Opens this dialog box	Notes
object	**Show Entire Object** • *Right-click* | canvas > **Visibility** > **Show Entire Object**	selected or not, on only the selected object.		
Invert the visibility on all objects	• *Menubar* | **Selection** > **Invert Visibility** • *Right-click* | canvas > **Visibility** > **Invert Visibility**	Inverts the visibility on all objects in the scene based on the current selection mode: Object, Patch, or Faces selection mode.		The **Invert Visibility** option works per selection mode. For instance, if you have hidden some, or all the faces or patches in an object, then using the **Invert Visibility** option in a different selection mode on the object, has no effect on the hidden areas of the object. The invisible patches, faces, or objects remain hidden until you make them visible in the mode they were hidden.
Select all areas	• *Menubar* | **Selection** > **Select All** • *Right-click* | canvas > **Selection** > **Select All** • *Right-click* | Patches Palette > **Select All Patches** • *Button* | Patches Palette > • *Keyboard* | **Ctrl**/**Cmd**+**A**	Selects all areas in the model, depending on your current selection mode (objects, patches, or faces).		

Function	How to access it	What it does	Opens this dialog box	Notes
De-select all areas	• *Menubar* | **Selection** > **Select None** • *Right-click* | canvas > **Selection** > **Select None** • *Right-click* | Patches Palette > **Select None** • *Keyboard* | **Ctrl**/**Cmd**+**E**	De-selects all areas on the model.		
Invert selection of areas	• *Menubar* | **Selection** > **Select Invert** • *Right-click* | canvas > **Selection** > **Select Invert** • *Right-click* | Patches Palette > **Select Invert Patches** • *Keyboard* | **Ctrl**/**Cmd**+**I**	De-selects the current selection and selects everything currently unselected.		Depends on your current selection mode (either objects, patches, or faces) for the scope of inversion. For example, in face mode, the unselected faces become selected and the other way around.
Select visible areas	• *Menubar* | **Selection** > **Select Visible** • *Right-click* | canvas > **Selection** > **Select Visible** • *Right-click* | Patches Palette > **Select Visible Patches**	Selects only areas that aren't hidden.		
Fill patches with black	• *Menubar* | **Selection**> **Fill** > **Black** • *Right-click* | canvas > **Fill** > **Black**	Fills the patch with black paint.		
Fill patches with gray	• *Menubar* | **Selection** > **Fill** > **Gray**	Fills the patch with gray paint.		

Function	How to access it	What it does	Opens this dialog box	Notes
	• *Right-click* | canvas > **Fill** > **Gray**			
Fill patches with white	• Menubar | **Selection** > **Fill** > **White** • Right-click | canvas > **Fill** > **White**	Fills the patch with white paint.		
Fill patches with foreground color	• *Menubar* | **Selection** > **Fill> Foreground** • *Right-click* | canvas > **Fill** > **Foreground**	Paints the selected patch(es) with the current foreground color.		
Fill patches with background color	• *Menubar* | **Selection** > **Fill** > **Background** • *Right-click* | canvas > **Fill** > **Background**	Paints the selected patch(es) with the current background color.		
Fill patches with transparency	• *Menubar* | **Selection** > **Fill** > **Transparent** • *Right-click* | canvas > **Fill** > **Transparent**	Makes the selected patch(es) transparent.		
Fill patches with a UV mask	• Menubar | **Selection** > **Fill** > **UV Mask** • Right-click | canvas > **Fill** > **UV Mask**	Fills the selected patches with a black and white UV mask for the current layer's current paint component, if paintable.		
Fill patches with wireframe	• Menubar | **Selection** > **Fill** > **Wireframe** • Right-click | canvas >	Fills the selected patches with a wireframe for the		

Function	How to access it	What it does	Opens this dialog box	Notes
	Fill > **Wireframe**	current layer's current component, if paintable.		
Fill selected Ptex faces with foreground color	• *Menubar* | **Selection** > **Fill Ptex Faces Foreground** • *Button* | Ptex Toolbar > • Right-click | canvas > **Fill** > **Fill Ptex Faces Foreground**	Fill the selected Ptex faces with the foreground color.		
Fill selected Ptex faces with background color	• *Menubar* | **Selection** > **Fill Ptex Faces Background** • *Button* | Ptex Toolbar > • Right-click | canvas > **Fill** > **Fill Ptex Faces Background**	Fill the selected Ptex faces with the background color.		
Lock unselected areas	• Menubar | **Selection** > **Lock Unselected** • Right-click | canvas > **Locking** > **Lock Unselected** • Right-click | Patches Palette > **Lock Unselected Patches**	Locks unselected patches so they cannot be edited.		
Lock selected areas	• *Menubar* | **Selection** > **Lock Selected**	Locks selected areas, so they cannot be		

Function	How to access it	What it does	Opens this dialog box	Notes
	• *Right-click* | canvas > **Locking** > **Lock Selected** • *Right-click* | Patches Palette > **Lock Selected Patches** • *Button* | Patches Palette >	painted.		
Unlock selected areas	• *Menubar* | **Selection** > **Unlock Selected** • *Right-click* | canvas > **Locking** > **Unlock Selected** • *Right-click* | Patches Palette > **Unlock Selected Patches** • *Button* | Patches Palette >	Unlocks selected locked areas, so they are again paintable.		
Lock all areas	• *Menubar* | **Selection** > **Lock All** • *Right-click* | canvas > **Locking** > **Lock All** • *Right-click* | Patches Palette > **Lock All Patches**	Locks all areas, so they cannot be painted.		
Unlock all areas	• *Menubar* | **Selection** > **Unlock All** • *Right-click* | canvas > **Locking** > **Unlock All** • *Right-click* | Patches Palette > **Unlock All Patches**	Unlocks all locked areas, so they are again paintable.		

Function	How to access it	What it does	Opens this dialog box	Notes
Unlock everything	• *Menubar* | **Selection** > **Unlock Entire Object** • *Right-click* | canvas > **Locking** > **Unlock Entire Object**	Unlocks everything on the current object, regardless of whether it's a face, patch, or channel.		
Create selection group for patches	• *Right-click* | canvas > **Create Patch Selection Set**	Creates a new selection set consisting of whichever patches you currently have selected.		Creates a new selection set in the Patches Palette
Create selection group	• *Right-click* | canvas > **Create Selection Group** • *Button* | Selection Groups Palette >	Creates a new selection group to save your selection modes.		You can create selection groups in any selection mode (objects, patches, or faces).
Select selection group	• *Right-click* | selection group in Selection Groups Palette > **Select Selection Group** • *Button* | Selection Groups Palette >	Selects the whole selection group in the canvas and highlights the group in the palette.		
Toggle selected selection groups	• When the **Select** tool is set to **Smart Selection** mode, and the **Type** is **Selection Group**, repeatedly click on a face in the canvas.	If a face belongs to more than one selection group, clicking repeatedly cycles through the other selection groups to which it belongs.		
Lock the	• *Right-click* | selection	Locks the selection		

Function	How to access it	What it does	Opens this dialog box	Notes
selection group	group in Selection Groups Palette > **Lock Selection Group** • *Button* | Selection Groups Palette >	group.		
Unlock the selection group	• *Right-click* | selection group in Selection Groups Palette > **Unlock Selection Group** • *Button* | Selection Groups Palette >	Unlocks the selection group.		
Hide the selection group	• *Right-click* | selection group in Selection Groups Palette > **Hide Selection Group** • *Button* | Selection Groups Palette >	Hides the contents of the selection group.		
Assign a shortcut	*Right-click* | selection group in Selection Groups Palette > **Assign Shortcut**	Opens the **Assign Keyboard Shortcut** dialog and allows you to set a shortcut for the selected object.		
Show the selection group	• *Right-click* | selection group in Selection Groups Palette > **Show Selection Group** • *Button* | Selection Groups Palette >	Shows the contents of the selection group.		

Function	How to access it	What it does	Opens this dialog box	Notes
Remove the selection group	• Right-click | selection group in Selection Groups Palette > **Remove Selection Group** • Button | Selection Groups Palette >	Removes the contents of the selection group.		
Add selection groups together	• **Shift**+click a selection group in Selection Groups Palette when another selection group is already selected.	Adds the two selection groups together on the canvas to show the visibility of both selections at once.		This does not create a new selection group automatically; it only adjusts the visible selections on the canvas.
Subtract selection groups from one another	• **Ctrl**+click a selection group in Selection Groups Palette when another selection group is already selected.	Subtracts one selection group from another on the canvas to show the visibility of a selection without the faces or patches of the other.		This does not create a new selection group automatically; it only adjusts the visible selections on the canvas.

Object Functions

A list of functions for adding, editing, moving, or removing objects can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Add object	• *Menubar* | **Objects** > **Add Object** • *Right-click* | Objects	Adds a new object to the project.	Add Object Settings Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
	Palette > **Add Object** • *Button* | Objects Palette >			
Add child object	• *Right-click* |an object in the Objects Palette > **Add Child Object**	Adds a new object as a child object to the object hierarchy.	Add Object Settings Dialog	
Add locator	• *Right-click* | Objects Palette > **Add Locator** • *Button* | Objects Palette >	Adds a locator so that you can rotate, scale, and move objects in 3D world/camera space.		
Add child locator	• *Right-click* |an object in the Objects Palette > **Add Child Locator**	Adds a locator as a child object so that you can rotate, scale, and move objects in 3D world/camera space.		
Remove object	• *Menubar* | **Objects** > **Remove Object** • *Right-click* | an object in the Objects Palette > **Remove Object** • *Button* | Objects Palette >	Removes the selected object from the project.		Projects must always have at least one object.
Duplicate object	• *Menubar* | **Objects** > **Duplicate** > **Object Only** or **Object And Shader Network** • *Right-click* | an object in the Objects Palette > **Duplicate** > **Object**	Duplicates either the object on its own or the object and the associated shader network that belongs to it.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Only or **Object And Shader Network**			
Add object version	• *Right-click* | an object in the Objects Palette > **Add Version**	Lets you select a file containing the new object version data.	Add Version Settings Dialog	Mari preserves selection groups between object versions. See Adding and Removing Object Versions.
Rename object version	• *Right-click* | an object in the Objects Palette > **Rename Version**	Renames the current object version.	Rename Version Dialog	
Remove object version	• *Right-click* | an object in the Objects Palette > **Remove Version**	Deletes the current object version.		Objects must always have at least one version.
Calculate ambient occlusion	• *Menubar* | **Objects** > **Ambient Occlusion** • *Right-click* | an object in the Objects Palette > **Ambient Occlusion**	Calculates the ambient occlusion for the object. This is required for any operations that use the ambient occlusion data (such as using ambient occlusion masking).		This can take a long time, depending on the complexity of the model.
Subdivide the object	• *Menubar* | **Objects** > **Subdivide** • *Button* | **Objects palette** > **Subdivide** • *Right-click* | an object in the Objects Palette > **Subdivide**	Specify how to subdivide the geometry and to what degree so that it appears as if it were a higher polygon model.	Subdivide Dialog	Subdivision can run in the foreground or background. See *Subdividing Objects* in the *Mari User Guide* for more information.
Generate Gaussian Blur	• *Menubar* | **Objects** > **Generate** > **Gaussian**	Allows you to apply	Gaussian Blur Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
	Blur • *Right-click* | an object in the Objects Palette > **Generate** > **Gaussian Blur**	an object-based blur effect across the whole object, not just on specific patches. This eliminates seaming because it can apply the effect across 3D boundaries.		
Generate Displacement	• *Menubar* | **Objects** > **Generate** > **Displacement** • *Right-click* | an object in the Objects Palette > **Generate** > **Displacement**	Allows you to generate a displacement map, using a high polygon source model and a low polygon destination model to calculate the displacement.	Displacement Dialog	If you want to generate a displacement map, the channel you have selected needs to be **16-bit (Half)** or **32-bit (Float)**.
Generate Height	• *Menubar* | **Objects** > **Generate** > **Height** • *Right-click* | an object in the Objects Palette > **Generate** > **Height**	Allows you to generate a height map, using a high polygon source model and a low polygon destination model to calculate the height.	Height Dialog	The values for the height are normalized between 0 and 1.
Assign a shortcut	• *Right-click* | an object in the Objects Palette > **Assign Shortcut**	Opens the **Assign Keyboard Shortcut** dialog and allows you to set a shortcut for the selected object.	Assign Keyboard Shortcut	
Reset move	• *Mouse* | **Shift**-click on	**Shift**+click on the		

Function	How to access it	What it does	Opens this dialog box	Notes
object - single object	the object	object resets that object to its original position.		
Reset move object - all objects	• *Keyboard* | '	Pressing the ' (apostrophe) key resets all objects back to their original positions.		
Reset object transform	• *Menubar* | **Objects** > **Reset Object Transform**	Resets the object transform back to its original coordinates for the selected object or locator in the **Objects** palette.		

Channel Functions

A list of functions for adding, editing, and removing channels and channels presets can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Add channel	• *Menubar* | **Channels** > **Add Channel** • *Right-click* | in Channels Palette > **Add Channel** • *Button* | Channels Palette >	Adds a new channel to the current project.	Add Channel Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
Create channel presets	• *Menubar* | **Channels** > **Channel Presets** • *Right-click* | in Channels Palette > **Channel Presets**	Allows you to create a new channel preset to use as a template for new channels.	Channel Presets Dialog	Click the checkbox for channels you want to add, and click **OK**. You can customize available categories of channel presets using an XML file (see Defining Custom Channel Presets) in a folder specified in your preferences.
Remove channel	• *Menubar* | **Channels** > **Remove Channel** • *Right-click* | in Channels Palette > **Remove Channel** • *Button* | Channels Palette >	Removes the selected channel, and any layers associated with it, from the channels list.		
Convert channel	• *Menubar* | **Channels** > **Convert Channel** • *Right-click* | in Channels Palette > **Convert Channel** • *Button* | Channels Palette >	Changes the color depth of the channel to either **8bit (Byte)**, **16bit (Half)**, or **32bit (Float)**.	Convert Type Dialog	Determines number of pixels per RGB channel.
Duplicate channel	• Keyboard | **Alt**+drag and drop channel	Duplicates the channel and all the layers belonging to that channel.		
Flatten	• Menubar | **Channels** >	Flattens the channel		

Function	How to access it	What it does	Opens this dialog box	Notes
channel	**Flatten** • *Right-click* | in Channels Palette > **Flatten**	data and bakes down the texture.		
Channel transfer	• Menubar | **Channels** > **Transfer** • *Right-click* | in Channels Palette > **Transfer**	Transfers channels from one object or version to another.	Channel Transfer Dialog	
Cut channel	• Menubar | **Channels** > **Cut** • *Right-click* | in Channels Palette > **Cut** • *Keyboard* | **Ctrl**/**Cmd**+**X**	Cuts the channel and all associated layers from the channels list and places it in the clipboard.		
Copy channel	• Menubar | **Channels** > **Copy** • Right-click | in Channels Palette > **Copy** • Keyboard | **Ctrl**/**Cmd**+**C**	Copies the channel, and all associated layers from the channels list and places it in the clipboard.		
Paste channel	• Menubar | **Channels** > **Paste** • Right-click | in Channels Palette > **Paste** • Keyboard | **Ctrl**/**Cmd**+**V**	Pastes any cut or copied channel and associated layers from the clipboard to the channels list.		
Export current channel	• *Menubar* | **Channels** > **Export Current Channel** • *Right-click* | canvas > **Export** > **Export Current Channel** • *Right-click* | in Channels Palette > **Export Current Channel**	Exports the current channel to a series of files.	Export Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
Export all channels	• *Menubar* | **Channels** > **Export** > **Export All Channels** • *Right-click* | canvas > **Export** > **Export All Channels** • *Right-click* | in Channels Palette > **Export** > **Export All Channels**	Exports all the channels in the channels palette to a series of files.	Export Dialog	
Export everything	• *Menubar* | **Channels** > Export **> Export Everything** • *Right-click* | canvas > **Export** > **Export Everything** • *Right-click* | in Channels Palette > **Export** > **Export Everything**	Exports the channels for all the objects in your project to a series of files.	Export Dialog	
Flatten and export the current channel	• *Menubar* | **Channels** > **Export Flattened** > **Export Current Channel Flattened** • *Right-click* | canvas > **Export Flattened** > **Export Current Channel Flattened** • *Right-click* | in Channels Palette > **Export Flattened** > **Export Current Channel Flattened**	Flattens the current channel and exports it to a series of files.	Export Dialog	
Flatten and export all	• *Menubar* | **Channels** > **Export Flattened** >	Flattens all the channels in the channels palette	Export Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
channels	**Export All Channels Flattened** • *Right-click* | canvas > **Export Flattened** > **Export All Channels Flattened** • *Right-click* | Channels Palette > **Export Flattened** > **Export All Channels Flattened**	and exports it to a series of files.		
Flatten and export everything	• *Menubar* | **Channels** > **Export Flattened** > **Export Everything Flattened** • *Right-click* | canvas > **Export Flattened** > **Export Everything Flattened** • *Right-click* | Channels Palette > **Export Flattened** > **Export Everything Flattened**	Flattens the channels for all objects in your project and exports them to a series of files.	Export Dialog	
Import into a layer stack	• *Menubar* | **Channels** > **Import** > **Import into Current Channel** • Right-click | canvas > **Import** > **Import into Current Channel** • *Right-click* | in Channels Palette > **Import** > **Import into Current Channel**	Imports the file as layers into the layer stack of your current, selected channel.	Import Dialog	
Import into a	• *Menubar* | **Channels** >	Import the file as a new	Import	

Function	How to access it	What it does	Opens this dialog box	Notes
new channel	**Import** > **Import into New Channel** • *Right-click* | canvas > **Import** > **Import into New Channel** • *Right-click* | in Channels Palette > **Import** > **Import into New Channel**	channel in the Channels Palette.	Dialog	
Resize channel	• Menubar | **Channels** > **Resize** • *Right-click* | in Channels Palette > **Resize**	Resize options are: • **256x256** • **512x512** • **1024x1024** • **2048x2048** • **4096x4096** • **8192x8192** • **16384x16384** • **32768x32768**		
Lock channel	• Menubar | **Channels** > **Lock Channel** • Right-click | channel in Channels Palette > **Lock Channel** • Button | Channels Palette >	Lock the selected channel so that the channel nor any of the layers in the associated layer stack can be edited or deleted.		Lock icons in the Channels Palette show the current state of the channel.
Unlock channel	• Menubar | **Channels** > **Unlock Channel** • Right-click | channel in Channels Palette > **Unlock Channel** • Button | Channels	Unlocks the selected channel so that the channel or the layers within the layer stack can be edited.		Lock icons in the Channels Palette show the current state of the channel.

Function	How to access it	What it does	Opens this dialog box	Notes
	Palette >			
Lock all channels	• Menubar	**Channels** > **Lock All Channels** • Right-click	Channels Palette > **Lock All Channels**	Locks all the channels in the Channels Palette, regardless of what is selected.		
Unlock all channels	*Menubar*	**Channels** > **Unlock All Channels** *Right-click*	Channels Palette > **Unlock All Channels**	Unlock all the channels in the Channels Palette, regardless of what is selected.		In addition to the unlock options listed here, you can also unlock channels by selecting **Unlock Entire Object.**
View the Next Channel	• *Menubar*	**Channels** > **Next Channel** • *Right-click* | in Channels Palette > **Next Channel** • *Mouse* |	Selects the next channel in the list.		
View the Previous Channel	• Menubar	**Channels** > **Previous Channel** • Right-click | in Channels Palette > **Previous Channel** • Mouse |	Selects the previous channel in the list.		
Snapshot all channels	• *Menubar*	**Channels** > **Snapshots** > **Snapshot All Channels** • *Right-click* | in Channels Palette > **Snapshots** > **Snapshot All Channels**	Takes snapshots of all the channels in the Channels Palette.		

Function	How to access it	What it does	Opens this dialog box	Notes
Snapshot the current channel	• *Menubar* | **Channels** > **Snapshots** > **Snapshot Current Channel** • *Right-click* | in Channels Palette > **Snapshots** > **Snapshot Current Channel**	Takes snapshots of only the current selected channel.		
Manage snapshots	• *Menubar* | **Channels** > **Snapshots** > **Manage Snapshots** • *Right-click* | in Channels Palette > **Snapshots** > **Manage Snapshots**	Opens the **Snapshots** palette.		
Delete all snapshots	• Menubar | **Channels** > **Snapshots** > **Delete All Snapshots** • Right-click | in Channels Palette > **Snapshots** > **Delete All Snapshots**	Deletes all the snapshots taken for the current project.		

Layer Functions

A list of functions for adding, editing, removing, exporting, or importing layers and layer stacks can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Add new layer	• Menubar | **Layers** > **Add New Layer** • *Right-click* | in Layers Palette > **Add New**	Adds a new layer to the layer stack.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Layer • *Button* | Layers Palette >			
Add adjustment layer	• Menubar | **Layers** > **Add Adjustment Layer** • Right-click | layer in Layers Palette > **Add Layer Adjustment** • Button | Layers Palette > • Keyboard | **Tab**	Adds the selected filter as an Adjustment layer to the layer stack.		If adding adjustments by pressing **Tab**, type in the name of the adjustment you want to add, for example, **Contrast**.
Add procedural layer	• Menubar | **Layers** > **Add Procedural Layer** • *Right-click* | in Layers Palette > **Add Procedural Layer** • *Button* | Layers Palette > • *Keyboard* | **Tab**	Adds the selected **Basic**, **Projection**, **Lighting**,**Geometry**, or **Procedural** layer to the layer stack.		If adding procedurals by pressing **Tab**, type in the name of the procedural you want to add, for example, **Cloud**.
Cut layer or layer mask	• Menubar | **Layers** > **Cut** • Right-click | layer in Layers Palette > **Cut** • Menubar | **Layers** > **Layer Mask** > **Cut** • Right-click | layer mask in Layers Palette > **Layer Mask** > **Cut** • Keyboard | **Ctrl**/**Cmd**+**X**	Cuts the layer from the layers list to allow you to paste it elsewhere in the project or temporarily remove it.		Using the **Ctrl**/**Cmd**+**X** shortcut cuts the whole layer. You cannot cut only the layer mask.
Copy layer or layer mask	• Menubar | **Layers** > **Copy**	Copies the layer from the layers list to allow		Using the **Ctrl**/**Cmd**+**C** shortcut copies the

Function	How to access it	What it does	Opens this dialog box	Notes
	• Right-click | layer in Layers Palette > **Copy** • Menubar | **Layers** > **Layer Mask** > **Copy** • Right-click | layer mask in Layers Palette > **Layer Mask** > **Copy** • Keyboard | **Ctrl**/**Cmd**+**C** or **Alt**+drag	you to paste it elsewhere in the project.		whole layer. You cannot copy only the layer mask.
Paste layer or layer mask	• Menubar | **Layers** > **Paste** • Right-click | layer in Layers Palette > **Paste** • Menubar | **Layers** > **Layer Mask** > **Paste** • Right-click | layer mask in Layers Palette > **Layer Mask** > **Paste** • Keyboard | **Ctrl**/**Cmd**+**V**	Pastes a cut or copied layer into the selected location in the project's Layers Palette. Layers can be paste as a copy in the layer list, or as a new layer in a mask stack or adjustment stack.		
Duplicate layer	• *Button* | Layers Palette >	Creates an exact copy of the selected layer.		'Copy' is added to the layer name to denote which layer is the duplicate.
Remove layers	• *Right-click* | layer in Layers Palette > **Remove Layers** • *Button* | Layers Palette >	Removes the selected layer from the current project.		
Merge layers	• *Right-click* | layer in Layers Palette > **Merge Layers**	Merges two or more layers together into a single new layer.		Any masks or layer groups contained in the selection are also

Function	How to access it	What it does	Opens this dialog box	Notes
				merged into the layer and are no longer displayed as masks or groups.
Layer transfer	• Menubar | **Layers** > **Transfer** • Right-click | layer in Layers Palette > Transfer	Transfers layers from one object or version to another.	Layer Transfer Dialog	
Add empty layer group	• Right-click | in Layers Palette > **Add Layer Group** • Button | Layers Palette >	Adds an empty layer group, which allows you to add layers, masks, adjustments or procedurals within a single layer group.		These layer groups can be flattened to create a single layer.
Group layers	• *Right-click* | layer in Layers Palette > **Group Layers** • Menubar | **Layers** > **Group Layers** • **Ctrl**+**G**	Gathers multiple selected layers together under a single group.		Selected layers are located under a parent layer, which is given a 'group' name.
Ungroup layers	• *Right-click* | layer in Layers Palette > **Ungroup Layers** • Menubar | **Layers** > **Ungroup Layers** • **Ctrl**+**U**	Deletes the group of layers and re-arranges the layers while respecting their hierarchy.		You can ungroup multiple layer groups at once.
Flatten layer group	• Right-click | layer in Layers Palette > **Flatten Layer Group**	Reduces multiple layers into a single layer, where all the textures are still		

Function	How to access it	What it does	Opens this dialog box	Notes
		present.		
Add mask	• Menubar	**Layers** > **Layer Mask** > **Add Mask** • Right-click	layer in Layers Palette > **Layer Mask** > **Add Layer Mask** • Button	Layers Palette >	Adds a layer mask to the selected layer. Select **Reveal All**, **Hide All**, or to create **From Alpha**.		
Bake mask	• *Menubar*	**Layers** > **Layer Mask** > **Bake Mask** • *Right-click*	layer in Layers Palette > **Layer Mask** > **Bake Mask**	Bakes the mask into the layer, essentially flattening it.		
Resize Mask	• *Menubar*	**Layers** > **Layer Mask** > **Resize Mask** • *Right-click*	layer in Layers Palette > **Layer Mask** > **Resize Mask**	Resizes the mask to a size listed in the **Resize Mask** dropdown menu.		
Remove mask	• *Menubar*	**Layers** > **Layer Mask** > **Remove Mask** • *Right-click*	layer in Layers Palette > **Layer Mask** > **Remove Mask**	Remove the mask from the layer.		
Make mask stack	• Menubar	**Layers** > **Layer Mask** > **Make**	Creates a mask layer		Clicking the mask

Function	How to access it	What it does	Opens this dialog box	Notes
	Mask Stack • Right-click | layer in Layers Palette > **Layer Mask** > **Make Mask Stack** • Button | Layers Palette >	stack from a single mask on a layer. Requires the selected layer to have a mask, unless you are adding the mask stack using the **Add Mask Stack** button in the **Layers** palette.		stack icon opens an undocked mask stack palette.
Flatten mask stack	• Menubar | **Layers** > **Layer Mask** > **Make Mask Stack** • Right-click | layer in Layers Palette > **Flatten Mask Stack**	Reduces the masks in a mask stack back into a single mask on the layer.		
Enable mask	• Menubar | **Layers** > **Layer Mask** > **Enable Mask** • Right-click | in Layers Palette > **Layer Mask** > **Enable Mask**	Enables the mask to be viewed on the model.		
Disable mask	• Menubar | **Layers** > **Layer Mask** > **Disable Mask** • Right-click | in Layers Palette > **Layers Mask** > **Disable Mask**	Disables the mask from being viewed on the model. Disabling the mask only temporarily removes it from view.		To remove the mask entirely, simply **Remove Mask** in the same menu.
Convert Mask	• Menubar | **Layers** > **Layer Mask** > **Convert Mask** • Right-click | in Layers Palette > **Layers Mask** > **Convert Mask**	Converts the mask's texel format to either **8bit (Byte)**, **16bit (Half)**, or **32bit (Float)**. This works much the same way as		

Function	How to access it	What it does	Opens this dialog box	Notes
		converting a channel's texel format.		
Add secondary adjustment	• Menubar	**Layers** > **Secondary Adjustment** > **Add Secondary Adjustment** • Right-click	adjustment layer in Layers Palette > **Secondary Adjustment**	Allows you to add another type of adjustment to the selected adjustment layer.		This only applies to adjustment layers. The option appears grayed out if the selected layer isn't an adjustment.
Remove secondary adjustment	• Menubar	**Layers** > **Secondary Adjustment** > **Remove Secondary Adjustment** • Right-click	adjustment layer in Layers Palette > **Remove Secondary Adjustment**	Removes any secondary adjustment that has been applied to the selected adjustment layer.		This only applies to adjustment layers. The option appears grayed out if the selected layer isn't an adjustment.
Add adjustment stack	• Menubar	**Layers** > **Adjustment Stack** > **Add Adjustment Stack** • Right-click	layer in Layers Palette > **Adjustment Stack** > **Add Adjustment Stack** • Button	Layers Palette >	Adds an adjustment stack to the selected layer. Add an adjustment layer from the menu of **Filter** adjustment layers.		This only applies to layers other than adjustment layers.
Bake adjustment stack	• Menubar	**Layers** > **Adjustment Stack** > **Bake Adjustment**	Bakes an adjustment stack down, essentially flattening		

Function	How to access it	What it does	Opens this dialog box	Notes
	Stack • Right-click | layer in Layers Palette > **Adjustment Stack** > **Bake Adjustment Stack**	the stack into a single adjustment layer.		
Remove adjustment stack	• Menubar | **Layers** > **Adjustment Stack** > **Remove Adjustment Stack** • Right-click | layer in Layers Palette > **Adjustment Stack** > **Remove Adjustment Stack**	Removes any adjustment stack that has been applied to the selected layer.		This only applies to layers other than adjustment layers.
Enable adjustment stack	• Menubar | **Layers** > **Adjustment Stack** > **Enable Adjustment Stack** • Right-click | layer in Layers Palette > **Adjustment Stack** > **Enable Adjustment Stack**	If an adjustment stack has been disabled, this option re-enables the adjustment stack.		This only applies to adjustment stacks that have been disabled. If a layer does not have an adjustment stack, or if the stack is already enabled, the option appears grayed out.
Disable adjustment stack	• Menubar | **Layers** > **Adjustment Stack** > **Disable Adjustment Stack** • Right-click | layer in Layers Palette > **Adjustment Stack** > **Disable Adjustment Stack**	Disables an adjustment stack to show the paint layer result as it is without the stack. Useful for comparing the model with and without the adjustment stack.		This only applies to adjustment stacks that have been added to a layer. If a layer does not have an adjustment stack, the option appears grayed out.

Function	How to access it	What it does	Opens this dialog box	Notes
Convert to paintable	• Menubar	**Layers** > **Convert to Paintable** • Right-click	layer in Layers Palette > **Convert to Paintable**	Converts non-paintable standard layers to paintable Paint layers. Converts non-paintable multi-channel layers to a set of paintable layers across streams.		Once converted to paintable layers, the layers display the color palette icon.
Share Layer	• Menubar	**Layers** > **Sharing** > **Share Layer** • Right-click	layers in Layers Palette > **Sharing** > **Share Layer** • Shift+drag layer	Creates a shared layer in the layer stack.		
Unshare layer	• Menubar	**Layers** > **Sharing** > **Unshare Layer** • Right-click	layer in Layers Palette > **Sharing** > **Unshare Layer**	Undoes the shared state of the shared (created) layer and the original layer.		If sharing a layer as a channel, unsharing the layer to remove the shared state doesn't remove the channel from the Channels Palette. This needs to be done manually.
Share Layers As Channel	• Menubar	**Layers** > **Sharing** > **Share Layers As Channel** • Right-click	layers in Layers Palette > **Sharing** > **Share Layers As Channel**	Copies the selected layers into the Channels Palette as a new Channel. Clicking on the channel icon opens a new palette with the shared layer already listed.		
Make shared	• Menubar	**Layers** >		Makes the shared		

Function	How to access it	What it does	Opens this dialog box	Notes
channel current	**Sharing** > **Make Shared Channel Current** • Right-click | layer in Layers Palette > **Sharing** > **Make Shared Channel Current**	channel the current channel in the **Current Channel** option of the **Default Shaders** tool of the **Project Controls** toolbar.		
Cache layer (s)	• Menubar | **Layers** > **Cache Layers** • Right-click | layer in Layers Palette > **Cache Layers**	Bakes the layer to speed up layer performance. Caching a layer caches it to the hard-disk.		**Warning:** Changing colorspace settings of a channel removes any cache from all layers of that channel.

Function	How to access it	What it does	Opens this dialog box	Notes
Cache up to here	• Menubar	**Layers** > **Cache Up To Here** • Right-click	layer in Layers Palette > **Cache Up To Here**	Caches up to the current selected layer in the layer stack.		**Note: Cache Layers** and **Cache Up To Here** both cache material layers across all siblings channels. See Managing Materials in Mari for more information on materials.
Uncache layer(s)	• Menubar	**Layers** > **Uncache Layers** • Right-click	layer in Layers Palette > **Uncache Layers**	Release the cached files and unlocks the layer stack, allowing it to be edited again.		
Uncache layer up to here	• Menubar	**Layers** > **Uncache Up To Here** • Right-click	layer in Layers Palette > **Uncache Up To Here**	Releases the cached files and unlocks the layer stack up to the selected layer, allowing those layers to be edited again.		
Export selected layers	• Menubar	**Layers** > **Export** > **Export Selected Layers** • Right-click	layer in Layers Palette > **Export** > **Export Selected Layers**	Exports the selected layer to an external source.	Export Dialog	
Export all layers	• Menubar	**Layers** > **Export** > **Export All Layers** • Right-click	layer in Layers Palette > **Export** > **Export All Layers**	Exports all the layers in the layer stack to an external source.	Export Dialog	
Export selected	• Menubar	**Layers** > **Export** > **Export**		Exports the mask of the selected layer(s) or	Export Dialog	The masks for layers and layered shaders

Function	How to access it	What it does	Opens this dialog box	Notes
masks	**Selected Masks** • Right-click | layer in Layers Palette or shader in **Shader Layers** tab > **Export** > **Export Selected Masks**	layered shader(s).		behave in the same way as they are in effect the same entity.
Export all masks	• Menubar | **Layers** > **Export** > **Export All Masks** • Right-click | layer in Layers Palette or shader in **Shader Layers** tab > **Export** > **Export All Masks**	Exports all the masks in a layer stack, in either the Layers Palette or the **Shader Layers** tab of the Shaders Palette.	Export Dialog	
Export selected layers flattened	• Menubar | **Layers** > **Export Flattened** > **Export Selected layers Flattened** • Right-click | layer in Layers Palette > **Export** > **Export Selected Layers Flattened**	Exports the selected layer or layers flattened into a single file to an external source.	Export Dialog	
Export all layers flattened	• Menubar | **Layers** > **Export Flattened** > **Export All Layers Flattened** • Right-click | layer in Layers Palette > **Export** > **Export All Layers Flattened**	Exports all the layers in the layer stack flattened into a single file to an external source.	Export Dialog	
Export selected masks	• Menubar | **Layers** > **Export Flattened** > **Export Selected asks**	Exports the mask of the selected layer(s) or layered shader(s)	Export Dialog	The masks for layers and layered shaders behave in the same

Function	How to access it	What it does	Opens this dialog box	Notes
flattened	**Flattened** • Right-click | layer in Layers Palette or shader in **Shader Layers** tab > **Export** > **Export Selected Masks Flattened**	flattened into a single file to an external source.		way as they are in effect the same entity.
Export all masks flattened	• Menubar | **Layers** > **Export Flattened** > **Export All Masks Flattened** • Right-click | layer in Layers Palette or shader in **Shader Layers** tab > **Export** > **Export All Masks Flattened**	Exports all the masks in a layer stack, in either the Layers Palette or the **Shader Layers** tab of the Shaders Palette, flattened into a single file to an external source.	Export Dialog	
Import into layer	• Menubar | **Layers** > **Import** > **Import into Current Layer** • Right-click | layer in Layers Palette > **Import > Import into Current Layer**	Selects a file with paint or mask data to be imported into the selected layer.	Import Dialog	
Import into layer stack	• Menubar | **Layers** > **Import into New Layer** • Right-click | layer in Layers Palette > **Import** > **Import into New Layer**	Selects a file with paint or mask data to be imported into the current layer stack.	Import Dialog	
Inspect layer properties	• *Right-click* | layer in Layers Palette > **Layer Properties**	Opens a separate window to display the properties of the		

Function	How to access it	What it does	Opens this dialog box	Notes
		selected layer.		
Refresh layer view	• Right-click | in Layers Palette > **Refresh Layer View**	Refreshes the entire layer stack.		

Node Graph Functions

A list of functions available in the Node Graph can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Add Paint node	• *Right-click* | Node Graph > Create Paint Node • *keyboard* | **P**	Opens a dialog box allowing you to add an image set.	Add Paint/Bake Point Dialog	
Add Group node	• *Right-click* | Node Graph > Group • *keyboard* | **Ctrl/Cmd+G**	Creates a Group node containing selected nodes or an empty Group if no nodes are selected.		
Import image into Paint node	• *Right-click* | Node Graph > File > Import	Imports an image on disk into a selected Paint node in the current Node Graph.		
Export image from Paint node	• *Right-click* | Node Graph > File > Export	Exports a selected Paint node from the current Node Graph to a file on disk.		

Function	How to access it	What it does	Opens this dialog box	Notes
Import nodes into Node Graph	• *Right-click* | Node Graph > File > Import Nodes	Imports a selection of nodes into the current Node Graph from **.mng** files.		
Export nodes from Node Graph	• *Right-click* | Node Graph > File > Export Nodes	Exports a selection of nodes from the current Node Graph to an **.mng** file.		
Export and flatten Channel nodes from Node Graph	• *Right-click* | Node Graph > File > Export All Channel Nodes Flattened	Flattens each patch as a single image for all Channel nodes and exports them to a series of files.		
Export and flatten selected Channel nodes from Node Graph	• *Right-click* | Node Graph > File > Export Selected Channel Nodes Flattened	Flattens each patch as a single image for the selected Channel nodes and exports them to a series of files.		
Copy selected nodes to clipboard	• *Right-click* | Node Graph > Edit > Copy • *keyboard* | **Cmd/Ctrl+C**	Copies the selected nodes from the current Node Graph to the OS clipboard.		
Paste nodes from clipboard	• *Right-click* | Node Graph > Edit > Paste • *keyboard* | **Cmd/Ctrl+V**	Pastes the nodes from the OS clipboard to the current Node Graph.		
Duplicate selected node (s)	• *Right-click* | Node Graph > Edit > Duplicate	Creates a duplicate of the selected node(s) in the current Node Graph.		
Bake paint to	• *Right-click* | Node	Converts a selected node		

Function	How to access it	What it does	Opens this dialog box	Notes
Paint node	Graph > Edit > Bake to Paint Node	to a Paint node, allowing you to export its output.		
Rename selected node	• *Right-click* | Node Graph > Edit > Rename Node • *keyboard* | **N**	Renames the selected node.		
Disable selected node (s)	• *Right-click* | Node Graph > Edit > Disable • *keyboard* | **D**	Disables the selected node or nodes, only passing their preferred input down the node tree.		
Bypass selected node inputs	• *Right-click* | Node Graph > Edit > Bypass • *keyboard* | **Ctrl/Cmd+D**	Cycles through the node's inputs to bypass all but one.		
Delete selected node (s)	• *Right-click* | Node Graph > Edit > Delete • *keyboard* | **Del**	Deletes the selected nodes from the current Node Graph.		
Extract the node	• *Right-click* | Node Graph > Edit > Extract • *keyboard* | **Ctrl/Cmd+Shift+X**	Detaches the selected node from any node it's connected to.		You can only extract one node at a time.
Search nodes	• *Right-click* | Node Graph > Edit > Search Node • *keyboard* | **J**	Searches nodes used in the Node Graph and jump to them.		
Bake Points	• *Right-click* | Node Graph > Edit > Bake Points > • Update Selected	• Updates the selected Bake Point nodes. • Updates the selected Bake Point node as well		

Function	How to access it	What it does	Opens this dialog box	Notes
	• Update Downstream • Smart Bake And Update • Bake Selected	as the ones that are located downstream. • Bakes the Bake Point nodes that have never been baked or are out-of-date. • Bakes the selected Bake Point nodes.		
Groups	*Right-click* | Node Graph > Edit > Groups > • Manage User Knobs • Expose Node Properties	• Opens the **Group Node Knobs** dialog. • Exposes all the properties of the selected node to the Group.		
Toggle between No Port List and Show Port List	• *Right-click* | Node Graph > Edit > Toggle Port List • *keyboard* | **Shift**+**S**	Toggles between **No Port List (Vertical)** and **Show Port List (Horizontal)** navigation types. See Mari Preferences Dialog for more information.		
Info Viewer	• *Right-click* | Node Graph > Edit > Info Viewer • *keyboard* | **I**	Shows information on the selected node(s).		
Add node	• *Right-click* | Node Graph > Nodes > Category > Node • *keyboard* | **Tab**	Adds the node to the current Node Graph at the cursor position or after a selected node.		In addition to creating nodes, you can also create shader networks, which is similar to

Function	How to access it	What it does	Opens this dialog box	Notes
				creating shaders through the Shaders Palette.
Autoplace selected nodes	• *Right-click* | Node Graph > Autoplace • *keyboard* | **L**	Rearranges all (or selected) nodes in the current Node Graph for clarity.		

Ptex Functions

A list of the functions for working with Ptex objects can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Double face resolution	• *Menubar* | **Ptex** > **Double Face Resolution** • *Button* | Ptex Toolbar >	Double the selected face resolution.		
Halve face resolution	• *Menubar* | **Ptex** > **Halve Face Resolution** • *Button* | Ptex Toolbar >	Halve the selected face resolution.		
Set the world space face resolution for the selected face	• *Menubar* | **Ptex** > **World Space Face Resolution** • *Button* | Ptex Toolbar >	Set the world space face resolution for the selected face.		
Set the face resolution for	• *Menubar* | **Ptex** > **Set Face Resolution**	Set the face resolution for the selected face.		

Function	How to access it	What it does	Opens this dialog box	Notes
the selected face	• *Button* | Ptex Toolbar >			

Patch Functions

A list of the functions for working with patches, including copying, mirroring, and transforming patches, can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Extract patch	• *Menubar* | **Patches** > **Extract Selected** • *Right-click* | canvas > **Patches** > **Extract Selected**	Extracts the selected patches as images, and displays them in the Image Manager Palette.		You can only extract images up to 8K in resolution.
Select patches by number	• *Menubar* | **Patches** > **Select Range** • *Right-click* | canvas > **Patches** > **Select Range**	Lets you enter a series of patch numbers in order to select those patches.	Select Patches	
Copy selected patches to system clipboard	• *Menubar* | **Patches** > **Copy Indexes to Clipboard** • *Right-click* | canvas > **Patches** > **Copy Indexes to Clipboard** • *Right-click* | Patches Palette > **Copy Indexes to Clipboard**	Copies selected patch IDs to the system clipboard.		Copies the UDIM of the selected patch(es) to the system clipboard, for use in other applications.
Paste selected	• *Menubar* | **Patches** >	Pastes a copied patch		

Function	How to access it	What it does	Opens this dialog box	Notes
patch	**Quick Paste** • *Right-click* | canvas > **Patches** > **Quick Paste** • *Keyboard* | **Ctrl**/**Cmd**+**V**	(from the clipboard).		
Copy selected patch	*Menubar* | **Patches** > **Quick Copy** *Right-click* | canvas > **Patches** > **Quick Copy** *Keyboard* | **Ctrl**/**Cmd**+**C**	Copies the selected patch.		Only copies the paint baked onto the patch (does not copy unbaked paint).
Copy textures	• *Menubar* | **Patches** > **Copy Textures** • *Right-click* | canvas > **Patches** > **Copy Textures**	Copies multiple textures from selected patches.	Copy Multiple Textures Dialog	
Copy the patch mask	• *Menubar* | **Patches** > **UV Mask to Image Manager** • *Right-click* | canvas > **Patches** > **UV Mask to Image Manager** • *Right-click* | patch in Patches Palette > **UV Mask to Image Manager**	Copies the outline (mask) of the UVs on the patch to the Image Manager.		
Copy the patches wireframe	• *Menubar* | **Patches** > **UV Wireframe to Image Manager** • *Right-click* | canvas > **Patches** > **UV**	Copies the wireframe of the UVs on the patch to the Image Manager.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Wireframe to Image Manager			
Extend the patch bleed area	• *Menubar* | **Patches** > **Bleed Patch Edges** • *Right-click* | canvas > **Patches** > **Bleed Patch Edges** • *Right-click* | in Patches Palette > **Bleed Patch Edges**	Extends the bleed area at the edges of the UV patches out to the edge of the entire patch.		
Flip patch vertically	• *Menubar* | **Patches** > **Transform** > **Flip Vertical** • *Right-click* | canvas > **Patches** > **Transform** > **Flip Vertical**	Flips the paint on the selected patch(es) vertically.		
Flip patch horizontally	• *Menubar* | **Patches** > **Transform** > **Flip Horizontal** • *Right-click* | canvas > **Patches** > **Transform** > **Flip Horizontal**	Flips the paint on the selected patch(es) horizontally.		
Rotate patch 90 degrees counterclockwise	• *Menubar* | **Patches** > **Transform** > **Rotate 90 CCW** • *Right-click* | canvas > **Patches** > **Transform** > **Rotate 90 CCW**	Rotates the paint on the selected patch(es) 90 degrees counterclockwise.		
Rotate patch 90 degrees clockwise	• *Menubar* | **Patches** > **Transform** > **Rotate 90 CW** • *Right-click* | canvas >	Rotates the paint on the selected patch(es) 90 degrees clockwise.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Patches > **Transform** > **Rotate 90 CW**			
Rotate patch 180 degrees	• *Menubar* | **Patches** > **Transform** > **Rotate 180** • *Right-click* | canvas > **Patches** > **Transform** > **Rotate 180**	Rotates the paint on the selected patch(es) 180 degrees.		
Mirror patch from left to right	• *Menubar* | **Patches** > **Mirror** > **Mirror Left to Right** • *Right-click* | canvas > **Patches** > **Mirror** > **Mirror Left to Right**	Mirrors the paint on the selected patch(es) from left to right.		
Mirror patch from right to left	• *Menubar* | **Patches** > **Mirror** > **Mirror Right to Left** • *Right-click* | canvas > **Patches** > **Mirror** > **Mirror Right to Left**	Mirrors the paint on the selected patch(es) from right to left.		
Mirror patch from top to bottom	• *Menubar* | **Patches** > **Mirror** > **Mirror Top to Bottom** • *Right-click* | canvas > **Patches** > **Mirror** > **Mirror Top to Bottom**	Mirrors the paint on the selected patch(es) from top to bottom.		
Mirror patch from bottom to top	• *Menubar* | **Patches** > **Mirror** > **Mirror Bottom to Top** • *Right-click* | canvas > **Patches** > **Mirror** >	Mirrors the paint on the selected patch(es) from bottom to top.		

Function	How to access it	What it does	Opens this dialog box	Notes
	Mirror Bottom to Top			
Link selected patches	• *Menubar* | **Patches** > **Link** > **Link Selected Patch Images** • *Right-click* | canvas > **Link** > **Link Selected Patch Images** • *Right-click* | Patches Palette > **Link Selected Patch Images**	Links patches that have been selected in the **Patches** palette. This allows you to quickly replicate changes from one patch across multiple patches.		A colored box highlights linked patches in the UV view.
Unlink selected patches	• *Menubar* | **Patches** > **Link** > **Unlink Selected Patch Images** • *Right-click* | canvas > **Link** > **Unlink Selected Patch Images** • *Right-click* | Patches Palette > **Unlink Selected Patch Images**	Unlinks any linked patches that have been selected in the **Patches** palette.		Removes the colored box from the unlinked patches in the UV view.
Selects all patches linked to a specific patch	• *Menubar* | **Patches** > **Link** > **Select Linked Patch Images** • *Right-click* | canvas > **Link** > **Select Linked Patch Images** • *Right-click* | Patches Palette > **Select Linked Patch Images**	Selects all of the patches that have been linked to the specified patch selection, and highlights them in the **Patches** palette.		

Function	How to access it	What it does	Opens this dialog box	Notes
Unlinks all linked patches	• *Menubar* | **Patches** > **Link** > **Unlink All Linked Patch Images** • *Right-click* | canvas > **Link** > **Unlink All Linked Patch Images** • *Right-click* | Patches Palette > **Unlink All Linked Patch Images**	Unlinks all linked patches for the entire project.		
Resize patches	• *Menubar* | **Patches** > **Resize Selected** > Size • *Right-click* | canvas > **Resize Selected** > Size • *Right-click* | Patches Palette > **Resize Selected** > Size	Resizes the selected patches to the selected size, including options to double or halve the size.		

Shading Functions

A list of functions for adding, editing, removing, and toggling shaders, as well as how to change the lighting, can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Add new shader	• *Menubar* | **Shading** > **Add New Shader** • *Right-click* | in Shaders	Adds a new shader from the choices: • **Principled BRDF**,	Create Shader Dialog when selecting	The **Choose Diffuse and Specular** option

Function	How to access it	What it does	Opens this dialog box	Notes
	Palette > **Add New Shader** • *Button* | Shaders Palette >	• **Unreal**, • **VRayMtl**, • **Layered**, and • **Choose Diffuse and Specular**.	**Choose Diffuse and Specular**. Masks Formats Dialog when selecting **Layered**.	allows you to create a custom shader. Refer to the Create Shader Dialog for more information.
Add new shader layer	• *Button* | Shaders Palette > **Shader Layers tab** >	Adds a new shader to the layer stack for the selected layered shader: • **Unreal**, • **Principled BRDF**, • **VRayMtl**, and • **Choose Diffuse and Specular**.	Create Shader Dialog when selecting **Choose Diffuse and Specular**.	The **Choose Diffuse and Specular** option allows you to create a custom shader. Refer to the Create Shader Dialog for more information.
Duplicate shader	• *Menubar* | **Shading** > **Duplicate Shader** • *Right-click* | in Shaders Palette > **Duplicate Shader** • *Button* | Shaders Palette >	Makes a copy of the selected shader.		
Delete shader	• Menubar | **Shading** > **Remove Shader** • *Right-click* | in Shaders Palette> **Remove Shader** • *Button* | Shaders Palette >	Deletes the selected shader from the project.		

Function	How to access it	What it does	Opens this dialog box	Notes
Cut shader	• *Right-click* | on shader in Shaders Palette > **Cut** • **Ctrl/Cmd+X**	Cuts the shader from the **Shaders** palette list and places it in the clipboard.		Default shaders cannot be cut or pasted.
Copy shader	• *Right-click* | on shader in Shaders Palette > **Copy** • **Ctrl/Cmd+C**	Copies the shader from the **Shaders** palette list and places it in the clipboard.		Default shaders cannot be copied or pasted.
Paste shader	• *Right-click* | on shader in Shaders Palette > **Paste** • **Ctrl/Cmd+V**	Pastes the cut or copied shader from the clipboard into the **Shaders** palette list.		
Add shader mask	• *Button* | Shaders Palette > **Shader Layers tab** >	Adds a mask to the selected shader layer in the **Shader Layers** tab.		
Add shader mask stack	• *Button* | Shaders Palette > **Shader Layers tab** >	Adds a mask stack to the selected shader layer in the **Shader Layers** tab.		
Remove shader layer	• *Button* | Shaders Palette > **Shader Layers tab** >	Removes the shader layer from the layer stack in the **Shader Layers** tab.		
Assign a color tag	• *Button* | Shaders Palette > **Shader Layers tab** >	Assigns the selected shader a color tag that can be searched for with the color tag filter.		
Set lighting on main mesh to flat	• *Menubar* | **Shading** > **Flat** • *Toolbar* |	Shades all pixels with the same value.		

Function	How to access it	What it does	Opens this dialog box	Notes
Set lighting on main mesh to basic	• *Menubar* | **Shading** > **Basic** • *Toolbar* |	Applies a basic lighting model.		
Set lighting on main mesh to full	• *Menubar* | **Shading** > **Full** • *Toolbar* |	Applies a full, configurable lighting model.		From within the **Lighting** palette, you can also adjust the four basic lights or the environment light. Mari saves these lighting values with the project.
Toggle shadows in the display	• *Menubar* | **Shading** > **Shadows** • *Toolbar* |	Toggles whether to show the shadows in the display.		This does not enable or disable shadow processing. To enable shadow processing, in the Mari Preferences Dialog, in the **GPU** tab, enable the **Shadow Maps** > **Allowed** checkbox. By default this preference is disabled.
Toggle wireframe view	• *Menubar* | **Shading** > **Toggle Wireframe** • *Keyboard* | **Shift**+**W** • *Toolbar* |	Toggles whether to show the wireframes on the model.		

Function	How to access it	What it does	Opens this dialog box	Notes
Toggle UV image rendering	• *Menubar* | **Shading** > **Toggle UvImage** • *Toolbar* |	Shows the patches in the UV view as they appear after export, including the overspill areas.		
Toggle whole patch project	• *Menubar* | **Shading** > **Toggle Whole Patch Project** • *Toolbar* |	Enables painting outside geometry shell boundaries in UV patches.		This is only available in UV views.

Painting Functions

A list of functions and shortcuts for painting and baking paint can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Undo all unbaked painting	• *Menubar* | **Painting** > **Clear Painting** • Toolbar | • *Mouse* |	Undo all painting not yet baked into the model (still in the paint buffer).		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)
Undo	• *Menubar* | **Edit** > **Undo** • *Keyboard* | **Ctrl**/**Cmd**+**Z** • *Toolbar* | • *Mouse* |	Undo the last action.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)

Function	How to access it	What it does	Opens this dialog box	Notes
				See also the History View Palette.
Redo	• *Menubar* | **Edit** > **Redo** • *Keyboard* | **Ctrl**/**Cmd**+**Y** • *Toolbar* | • *Mouse* |	Redo the last (undone) action.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.) See also the History View Palette.
Bake painting into the model	• *Keyboard* | **B** • *Menubar* | **Painting** > **Bake** • *Mouse* | • *Status bar* |	Saves the painting as part of the project.		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)
Show or hide painting	• *Keyboard* | **.**	Shows or hides the contents of the paint buffer (that is, the unbaked paint).		
Show or hide paintable areas (not currently masked out)	• *Menubar* | **Painting** > **Toggle Mask Preview** • *Keyboard* | **,** • Painting Palette | **Projection Settings** |**Mask Preview** | **Mask Preview Enabled**	Shows or hides the current masking (including all masking currently turned on - the edge mask, channel mask, ambient occlusion mask, depth mask, and backface mask).		When this is on, this shows paintable areas normally, and unpaintable areas are colored.

Function	How to access it	What it does	Opens this dialog box	Notes
Save current painting as a file	• *Menubar* | **Painting** > **Save Painting**	Saves the current unbaked paint (the contents of the paint buffer) as an **.exr** file.	Save Window Painting	Saves the paint buffer as an **.exr** file, and puts it in the Image Manager Palette.
Load previously saved painting files	• *Menubar* | **Painting** > **Load Painting**	Loads an **.exr** file and projects it onto the model as the current painting.	Load Window Painting	
Draw straight line between two points	• *Mouse* | Click, then **Shift**+click	Draws a line with the current brush between the two selected points.		
Draw straight line horizontally or vertically	• *Mouse* | **Shift**+click and drag	Constrains painting to straight horizontal or vertical lines.		
Toggle Last Brush Tip	• *Mouse* | • *Keyboard* | **N**	Switches to the last brush tip used.		

Filter Functions

A list of the filters available in Mari, where to find them, and how to apply them can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Apply Invert filter	• *Menubar* | **Filters** > **Invert**	Inverts the colors in the selection's alpha	Apply Invert	

Function	How to access it	What it does	Opens this dialog box	Notes
		channel. Replaces colors with their "opposite" in the color chart. For example, replaces a color with a float value of 0.3 with a color with float value 0.7.		
Apply Luminosity filter	• *Menubar* | **Filters** > **Luminosity**	Outputs just the luminosity value of each pixel in the selection. That is, it outputs a grayscale image based on the brightness of every pixel in the original image.	Apply Luminosity	
Apply In Blur filter	• *Menubar* | **Filters** > **Blur** > **Blur**	Applies a standard blur to the selection.	Apply Blur	You can select the blur radius - the number of pixels Mari uses when calculating the blur value for each pixel. The higher the radius, the more blurred the results. If you use a larger blur radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles patch edges. This filter is not supported with Ptex channels.

Function	How to access it	What it does	Opens this dialog box	Notes
Apply Soften Blur filter	• *Menubar* | **Filters** > **Blur** > **Soften**	Applies a subtle softening blur. This is a quick, predefined blur filter. If you need more control over the degree or type of blurring, use either the standard **Blur** or **Gaussian Blur** filters.	Apply Soften	This filter is not supported with Ptex channels.
Apply Gaussian Blur filter	• *Menubar* | **Filters** > **Blur** > **Gaussian**	Applies a Gaussian blur. Compared to the standard **Blur** filter, this gives you much finer control over the degree of blurring, and the option of using much higher blur values.	Apply Gaussian	You can select the blur radius. The higher the radius, the more blurred the results. If you use a higher blur radius, you may see artifacts along the boundaries of geometry shells and UV patches. This filter is not supported with Ptex channels.
Apply Gamma filter	• *Menubar* | **Filters** > **Gamma**	Changes the gamma levels in the selection.	Apply Gamma	**Do Inverse** reverses the gamma conversion. That is, a higher setting for **Gamma** results in a more washed-out looking image.
Apply sRGB To Linear Filter	• *Menubar* | **Filters** > **sRGB To Linear**	Applies an sRGB to linear colorspace conversion.	Apply sRGB To Linear	Checking **Invert** applies a linear to sRGB colorspace conversion.
Apply Hue filter	*Menubar* | **Filters** > **Hue**	Changes the hue, saturation, and value of	Apply Hue	To change the hue of

Function	How to access it	What it does	Opens this dialog box	Notes
		the colors on the model.		the colors, set the value for Hue. This moves the colors around the color wheel. The rotation value is the degree around the color wheel that each color shifts, between 0 and 360 (which both correspond to the original colors). The options to change the **Saturation** are: • **Saturation** - how much saturation is applied. Select a multiplier from 0 to 2, where 1.00 is the original saturation value. • **Offset** - an offset value to add to the initial saturation value, before applying the scale. You can also edit the **Saturation** curve. This controls how the filter translates current saturation values to new values. For each input value along the bottom of the graph, the output value is set

Function	How to access it	What it does	Opens this dialog box	Notes
				from the position of the line above that point. Edit the curve by: • pulling the anchor points, • clicking between anchor points to add a new point, and • right-clicking to select from a set of preset values. You can edit the **Scale**, **Offset** and **Value** curve for the value of the colors in the same way.
Apply Brightness filter	• *Menubar* | **Filters** > **Brightness**	Changes the brightness of the paint, either overall or for specific color channels. You can also change the contrast.	Apply Brightness	You can either change the brightness of all color channels at once, or one- by-one. Changing the main **Brightness** control changes (overwrites) any changes you make to brightness values for individual color channels.
Apply Contrast filter	• *Menubar* | **Filters** > **Contrast**	Changes the contrast of the paint.	Apply Contrast	You can select the amount of contrast using the slider.
Apply Clamp filter	• *Menubar* | **Filters** > **Clamp**	Clamps color values to lie within the specified upper and lower values.	Apply Clamp	Set the **Upper Value** and **Lower Value** by adjusting the sliders or

Function	How to access it	What it does	Opens this dialog box	Notes
				entering values in the text fieldes. Select the individual **Color Component** to clamp in the dropdown menu.
Apply Levels filter	*Menubar* | **Filters** > **Levels**	Changes the color levels in the paint, by setting the white point, midtone, and black point.	Apply Levels	You can change the levels of all color components or each color component separately. You can set the: • **Color Component** - whether this affects all color components, or a specific one (**Red**, **Green**, or **Blue**). • **White Point** - the upper limit of displayed color intensity. Areas with a value higher than this are mapped to 1 (white). • **Midtone** - the middle point between white and black. Mari remaps the values so that this is the middle of the range between the white point and black point. Moving this pushes the values towards that end of

Function	How to access it	What it does	Opens this dialog box	Notes
				the spectrum. • **Black Point** - the lower limit of displayed color intensity. Areas with a value lower than this are mapped to 0 (black). • **White Output Level** - how strongly to output the white in the filtered image. Higher values show the entire white output, lower values show decreasing amounts. • **Black Output Level** - how strongly to output the black in the filtered image. Higher values show the entire black output, lower values show decreasing amounts.
Apply Color Curves filter	• *Menubar* | **Filters** > **Color Curves**	Changes the color curve for each color component (RGB) in the painting.	Apply Color Curves	You can also edit each color curve, controlling how each color component appears on-screen. This controls how the filter translates current color values to new values. For each

Function	How to access it	What it does	Opens this dialog box	Notes
				input value along the bottom of the graph, the output value is set from the position of the line above that point. Edit the curve by: • pulling the anchor points, • clicking between anchor points to add a new anchor point, • right-clicking to select from a set of preset values.
Apply Color Switches filter	• *Menubar* | **Filters** > **Color Switches**	Turns individual color components on or off.	Apply Color Switches	Use the checkboxes to select which color components are displayed. When Mari applies the filter, it removes any unchecked color components from the painting.
Apply Tone Mapping filter	• *Menubar* | **Filters** > **Tone Mapping**	Varies the "exposure" of the painting.	Apply Tone Mapping	You can set the: • **White Point** - the upper limit of color intensity displayed; any colors of greater intensity map to white. • **Exposure** - lower exposures give darker images; higher exposures give lighter

Function	How to access it	What it does	Opens this dialog box	Notes
				ones.
Apply Copy Channel filter	• *Menubar* | **Filters** > **Copy Channel**	Copies the value from one RGB color component to the other two. The result is a grayscale image with the intensity values from the selected component.	Apply Copy Channel	Select the source component from the list. When you apply the filter, Mari copies the selected component over the other two components.
Apply Premultiply Alpha filter	• *Menubar* | **Filters** > **Premultiply Alpha**	Either pre- or post-multiplies the alpha in the selected image. If you are painting using an image without premultiplied alpha onto one with it, use this filter to perform the premultiplication, so the images match and you avoid lines around the outside of the patch. Post-multiply works the same, but in reverse (removes premultiplication to match images that do not have premultiplied alphas).	Apply Premultiply Alpha	If the **Postmultiply** box is checked, the premultiplied alpha is removed.
Apply Edge Detect filter	• *Menubar* | **Filters** > **Edge Detect**	Finds "edges" (transitions between colors) in the image and heightens them.	Apply Edge Detect	This filter is not supported with Ptex channels.
Apply	• *Menubar* | **Filters** >	Applies an "emboss"	Apply	This filter is not

Function	How to access it	What it does	Opens this dialog box	Notes
Emboss filter	**Emboss**	effect, so that textures look "raised" on the surface.	Emboss	supported with Ptex channels.
Apply Sharpen filter	• *Menubar* | **Filters** > **Sharpen**	Removes 'fuzziness' and clarifies detail in the image.	Apply Sharpen	You can set the amount of sharpening. Higher values mean more sharpening, while a value of 0.00 is the picture without the filter applied. This filter is not supported with Ptex channels.
Apply High Pass Filter	• *Menubar* | **Filters** > **High Pass**	Removes low frequency information from the image while maintaining higher frequency detail. This can be useful for quickly creating specular maps where only the higher frequency image highlights are maintained.	Apply High Pass	You can set the kernel size (the amount of filtering) by adjusting the **Radius**. If you use a large radius, you may get slight lines on the boundaries between patches. This is an artifact of the way that Mari handles patch edges. This filter is not supported with Ptex channels.
Apply OCIO Colorspace filter	• *Menubar* | **Filters** > OCIO Colorspace	Applies a a given OCIO colorspace to the current channel or painting.	Apply OCIO Colorspace	Set the colorspace of the current channel or painting in the **Input Colorspace** dropdown.

Function	How to access it	What it does	Opens this dialog box	Notes
				Set the colorspace correction to apply to the current channel or painting in the **Output Colorspace** dropdown.
Apply Add Noise filter	• *Menubar* | **Filters** > **Add Noise**	Applies noise to the current channel or painting.	Apply Add Noise	Use the **Amount** slider to adjust how much noise is applied and select the component you want to affect from the **Color Component** dropdown. For additive grayscale noise that modifies the intensity while retaining the original color, check the **Grayscale** checkbox. You can regenerate the noise using different random samples by dragging on the **Seed** slider. Adjust the size of the noise applied by dragging on the **Size** slider.

Camera Functions

A list of functions for using cameras, including how to project and unproject them, saving views and snapshots, and commands for moving the camera, can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
View all selected areas	• *Menubar* | **Camera** > **View All** • *Keyboard* | **A** • *Mouse* |	Fills the canvas with all selected areas (or all objects in the project if nothing is selected).		For the mouse gesture: press and hold ****, click and drag in the direction indicated. (A thin red line traces gestures onscreen.)
Load Alembic or FBX Camera	• *Menubar* | **Camera** > **Load Camera**	Loads a 3rd party **.abc** or **.fbx** camera view point of the current model. (The view point may seem arbitrary, depending on the model)	**Load Camera**	File format is **.abc** or **.fbx**
Project onto the front patches from the current view	• *Menubar* | **Camera** > **Quick Project Front**	Loads the data from the file set in the **Quick Projection** dialog. The result is projected onto the patches of the model visible from the current view, and baked on.	**Quick Projection** (if details have not already been set)	
Project onto all images from the current view	• *Menubar* | **Camera** > **Quick Project Through**	Loads the data from the file set in the **Quick Projection**	**Quick Projection** (if details have not already	

Function	How to access it	What it does	Opens this dialog box	Notes
		dialog. The result is projected onto all patches of the model (whether visible in the current view or not) and baked on.	been set)	
Unproject from the current view	• *Menubar* | **Camera** > **Quick Unproject**	Saves the current baked painting, as visible in the current view, to an image file (as set in the **Quick Projection** properties dialog box).	**Quick Projection** (if details have not already been set)	
Set up quick projection	• *Menubar* | **Camera** > **Quick Projection Settings**	Lets you set the image **Resolution**, **Path** and **Lighting** setup for the projected file.	**Quick Projection**	
View left side of model	• *Menubar* | **Camera** > **Camera Left** • *Keyboard* | **1**	Changes the view to show the "left" side of the model.		Model's "left" (may seem arbitrary, depending on the model).
View right side of model	• *Menubar* | **Camera** > **Camera Right** • *Keyboard* | **2**	Changes the view to show the "right" side of the model.		Model's "right" (may seem arbitrary, depending on the model).
View top of model	• *Menubar* | **Camera** > **Camera Top** • *Keyboard* | **3**	Changes the view to show the "top" of the model.		Model's "top" (may seem arbitrary, depending on the model).
View bottom of model	• *Menubar* | **Camera** > **Camera Bottom** • *Keyboard* | **4**	Changes the view to show the "bottom" of the model.		Model's "bottom" (may seem arbitrary, depending on the

Function	How to access it	What it does	Opens this dialog box	Notes
				model).
View front of model	• *Menubar* | **Camera** > **Camera Front** • *Keyboard* | **5**	Changes the view to show the "front" the model.		Model's "front" (may seem arbitrary, depending on the model).
View back of model	• *Menubar* | **Camera** > **Camera Rear** • *Keyboard* | **6**	Changes the view to show the "rear" the model.		Model's "back" (may seem arbitrary, depending on the model).
Reset the camera	• *Menubar* | **Camera** > **Reset Camera**	Resets the physical camera attributes within the viewport including the focal point of the camera.		**Article:** Take a look at the Differences between 'Reset navigation settings to their default' and 'Reset Camera' article.
Create a projector	• *Right-click* | in Projectors Palette > **Create Projector** • *Button* | Projectors Palette >	Sets the current camera and paintable region, to frame a particular view on the model.		
Use the view saved in a particular projector	• *Right-click* | **Projector** in Projectors Palette > **Make Projector Current**	Changes the current view to the view set in the selected projector.		

Function	How to access it	What it does	Opens this dialog box	Notes
Save the current camera view	• *Right-click* | **Projector** in Projectors Palette > **Save Projector** • *Button* | Projectors Palette >	Saves the current projector (including all details of the view) to disk.	Save	
Save all projectors defined in the project	• *Right-click* | in Projectors Palette > **Save All Projectors**	Saves all the projectors currently defined in the project. Mari saves all the projectors as a single file.		File format is **.mpc** (Mari Projector Collection).
Load projectors from disk	• *Right-click* | in Projectors Palette > **Load Projector** • *Button* | Projectors Palette >	Loads one or more projectors from a file, and changes the current view to the first projector in the file.	Load	
Save a snapshot from a projector to a file	• *Right-click* | **Projector** in Projectors Palette > **Unproject**	Saves the current baked painting from the selected projector to an image file.	**Save Unprojected Image** (if **Output Path** has not been set yet)	
Load image data from the current projector	• *Right-click* | **Projector** in Projectors Palette > **Project**	Loads the data from the file specified for that projector, projects it onto the model, and bakes it on.	**Pick Path** (if **Input Path** has not been set yet)	
Load image data into the paint buffer	• *Right-click* | **Projector** in Projectors Palette > **Import Image**	Loads the data from the file specified for that projector, and	**Pick Path** (if Input Path has not been set	

Function	How to access it	What it does	Opens this dialog box	Notes
from the current projector		puts it into the paint buffer.	yet)	
Load a projector	• *Right-click* | **Projector** in Projectors Palette > **Make Projector Current**	Displays the current project with that projector (camera and paintable region).		
Remove a projector	• *Right-click* | **Projector** in Projectors Palette > **Remove Projector** • *Button* | Projectors Palette >	Deletes that projector.		
Generate a render turntable from a projector	• *Right-click* | **Projector** in Projectors Palette > **Render Turntable**	Takes the scene as viewed through the selected projector, and creates a series of images showing the scene rotating through an axis. You can include custom text or thumbnails of reference images in the turntable.	Render Turntable Dialog	
Generate a diagnostic turntable from a projector	• *Right-click* | **Projector** in Projectors Palette > **Diagnostic Turntable**	Takes the scene as viewed through the selected projector, and creates a turntable for a set of channels. The turntable has flat lighting and uses the default shader.	Diagnostic Turntable Dialog	

Function	How to access it	What it does	Opens this dialog box	Notes
Orbit	• *Mouse* | **Alt** and drag	Drag to orbit the model in 3D space.		Keys plus mouse movement, uses last mouse click as pivot point.
Zoom	• *Mouse* | **Alt**+**Ctrl**/**Cmd** and drag left or right	Drag left to zoom out, right to zoom in.		Keys plus mouse movement.
Pan	• *Mouse* | **Alt**+**Shift** and drag	Drag the model to a different place on the canvas.		Keys plus mouse movement
Roll	• *Mouse* | **Ctrl**/**Cmd**+**R** and drag	Drag up to roll the model clockwise, left to roll counterclockwise		Keys plus mouse movement.
Focus	• *Keyboard* | **F**	Re-centers view on selected patch(es).		If no patches are selected, focuses on the center of the model.
Switch to Ortho Camera	• *Keyboard* | **F10**	Switches the canvas view to **Orthographic** mode.		
Switch to Perspective Camera	• *Keyboard* | **F11**	Switches the canvas view to **Perspective** mode.		Enables the **Near**, **Far**, and **FoV** fields. See Projectors Palette Fields for more information.
Switch to UV Camera	• *Keyboard* | **F12**	Switches the canvas view to **UV** mode.		
Reset the model position	• *Keyboard* | **1**	In the UV view, resets the model to its default position.		

Function	How to access it	What it does	Opens this dialog box	Notes
Rotate the model 90 degrees counter-clockwise	• *Keyboard* | **2**	In the UV view, rotates the model 90 degrees counter-clockwise from the default position.		
Rotate the model 180 degrees counter-clockwise	• *Keyboard* | **3**	In the UV view, rotates the model 180 degrees counter-clockwise.		
Rotate the model 270 degrees counter-clockwise	• *Keyboard* | **4**	In the UV view, rotates the model 270 degrees counter-clockwise.		

View Functions

A list of functions for changing your view and saving layouts can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
Set the display properties	• *Right-click* | main canvas > **Display Properties** • *Menubar* | **View** > **Display Properties**	Opens a dialog box allowing you to control how the main canvas displays.	Display Properties Dialog	
Save the current layout	• *Menubar* | **View** > **Save Layout**	Saves the current layout of your Mari workspace as a ***.mws** file.	Save Window Layout	Saves the position of palettes and other elements of the GUI.

Function	How to access it	What it does	Opens this dialog box	Notes
Open a saved layout	• *Menubar* | **View** > **Load Layout**	Opens a saved layout (***.mws** file).	Load Window Layout	By default, Mari starts with the last-used layout.
Revert to the default layout	• *Menubar* | **View** > **Default Layout**	Resets the layout back to the Mari default.		
Save image of current canvas	• *Menubar* | **View** > **Take Screenshot**	Saves the current view of the canvas as an image file.		Various image formats available.
Adjust the screenshot settings	• *Menubar* | **View** > **Screenshot Settings**	Opens a dialog box that allows you to set how screenshots are saved.	Screenshot Settings	
Hide docked palettes	• *Menubar* | **View** > **Hide Palettes** • *Keyboard* | **Space bar**	Hides all docked palettes, so the canvas takes up the entire workspace.		The **Space** bar cycles between showing all palettes, hiding undocked palettes, and hiding all palettes.
Hide GUI components on canvas	• *Keyboard* | **/**	Hides displayed information (such as the histogram, heads-up display, etc) from the canvas.		
View a particular palette	• *Menubar* | **View** > **Palettes** > <palette name> • *Right-click* | **Menubar** or **Toolbar** > <palette name>	Displays the selected palette.		The palette re-displays in its last position.
Enter full screen mode	• *Menubar* | **View** > **Full Screen**	Enters full screen mode		Click **View** > **Full Screen** to exit full screen mode.

Tool Functions

A list of functions for managing plug-ins, imported custom content, and other tools can be found in the table below.

Function	How to access it	What it does	Opens this dialog box	Notes
View installed plug-ins	• *Menubar* | **Tools** > **Plug-ins**	Displays a list of installed plug-ins.	Plug-ins	Drag the edge of the dialog box, and column headings within, to display full content.
Open the Shader Console	• *Menubar* | **Tools** > **Shader Console**	Opens the **Shader Console** to allow you to see GLSL errors and source code information, as well as to reload XML filed and shaders.	Shader Console	
Imports Custom Brushes	• *Menubar* | **Tools** > **Import Brushes**	Allows you to select the custom Photoshop brush file that you want to import into Mari	Import Brushes	Photoshop custom brush files must be in the **.abr** file format.

Python Functions

A list of functions for working with Python and accessing the Python and API documentation can be found in the table below.

Function	How to access it	What it does	Notes
Execute example scripts	• *Menubar* | **Python** > **Examples**	Executes example scripts and displays results in the output pane of the **Python Console**.	
View Python API documentation	• *Menubar* | **Python** > **Documentation**	Displays the Python API documentation in a web browser.	
View Software API Overview documentation	• *Menubar* | **Python** > **Readme**	Displays the Software API Overview documentation.	An introduction to Python in Mari and GLSL shaders
Display Python Console	• *Menubar* | **Python** > **Show Console**	Displays the Python Console	
Zoom in on Python Console	• **Ctrl**/**Cmd**+ **+** on the text in the Python Console	Zooms in on the text in the **Python Console** output pane.	
Zoom out from Python Console	• **Ctrl**/**Cmd**+ **-** on the text in the Python Console	Zooms out from the text in the **Python Console** output pane.	
Evaluate a script in the Python Console	• **Ctrl**/**Cmd**+**Return**	Evaluates a script that is written in the **Python Console** input pane or that is input in the **Script Path** field.	

Help Functions

A list of the functions for accessing the documentation and training information in Mari can be found in the table below.

Function	How to access it	What it does	Notes
Create an example Mari project	• *Menubar* | **Help** > **Create Example Project**	Creates an example project using the **Blacksmith Head** model.	
Create an example Mari material design project	*Menubar* | **Help** > **Create Example Material Design Project**	Creates an example material design project using the **materialGeo** model.	
View information about Mari	• *Menubar* | **Help** > **About**	Displays the Mari info screen.	About this release, credits, and so on.
View the Mari Release Notes	• *Menubar* | **Help** > **Release Notes**	Displays a web page that links to the release notes.	
View the Mari Training Resources web page.	• *Menubar* | **Help** > **Training and Tutorials**	Displays a web page with links to user guide assets and video tutorials.	
View the Mari Online Help	*Menubar* | **Tools** > **Foundry Online Help**	Displays the Online Help where you can find information on how to work with Mari and a Reference Guide.	
Make changes to your license	*Menubar* | **Tools** > **License**	Lets you: • purchase, activate, and install licenses, • tell Mari where to find a license server, • obtain 15-day trial licenses, • see how many days remain before your license expires (if you are using a temporary license), • see your System ID (if you are using a node locked license), and • see your server name (if you are using a floating license).	Mari Licensing

Function	How to access it	What it does	Notes
View Software API Overview documentation	• *Menubar* | **Help** > **SDK** > **API Overview**	Displays the Software API Overview documentation.	An introduction to Python in Mari and GLSL shaders.
View the Python API documentation and examples.	• *Menubar* | **Help** > **SDK** > **Python**	Displays the Python API documentation and a list of examples.	
View the C API documentation and examples.	• *Menubar* | **Help** > **SDK** > **C API**	Displays the C API documentation and a list of examples.	

Dialogs

Learn about Mari's dialogs, how to access them and what they do.

Use the left menu to see the details for Mari's dialogs.

Project Dialogs

Learn about the Mari dialogs, how to access them and what they do.

New Project Dialog

The **New Project** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

The **New Project** dialog lets you select geometry and various options for a new Mari project. Use the **Path** control to select an **.obj**, **.ptx**, **.abc**, or **.fbx** file you want to use in the project.

Resize the dialog box to display long filenames or select multiple files to display more than one geometry, if they are of the same file type. Mari checks the model for any errors that might prevent it from processing, for example, whether UVs overlap or touch border edges. A **Mesh Sanity Check** dialog displays any warnings or errors. (Errors cancel opening the model, but you can continue with warnings.)

Depending on the size and complexity of textures, it can take several minutes the first time you load them into a Mari project. Once saved, however, the project should open quickly in future.

How to Access It

- *Menubar* | **File** > **New**
- *Keyboard* | **Ctrl**/**Cmd**+**N**
- *Projects tab* | **New** button
- *Projects tab* | *Right-click* > **New**
- *Toolbar* |

New Project Dialog Box Fields

Control	Type	What it does	Opens this dialog box	Notes
Project Name	text field	Lets you specify the name of the project.		This is for you to identify the project only (this is not a filename).
Geometry				
Path	text field, dropdown, button	Opens **.obj**, **.ptx**, **.abc**, or **.fbx** files. Dropdown lists the last 10 opened. Once you select a file, the	**Pick Mesh** dialog	Select by either typing the **Path**, selecting from the dropdown of the last 10 files selected, or clicking the button to

Control	Type	What it does	Opens this dialog box	Notes
		Mesh options appear under the root **Path**.		browse to a file.
Mesh options | **Mapping Scheme**	dropdown	If there is a UV map available, and you would like to use it**,** select **UV if available, Ptex otherwise**. If you prefer to use Ptex, select **Force Ptex**. This option loads the selected geometry and replaces the existing UV map with a Ptex map.		Only available when choosing **.obj**, **.abc**, or **.fbx** files in the **Path**.
Mesh options | **Selection Groups**	dropdown	Selection groups have two options, to either: • **Create From Face Groups** - this is the default behavior and allows the use of selection groups in Mari. • **Don't Create** - if you choose not to create a selection group, Mari does not create any from the file.		Only available when choosing **.obj** files in the **Path**.
Mesh options | **Multiple Geometries Per Object**	dropdown	If your **.obj** consists of multiple geometries, you have the option to **Merge Geometries Into One**, which is the default behavior, or **Create Separate Geometries (Ptex Only)**. If you select		Only available when choosing **.obj** files in the **Path**. Even if you select **Create Separate Geometries (Ptex Only)**, Mari only displays a single geometry on the canvas.

Control	Type	What it does	Opens this dialog box	Notes
		Merge Geometries Into One, Mari combines the individual geometries into a single geometry, while selecting **Create Separate Geometries (Ptex Only)** creates individual geometries.		However, when you export a channel you have the option to export individual **.ptx** files for each geometry.
Mesh options | **Start Frame** and **End Frame**	text field	Set the frame range from the source file to import.		Only available when you are loading an animated sequence using the **.obj**, **.abc**, or **.fbx** formats.
Mesh options | **Single Frame**	button	Sets the other fields to only import the first frame from the animation in the object file.		Only available when you are loading an animated sequence using the **.obj**, **.abc**, or **.fbx** formats.
Mesh options | **All Frames**	button	Sets the other fields to import the entire animation from the object file.		Only available when you are loading an animated sequence using the **.obj**, **.abc**, or **.fbx** formats.
Mesh options | **Merge Type**	dropdown	For Ptex files, there are two choices for the geometry mesh options: • **Separate Meshes** - creates only one object with one geometry. • **Single Mesh** - creates only one object, but has multiple geometries within it.		Only available when choosing **.ptx** files in the **Path**. The option to **Separate Geometry In One Object** is essentially the inverse of the Ptex channel export process, and allows project reconstruction.
Mesh options | **Merge Type**	dropdown	Choose whether to merge selected items		Only available when choosing **.abc** or **.fbx**

Control	Type	What it does	Opens this dialog box	Notes
		hierarchy tree as: • **Single Mesh** - merges selected geometry into one single mesh in a GeoEntity. **Note:** Note that this implies that the information about the original geometries cannot be accessed once the merge is done. • **Just Merge Nodes** - merges selected geometry into one GeoEntity while retaining the metadata and mesh information about the original individual geometries.		files in the **Path**.

Control	Type	What it does	Opens this dialog box	Notes
		Note: This information can only be accessed by more technical artists using the C API. see, Software API Overview.		
Mesh options |**Merge Selection Groups**		You can choose to **Merge selection groups with the same name** or choose the **Do not merge selection groups** option.		For more information on selection groups, see Selection Groups.
Mesh options |**Take**	dropdown, information	If an object has been exported as part of an **.fbx** file with animation takes, the takes are shown in the dropdown menu. Otherwise, information regarding the only take available is shown.		Only available when choosing **.fbx** files in the file browser. When adding an object with animation takes, the take can only be selected on import and cannot be changed at a later point.
Mesh options | **Scene**	information	In the object hierarchy, shows all the geometries or meshes, cameras, and groups that are part of the selected Alembic file. Expanding the parent row, reveals the child geometries beneath it.		Only available when choosing **.abc** or **.fbx** files in the **Path**.

Control	Type	What it does	Opens this dialog box	Notes
Mesh options | **Merge**	checkbox	Select the checkbox to determine if a geometry is to be merged with another on project creation.		Only available when choosing **.abc** or **.fbx** files in the **Path**. For more information on merging Alembic or FBX geometry, see Creating a New Project with Alembic and FBX Files
Mesh options | **Frame Offset**	text field	Set by how many frames to offset the loading of an animated object. You can specify a frame range, or simply a number of frames.		Only available when choosing **.abc** files in the **Path**.
Channels				
Root Path	dropdown, button	Displays the root path for the project. This is where Mari looks for any existing textures to import as it creates the channels.	**Choose Base Path** dialog	This is the value that Mari substitutes for $PATH in the import templates in the bottom list.
Scan	button	Scans the **Root Path** for any existing texture files.		After scanning the **Root Path**, you should always set the **Size** of the texture files manually. The default values Mari sets may not be correct as channels can contain many different patch sizes.
Category	dropdown	Selects a Mari channel preset template from the		**General**, **Principled BRDF - Metallic -**

Control	Type	What it does	Opens this dialog box	Notes
		list of available shaders.		**Games**, **Principled BRDF - Metallic - VFX, Principled BRDF - Non Metallic - Games, Principled BRDF - Non Metallic - VFX**, **Unreal - Metallic - Games**, **Unreal - Metallic - VFX**, **Unreal - Non-Metallic - Games**, **Unreal - Non-Metallic - VFX**, **VRayMtl - VFX**
Template	dropdown	Specifies a channel template to control the format of the filenames when Mari looks for existing textures on disk.		For more information about channel templates, see Defining Custom Channel Templates, Reviewing the Custom Channel Template File Format, and Example File: DefaultTemplates.mtt.
Prefix	text field	Specifies the filename prefix for existing texture files, if applicable.		
Create	checkbox	When enabled, creates a shader and connects channels to the shader on project creation.		Most presets contain enabled channels by default.
Import	checkbox	Imports existing textures into the channel after creating it.		
Name	information	Specifies the name of the channel to create. Mari		For example, color, specRough, dispFine,

Control	Type	What it does	Opens this dialog box	Notes
		sets a default value, but you can change this.		dispBroad, colorDetail, colorDetailMask.
Size	dropdown	Specifies the patch size for the new channel. Mari sets a default value, but you can change this.		**256x256**, **512x512**, **1024x1024**, **2048x2048**, **4096x4096**, **8192x8192**, **16384x16384**, **32768x32768**
Colorspace	dropdown	The colorspace in which the colors within your project are interpreted. Either select from the shipped config files or allow Mari to set an **Automatic** colorspace.		You can change these settings within the project after project creation, if necessary.

Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.

Scalar	checkbox	When **Colorspace** is set to **Automatic**, checking **Scalar** converts the resolved colorspace to scalar data. Scalar data in Mari is used to define the properties of a texture such as bumpiness, roughness, or shininess. See Color Data and Scalar Data for more information.		**Scalar** is disabled by default.

Control	Type	What it does	Opens this dialog box	Notes
		Note: Automatically resolved 32-bit channels cannot be converted to scalar data. Only 8- and 16-bit channels are supported.		
File Space	dropdown	Specifies the file space used when generating vectors. Mari sets a default but you can change this.		**Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**
Fill	swatch	Specifies the color to fill the new channel. Mari sets a default value, but you can change this.		Click the swatch to change the color.
Depth	dropdown	Specifies the color depth size. Mari sets a default value, but you can change this.		**8bit (Byte)**, **16bit (Half)**, or **32bit (Float)**.
Files	entry	Specifies the filename template used to pick existing textures to import into the channel. Mari sets a default value, but you can change this, or you can use the button to open a dialog allowing you to manually pick the		This option is ignored if the **Import** checkbox is disabled.

Control	Type	What it does	Opens this dialog box	Notes
		files.		
	button	Opens a dialog box allowing you to pick the textures for Mari to import into the new channel.	Import Dialog	
	image	Indicates that Mari has not found any textures to import.		If there are textures to import, Mari displays the number of patches available.
Color Settings				
Note: All of the following **Colorspace** options can also be changed after project creation, if you decide you no longer want to use the colorspace setting that was chosen during project creation.				
Color Management Enabled	checkbox	Enables or disables the OpenColorIO color management.		
OCIO Config	dropdown	A list of either user-defined config files (**Custom**) or shipped OpenColorIO config files that handle colorspace.		
Custom OCIO Config	text field, button	If you chose **Custom** for the **OCIO Config** file choice, specify the filepath to the config file.		
Color Monitor	dropdown	The default colorspace, which applies to thumbnails and UI elements.		

Control	Type	What it does	Opens this dialog box	Notes
		Typically, this should be the same colorspace as you intend to use for the canvas.		
Scalar Monitor	dropdown	The default colorspace for all user interface elements which are displaying non-color managed (i.e. scalar) swatches, pickers, and thumbnails. (To understand the difference between color and scalar data in Mari, see Color Data and Scalar Data). This preference does not affect the canvas and image viewers, however it is recommended to set this to match the scalar View Transform. (The active selection can be changed using the View Transform Toolbar .)		
Color Picking (advanced and intermediate color management modes only)	dropdown	The colorspace for all color pickers, swatches, and image viewers.		
Working (advanced color	dropdown	Default for painting, lighting, applying filters,	Working (advanced	dropdown

Control	Type	What it does	Opens this dialog box	Notes
management mode only)		and similar operations. Typically, you'll want to select the same linear-based colorspace as your 16/32-bit float channels to get accurate results.	mode only)	
Blending (advanced color management mode only)	dropdown	Default for applying blending operations such as: a comp of the paint buffer and the current paint target, blending between layers, and combining inputs in a merge node. Typically, you'll want to set this to **Same as 8 bit Data** to get the same results as painting prior to Mari 3.0, and to get less banding artifacts when painting 8-bit channels and images.	Blending (advanced mode only)	dropdown
8 bit Color	dropdown	The default colorspace for 8-bit channels, and reading and writing image files with 8-bit data.		
16-bit Integer Data	dropdown	The default colorspace for 16-bit channels, and reading and writing image files with 16-bit data.		

Control	Type	What it does	Opens this dialog box	Notes
8/16-bit Scalar	dropdown	The default colorspace for masks, heights, normals, depths and, generally, any non-color image with 8-bit or 16-bit data.		
16/3-bit Float Data	dropdown	The default colorspace for 16- and 32-bit (float) channels, and reading and writing image files with 16- and 32-bit, floating point data.		
Lighting				
Shader	dropdown	Specifies the shading model from one of the following: • **Current Shader** • **Current Channel** • **Current Layer & Below** • **Current layer** • **Current Paint Target**		
Lighting	dropdown	Specifies the lighting on the main mesh. Mari comes with five default modes for lighting the model, and optional shadow control: • **Flat** - a flat, uniform, directionless light.		

Control	Type	What it does	Opens this dialog box	Notes
		• **Basic** - basic diffuse lighting without specular. • **Basic with Shadows** - basic lighting with diffuse lighting, as well as shadows. • **Full** - full lighting with diffuse and specular lighting. • **Full with Shadows** - full lighting with diffuse and specular lighting, as well as shadows.		
Active Light Count	text field, slider	Specifies the number of active directional lights.		
Environment Map	arrows, preview button, file browser, reset	Click the blank image space to choose an environment from the Mari Environment library. Clicking either of the arrow keys cycles backwards or forwards through the Mari Environment library, loading the image as it does. Clicking the file browser opens the **Open an Image** dialog. Select an image of any of the file formats listed in the **File of Type** field.		For more about the options in the dialog, refer to Open an Image Dialog.

Control	Type	What it does	Opens this dialog box	Notes
		You can also drag-and-drop an image from a file browser onto the blank image space to load the image as an environment light.		
Create New Project	button	Creates and opens the new project.		
Cancel	button	Cancels the new project creation.		

Importing USD Files

USD is Pixar's extensible, layered scene description framework that offers scalable, collaborative, non-destructive workflows for VFX and animation production. Mari ships with a user friendly importer for bringing USD files directly into your project.

When loading in a USD file, you are offered specific settings within the **Geometry** tab. Options found in the **Channels**, **Color Settings** and **Lighting** tabs are the same as they would be when loading a .obj, .abc or .fbx file.

> **Note:** Mari can open **.usd**, **.usda**, **.usdc**, or**.usdz** files.

Use **File** > **New** to open the **New Project** dialog, or **Objects** > **Add Object** to open the **Add Objects** dialog.

> **Note:** USD Cameras can only be loaded into Mari via the **New Project** dialog. Camera type prims are not shown or importable via the **Add Object** dialog.

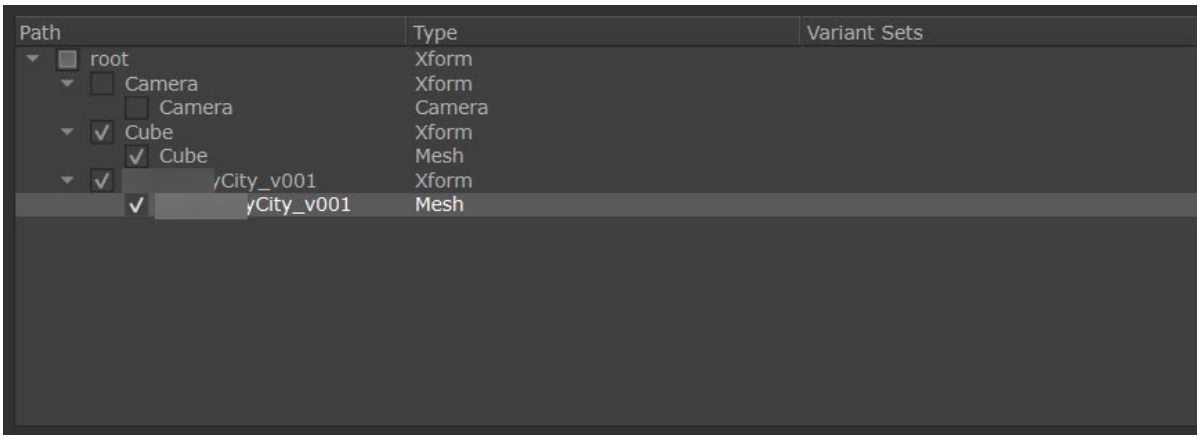
In the **New Project** dialog, select a **Path** for the file by typing, using the dropdown to select from the last ten files, or clicking the browse button to open the **Mesh Options** dialog.

If you import a USD Camera, it will appear in Mari as a Projector, so the camera angles and information will be consistent with what was imported, and will be consistent throughout the pipeline.

Filtering USD Imports Using the Mesh Options Tree

Mari's USD importer allows you to filter the scene by path and variant sets, before the import.

The **Mesh Options** tree shows a hierarchical collection of all prims contained within the incoming USD file. The type of prim is defined in the **Type** column.



Choose the items you want to import by toggling the checkmark next to the prim name.

Use the **Mesh Options** icons to filter through the list.

Selection options are:

- **Select All** - Check all prims in the tree for import.
- **Select None** - Uncheck all prims to exclude from import. This option is useful if you then want to select particular prims.
- **Select by Expression** - Use a simple expression to filter the prims. This option lets you filter by path name without having to choose from the mesh options tree.
  To include prims, add prim paths separated by commas. For example: **/room/table,/room/chairs**
  To exclude prims, prefix the prim path name with a **!**. For example:**/room,!/room/cupboard**
  Note: To exclude a prim, you must first include its parent prim.
  You can combine inclusion and exclusion as required.

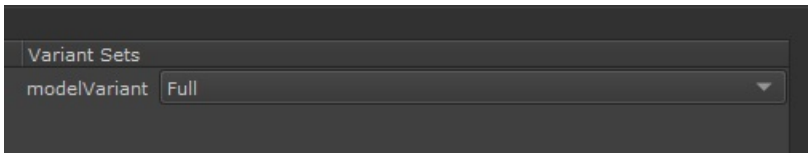
The buttons above the import table offers the following tree-viewing options:

- **Show Selected Items** - Shows the selected items in a separate view.
- **Expand All** - Fully expand the tree.
- **Collapse All** - Collapse the tree to the top-level prims.

Filtering USD Imports by Variant Sets

In USD, a Variant Set is a convenient way of managing alternate versions of models so you can choose which you prefer. USD supports two main types of variant sets: model variant sets and shading variant sets. Currently, Mari lets you load model variant sets only.

If there are any variant sets in the USD file, you must select your preferred variant from each set. The New Project dialog is populated with all model variant sets with the name of each set and a dropdown to choose the preferred variant.



Additional USD Import Options

Refine the import using the following options.

Control	Type	Role
Merge Type	dropdown	Choose whether to merge the models: • **Merge Models** - merges all of your models into one object. • **Keep Models Separate** - imports all specified models as separate objects.
UV Set	dropdown	Specify which UV Set from your USD file to use in Mari.
Frame Numbers	text field	Specify which frame numbers to load.
Mapping Scheme	dropdown	Specify the Mapping Scheme, there are two options: • **UV if available, Ptex otherwise** - allows you to work with UV maps specified in the **UV Set** parameter.

		• **Force Ptex** - ignores any UV sets assigned to the model and uses Ptex instead. • **UV if available, empty otherwise** - Load the UV layout if available. If there is no UV layout, an empty UV layout. • **Force empty** - Force to create empty UV layout no matter if there is UV layout. For more information on Ptex, also known as Per-face Texturing, see Ptex.
 Keep everything centered	button	Toggle on to discard any transforms applied to your model and keep everything centered.
 Conform to Mari Y as up	button	When toggled on, Mari considers the Y axis in your project as up. On by default.
 Load Invisible Models	button	When toggled on, meshes marked as invisible in the USD file are loaded into Mari. When off, invisible meshes are ignored.
 Create Face Selection Group per Mesh	button	Toggle on to create a selection group for each mesh in the project. For more information on selection groups, see Selection Groups.

Project Settings Dialog

The **Project Settings** dialog box lets you change the colorspace options for your Mari project after project creation. These colorspace options are all presented in the **New Project** dialog when first setting up a project, however, you may wish to change it at a later stage. This dialog provides all the same options and makes the changes on a project-wide basis.

Accessing the Project Settings

• *Menubar* | **File** > **Settings**

Project Settings: About

Description text box	text box	Specifies a description for the project.
Created	string	Specifies the date the project was created.
Created By	date and time	Specifies the name of the artist who created the project.
Modified	string	Specifies the date the project was modified.
Modified By	string	Specifies the name of the artist who modified the project.
UUID	string	Specifies the Universally Unique Identifier (UUID) of the project.

Project Settings: Colorspace

Most of the Colorspace settings have a reset button to revert to the default. The defaults are specified in the OCIO config file. For more information on roles in the OCIO, see Color Management.

Color Management Enabled	checkbox	Toggles colorspace conversions. When switched off all data will be treated as raw, similar to how it was treated prior to Mari 3.0.
OCIO Config	dropdown	The OpenColorIO config file to use. You can choose between the config files packaged with Mari, or select **Custom** to specify a specific config. If the OCIO environment variable is set, that config takes priority

		and this control will be disabled.
Custom OCIO Config	text field, button	If you chose **Custom** for the **OCIO Config** file choice, specify the filepath to the configuration file.
Color Monitor	dropdown	The default colorspace for all user interface elements which are displaying color managed swatches, pickers and thumbnails. This preference does not affect the canvas and image viewers, however it is recommended to set this so that it matches the View Transform. (The active selection can be changed using the View Transform Toolbar .)
Scalar Monitor	dropdown	The default colorspace for all user interface elements which are displaying non-color managed (i.e. scalar) swatches, pickers, and thumbnails. (To understand the difference between color and scalar data in Mari, see Color Data and Scalar Data). This preference does not affect the canvas and image viewers, however it is recommended to set this to match the scalar View Transform. (The active selection can be changed using the View Transform Toolbar .)
Color Picking (advanced and intermediate color management modes only)	dropdown	Default colorspace for all color pickers, swatches, and image viewers. Typically, you'll want to select the same colorspace that your 8-bit color channels default to, in order to achieve perceptually consistent picked colors when switching between channels of different bit depths and colorspaces.
Working (advanced color management mode only)	dropdown	Default for painting, lighting, applying filters, and similar operations. Typically, you'll want to select the same linear-based colorspace as your 16/32-bit float channels to get accurate results.
Blending (advanced color management mode only)	dropdown	Default for applying blending operations such as: a comp of the paint buffer and the current paint target, blending between layers, and combining inputs in a merge node. Typically, you'll want to set this to **Same as 8 bit Data** to get the same results as painting prior to Mari 3.0, and to get less banding artifacts when painting 8-bit channels and images.

8-bit Color	dropdown	Default colorspace for 8-bit channels, and reading and writing image files encoded with 8-bit data like JPGs, PNGs or TIFFs.
16-bit Integer Data	dropdown	The default colorspace for reading and writing image files which are encoded with 16-bit integer data such as 16-bit TIFFs.
8/16-bit Scalar	dropdown	Default colorspace for masks, heights, normals, depths and, generally any non-color image or channel with 8-bit or 16-bit scalar data.
16/32-bit Float Data	dropdown	Default colorspace for 16-bit or 32-bit channels, and reading and writing image files encoded with 16-bit or 32-bit floating point data like EXRs.

Project Settings: Settings

Backup Path	string	The project backup location on a local or network disk. **Note:** This is a separate location to the existing Project Location and Cache location. For more information, see Using Automatic Project Backup & Restore.
Export Root Path	string	The location for saving channel exports. This is inherited by the Export Manager. For more information, see Export Manager Dialog.

Manage Toolbar Dialog

The **Manage Toolbar** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you add, edit and remove toolbars for quick access to Mari functions.

Click **Add** to add a new toolbar, **Remove** to remove the selected toolbar. Drag icons for actions listed to and from existing toolbars.

How to Access It

• *Menubar* | **Edit** > **Toolbars**

Manage Toolbars Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Search Actions	text field, dropdown	Search for actions in the list below to quickly find a particular action shortcut or information.		
	button	Clear the search bar for the **Search Actions** field.		
	button	Go back to the previous search result.		
	button	If on a previous result, go forward to the next search result.		
	button	Reset all shortcuts to their default entries.		
	button	Import custom shortcuts from outside Mari.		
	button	Export custom shortcuts to an external source.		
Lookup Shortcut	text field	Enter the shortcut to look up what action the shortcut is set to.		
(actions) | **Name**	information	The name of the action.		

Control	Type	What it does	Opens this dialog box	Notes
(actions) | **Description**	information	A brief description of the action.		
(actions) | **Shortcut**	information, entry	Displays assigned keyboard shortcuts.		
(toolbars)| **Visible**	checkbox	Whether the action is currently displayed.		Grouped by Default (the standard toolbars that come with Mari) and Custom (user-added toolbars).
(toolbars)| **Add**	button	Add a new toolbar.	Make New Toolbar Dialog	Opens a dialog box to name the new toolbar.
(toolbars)| **Remove**	button	Remove selected custom toolbar.		

Manage Keyboard Shortcuts Dialog

The **Manage Keyboard Shortcuts** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you add, edit, and remove keyboard shortcuts for quick access to Mari functions.

Click in the **Shortcut** column and type the keyboard shortcut for a function. A 'collision' icon displays if that shortcut has already been assigned to another function.

How to Access It

• *Menubar* | **Edit** > **Shortcuts**

Edit Shortcuts Dialog Fields

Column	Type	What it does	Notes
Search Actions	text field, dropdown	Search for actions in the list below to quickly find a particular action shortcut or information.	
	button	Clear the search bar for the **Search Actions** field.	
	button	Go back to the previous search result.	
	button	If on a previous result, go forward to the next search result.	
	button	Reset all shortcuts to their default entries.	
	button	Import custom shortcuts from outside Mari.	
	button	Export custom shortcuts to an external source.	
Lookup Shortcut	text field	Enter the shortcut to look up what action the shortcut is set to.	
Apply to:	dropdown	Select the scope for which to configure shortcuts. You can choose between Node Graph-specific shortcuts (**Node Graph**) and application-wide shortcuts (**All**).	
(actions) | **Name**	information	The name of the action.	
(actions) | **Description**	information	A brief description of the action.	
(actions) | **Shortcut**	information	Displays assigned keyboard shortcuts. Click and type over new shortcuts.	

Note: To know more about customizing keyboard shortcuts, see Customizing General Keyboard Shortcuts.

HUD Manager Dialog

The **HUD Manager** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you select what appears on the "Head-Up Display" (HUD) that shows information about Mari and the current project in the top left-hand corner of the canvas. For each piece of information, the **HUD Manager** shows its:

- **Name** - displayed hierarchically.
- **Type** - QString, int, bool, qint64, float, PaintSelectedMarkFlag, MriColor, PaintOnFlag, DisplayQuality, MriGradient, LightingMode, Qdate, MriFileList, Background, MriPaintMode.
- **HUD** - a checkbox for whether to include in the HUD.
- **Control** - this is debug information, and can be ignored.

How to Access It

- *Menubar* | **Edit** > **HUD Manager**

HUD Manager Dialog Fields

Control	Type	What it does	Notes
Shader Network Glsl Code Generator Manager			
General | **GLSL Generator**	QString	Shows the GLSL for the shader network.	
Project			
Data | **Texture Image Count**	int	Shows the number of texture images (patches) present.	
Data | **Total Image Count**	int	Shows the total number of images present.	
Data | **Image Tile Count**	int	Shows the total number of image tiles	

Control	Type	What it does	Notes
		present.	
Data | **DDI Hash Count**	int	Debug info.	
Project | **Current Modular Shader**	QString	Shows the name of the current shader network.	
Project | **Current Object**	QString	Shows the name of the object currently selected.	
Project | **Current ImageSet**	QString	Shows the name of the image set currently selected.	
Canvas			
Rendering | **Deferred Texture Array**	bool	Debug info.	
Rendering | **Deferred Shading**	bool	Debug info.	
Rendering | **Selected Patches**	int	Shows the number of patches currently selected.	
Rendering | **Triangles**	int	Shows the number of triangles on the model.	
Rendering | **Current Uniforms**	int	Shows the number of GLSL uniforms for the currently selected shader.	
Rendering | **Uniforms**	int	Shows the maximum number of GLSL uniforms used to draw the canvas.	
Rendering | **FPS**	float	Shows the current Frames Per Second of the display. This shows how fast the display is rendering, and how smoothly movement displays.	
Rendering | **FPAnim**	int	Shows the number of frames per animation.	
Tool Manager			
General | **Tool Help**	QString	Shows basic tool help for the current tool.	

Control	Type	What it does	Notes
General | **Current Tool**	QString	Shows the name of the current tool.	
Gpu 0			
Capabilities | **Global Cubemap Seams Allowed**	bool	Debug info.	
Capabilities | **Threaded Operations Allowed**	bool	Debug info.	
Capabilities | **Cubemap Arrays Allowed**	bool	Debug info.	
Capabilities | **Tessellation Allowed**	bool	Debug info.	
Capabilities | **Geometry Shaders Supported**	bool	Debug info.	
Capabilities | **Offscreen AA**	bool	Debug info.	
Capabilities | **Float Data**	bool	Debug info.	
Capabilities | **Offscreen**	bool	Debug info.	
Capabilities | **Shaders**	bool	Debug info.	
Capabilities | **Max Texture Size**	int	Debug info.	
Capabilities | **Texture Units**	int	Debug info.	
Driver | **Renderer**	QString	Debug info.	
Driver | **Glew**	QString	Debug info.	
Driver | **Version**	QString	Debug info.	
Driver | **Vendor**	QString	Debug info.	
Resources | **Max Uniforms**	QString	Debug info.	
Resources | **Memory Info**	QString	Shows the memory usage and free memory information for the project.	

Control	Type	What it does	Notes
Resources| **Used Texture Units**	int	The number of texture units used in the project.	
Resources | **Free Offscreen**	int	Debug info.	
Resources | **Offscreen Buffers**	int	Debug info.	
Resources | **Gpu Ram**	qint64	Debug info.	
Resources | **Max Image Memory**	qint64	Debug info.	
Resources | **Buffer Memory**	qint64	Debug info.	
Resources | **Tile Memory**	qint64	Debug info.	
Resources | **Image Tiles**	int	Debug info.	
Channels			
Current Channel Path	QString	Show the path of the currently selected channel.	
Current Channel	QString	Shows the name of the current channel on the current object.	
Colorspace			
Color Swatches and Pickers	QString	Displays what colorspace is used for the color swatches and color pickers in the project, as defined by the **Color** > **Color Swatches and Pickers** field in the Preferences.	
3D LUT Size	QString	Displays what size the 3D LUT is, as defined by the **Color** > **3D LUT Size** field in the Preferences.	
Current OCIO Config	QString	Displays the OCIO config file name that is being used for the project.	
Current Colorspace	QString	Displays the current colorspace for the	

Control	Type	What it does	Notes
		selected imageset.	
Current Working Colorspace	QString	Displays the current colorspace for 16- and 32-bit data.	
Current Output Colorspace	QString	Displays the current colorspace for the output.	
Layers			
Current Layer Path	QString	Shows the path of the currently selected layer.	
Current Layer	QString	Shows the name of the current layer in the current channel and object.	
Shaders			
Current Shader	QString	Shows the name of the shader currently selected.	
Clock			
Animation Frame	int	The frame number currently displayed, if applicable.	
Animation Loop	int	The number of frames currently being looped.	
Animation FPS	int	Frames Per Second of the animation.	
Animation Time	QString	Time length of the animation in the scene.	
<Project>			
General | **Description**	QString	Shows description of current project.	
Creation | **Created By**	QString	Displays the login name of the user who originally created the project.	
Creation | **Creation Date**	QDate	Displays the initial creation date of the project.	

Control	Type	What it does	Notes
Debug | **Unique ID**	QString	Debug info.	
Editing | **Last Edited**	QDate	Displays date that the project was last edited.	
Editing | **Last Editor**	QString	Displays user who last edited the project.	
Geometry | **Mesh Creator**	QString	Displays the user who created the geometry file upon which the project is based.	
Geometry | **Mesh Created**	QString	Displays the date and time that the geometry file was created.	
Geometry | **Mesh Path**	QString	Displays the path for the mesh file.	
Colorspace | **OCIO Config**	QString	Displays the OCIO config file name that is being used for the project.	
Colorspace | **Custom OCIO Config**	MriFileList	Displays the Custom OCIO config file name, if one is used for the project. Otherwise, the entry is blank.	
Colorspace | **Monitor**	QString	Displays the **Monitor** colorspace used for the project.	
Colorspace | **Color Picking**	QString	Displays the **Color Picking** colorspace used for the project.	
Colorspace | **8 bit Data**	QString	Displays the **8 bit Data** colorspace used for the project.	
Colorspace | **16 bit Data**	QString	Displays the **16 bit Data** colorspace used for the project.	
Colorspace | **8 bit Scalar**	QString	Displays the **8 bit Scalar** colorspace used for the project.	
Colorspace | **16/32 bit Float Data**	QString	Displays the **16/32 bit Float** colorspace used for the project.	

Mari Preferences Dialog

The **Mari Preferences** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you set preferences for how Mari works.

How to Access It

• *Menubar* | **Edit** > **Preferences**

Bakery Tab

Setting	Type	Details
Tile Size	dropdown	Select the pixel size for the tiles. The baker splits the work into tiles to try to avoid time-consuming blocks of work. Choose from: • **Balance** - Mari makes the best guess on balancing stability and performance based on bake setup complexity. • **Stability** - Mari optimizes the tile size for stability. • **Performance** - Mari optimizes the tile size for performance. • A specific resolution - Choose a resolution if you'd rather adopt a particular setting. A lower resolution is more stable and responsive, while a higher one is more efficient and supports a higher-quality edge bleed.
Timeout	dropdown	Select the time for the Bakery to wait until it should close the work

Setting	Type	Details
		due to a perceived issue. You can turn this **off** or use **TDR** to wait for a percentage of the Windows' Timeout Detection and Recovery (TDR) value. Alternatively, choose a duration in seconds.
Use Memory Extensions	checkbox	Select this to use Vulkan Memory Priority Extensions. You may find the Bakery fails to launch on some older graphics cards, which may be due to the extensions being reported as available but not working. In this case, deselect this option and then restart Mari to apply the setting.
GPU Memory Limit (%)	text field and slider	Adjust the maximum percentage of the total GPU device memory requested per allocation. Lower this value to help prevent errors if the Bakery fails to run.

Color Tab

Setting	Type	What it does	Notes
Color			
Color Management Defaults | **Color Picker Maximum**	text field, slider	The maximum color intensity that can be chosen in the color picker.	From 1 to 65,365; defaults to 4,096.
Color Management Defaults | **Color Profile**	dropdown	Sets the default color profile for displaying colors on your monitor.	
Color Picker | **Auto Convert Working Color**	checkbox	When enabled, the picked color is automatically converted when you switch between Color and Scalar color picking modes to preserve color perception.	
Display | **Center F-Stop**	text field, slider	Sets the center F-Stop value for when the Gain reset button is	

Setting	Type	What it does	Notes
		clicked.	
General | **Color Swatches and Pickers** (advanced and intermediate color management modes only)	dropdown	Specify whether the color swatches and color pickers within Mari are based on the OCIO colorspace for the project (found within the **File** > **Settings**), the colorspace listed within the **Color Manager (View Transform)** palette, or the **Raw** data, which has no colorspace transformation.	
General | **Image Transform Clamping** (advanced and intermediate color management modes only)	dropdown	Specify whether the colorspace transforms that are applied to images are clamped between the values of 0 and 1, with: **Off** - not clamping the values applied to any image at all. Clamping can still occur in places like the **Paint through** tool when either the paint buffer or target image set is 8-bit. **Byte** - clamping only when the colorspace transform is applied to images that are 8-bit. **All** - clamping all images regardless of bit depth.	
General | **Detect Path Colorspace**	checkbox	When reading image files from disk, enable this option to detect the colorspace of the image from the colorspace names found in the image's file name and path.	
General | **Color/ Scalar detection**	checkbox	When enabled, Mari will automatically set the Color/ Scalar	

Setting	Type	What it does	Notes
		data on an imported image based on its naming convention.	
General | **Default Import Colorspace**	dropdown	Sets the default colorspace when importing images into channels, layers, or paint nodes: **Automatic** - assigns the default value for **Colorspace** during the image import process to **Automatic** using the project's **Color Settings**. For example, if your project **Color Settings** have **8-bit Color** set to **mari_int8 (sRGB)**, then the default value for **Colorspace** in the import dialog for any imported 8-bit images is automatically set to sRGB. **Same as Target** - assigns the default value for **Colorspace** during the image import process to match the colorspace of the channel, layer, or paint node the image is being imported into. For example, if your channel's **Colorspace** is set to **linear**, the default value for **Colorspace** in the import dialog for any image imported to this channel is automatically set to **linear**.	See Color Management for more information on colorspace in Mari.
General | **Transform Cache Size**	text field, incrementers	Sets the maximum memory size Mari uses to cache data that has undergone a colorspace transformation.	You can assign a shortcut for the **Transform Cache Size** option in the Manage Keyboard Shortcuts Dialog.

Data Tab

Setting	Type	What it does	Notes
Data			
Autosave | **Frequency (Mins)**	text field, incrementers	Sets how frequently (in minutes) Mari automatically saves your work.	
Channel Cache | **Resolution Mode**	dropdown	Determines if Mari caches at the given channel's resolution or a lower proxy level. Setting this to **Fixed** uses the **Default resolution** and **Type** listed below.	This can speed up the caching process if you don't require the full resolution of the channel to be cached.
Channel Cache | **Default Resolution**	dropdown	If the **Resolution Mode** is set to **Fixed**, Mari caches at the selected resolution.	This can save time when caching large patches if you don't require the full resolution while working.
Channel Cache | **Default Type**	dropdown	If the **Resolution Mode** is set to **Fixed**, Mari caches at the selected bit depth.	This can save time when caching large patches if you don't require the full resolution while working.
Channels | **Export Transform**	dropdown	Specifies which colorspace transform method to use when performing flattening and baking on export: **Accurate** - performs all colorspace transforms on the CPU. This gives the most mathematically correct results but incurs a cost of decreasing the speed of export.	

Setting	Type	What it does	Notes
		Fast - performs all colorspace transforms on the GPU. This maximizes the speed of export, but may introduce errors due to the math in some cases, being an approximation to the true methods. **Automatic** - uses the CPU or GPU, on case-by-case basis, based on the accuracy of colorspace transform itself.	
Channels | **Importer**	dropdown	Sets the mode for the import function. One of: • **Fast Importer** - takes over your Mari session while it imports the textures, but works quicker. • **Background Importer** - works in the background, letting you paint and work on your project, but takes longer.	
Channels | **Exporter**	dropdown	Sets the mode for the export function. One of: • **Fast Exporter** - takes over your Mari session while it exports the textures, but works quicker. • **Background Exporter** - works in the background, letting you paint and work on your project, but takes longer.	
Channels | **Export Mask Stacks**	dropdown	Determines how mask stacks are exported. Either: • **Export Baked Mask Stack and an Invisible Group** - exports a group along with the mask stack in order	

Setting	Type	What it does	Notes
		to group the exported masks. • **Export Baked Mask Stack Only** - exports on the mask stack with nothing else as part of the export.	
Channels | **Autosnapshot**	dropdown	Sets whether automatic snapshots are enabled or disabled. The options available are: • **Disabled** - Disables the Autosnapshot feature. • **Enabled with Limit** - Enables the Autosnapshot feature but limits the number of autosnapshots that can be stored in the Snapshots palette. • **Enabled - Unlimited** - Enables the **Autosnapshot**feature without a limit on the number of autosnapshots that can be stored in the **Snapshots** palette.	
Channels | **Autosnapshot Limit**	text field, slider	When **Enabled with Limit** is selected, Mari limits the number of automatic snapshots that are stored in the **Snapshots** palette to the number specified by the **Autosnapshot Limit**. The maximum value is 20.	
General | **Fix Edge UVs**	dropdown	Sets whether Mari should "fix" any UVs that sit on the edge of the patch. With this on, you can paint models that have UVs at the edge of the patch.	
General | **Channel Input**	dropdown	When creating a new project, sets whether the **New Channel** dialog box adds either:	

Setting	Type	What it does	Notes
		• **Single Channel** - a single channel. • **Channel Presets** - a set of channels that you select from a list of presets.	
Project | **Max Memory**	text field, incrementers	Sets the maximum amount of memory Mari uses for the DDI cache blocks. You can't set this to a value higher than 80% of the memory you have available (rounded to the nearest half-GB). For example, if you have 6GB of memory available, you can't set **Max Memory** to a value higher than 4.5GB. This is to avoid using all the physical memory available.	DDI is the data caching system used in Mari. The **Reset** (**R**) icon resets the maximum memory value.
Project | **Max Write Memory Factor**	text field, slider	Set the how much of the **Max Memory** is used for pending write data. Higher values result in a faster exporting process but may slow rendering. The number corresponds to the percentage of **Max Memory** used. For example, a value of 0.3 means that 30% of **Max Memory** is used to temporarily hold data that is pending to be written to disk.	
Project | **Threaded Compression**	checkbox	Enables or disables multi-threading for compression of data. When enabled, the system-recommended thread count is used. To override the thread count use the	

Setting	Type	What it does	Notes
		environment variable MARI_DATA_BLOCK_COMPRESSION_THREAD_COUNT to specify the desired thread count. Use 0 for the system-recommended thread count. Use a negative number for disabling multi-threaded compression.	
Project | **Delete Unused Patches**	dropdown	Deletes the unused patches in the project: When set to **Off**, unused patches will not be deleted. When set to **Prompt**, the user will be prompted during the garbage collection process. When set to **Automatic**, when running the garbage collection process, unused patches will be deleted.	
Project | **Delete Unused Image Sets**	dropdown	Deletes the unused image sets in the project: When set to **Off**, unused image sets will not be deleted. When set to **Prompt**, the user will be prompted during the garbage collection process. When set to **Automatic**, when running the garbage collection process, unused image sets will be deleted.	
Project | **Cache Space Update Interval**	text field, slider	Sets the interval, in seconds, that the cache drives are checked for	

Setting	Type	What it does	Notes
		sufficient disk space.	
Project | **Project Location**	text field, button	Specifies the location where Mari stores the project data. This can only be modified if the MARI_CACHE environment variable has not been set, and no project is loaded.	Do not delete this directory! If you do, you will lose all your project data and existing work.

GPU Tab

Setting	Type	What it does	Notes
GPU			
Baking and Projection |**Mip-Map Generation**	dropdown	Sets the method used when generating mip-maps for images. **Accurate** - Image data is linearized before down-sampling and then converted back into the native colorspace. This gives the most mathematically correct results but incurs a cost of decreasing the speed of processes such as baking. **Fast** - No linearization is performed and the down-sampling is done in the native colorspace of the image data. This maximizes the speed of processes such as baking, but can introduce errors due to the math being non-linear.	
Baking and Projection | **Render Bucket Size**	dropdown	Sets the maximum tile size for tiled render operations (for example,	

Setting	Type	What it does	Notes
		unprojections). Setting this to a lower value improves accuracy but may result in slower render operations.	
Baking and Projection | **Bake Bucket Size**	dropdown	If baking calculation has to be divided into sections in order to continue efficiently, this specifies the maximum square size of these sections.	Increasing the number of squares per section can increase baking time, but may impact stability.
Canvas | **Deferred Shading**	checkbox	Specifies whether to use deferred shading. This is on by default and should increase painting and navigation speed.	
Canvas | **Deferred Texture Array**	checkbox	Turns **Deferred Texture Array** on/off. When this is active, Mari makes more efficient use of OpenGL.	
Canvas | **Pause Canvas Rendering**	checkbox	Pauses canvas rendering when the mouse cursor is outside the canvas area. This allows for faster interaction with the UI. **Tip:** Enable **FPS** under **Canvas** in **Edit** > **HUD Manager** to see the render frame rate in the canvas.	
General | **Max Image Memory**	text field, incrementers	Sets the maximum amount of additional GPU memory Mari uses for tasks other than painting, for example, filter operations.	
General | **Deferred Quad**	text field,	Sets the quad split count. Increasing	Used in conjunction

Setting	Type	What it does	Notes
Split Count	slider	this value splits the canvas into smaller parts.	with **Draw Batch Size**, adjusting this value can improve stability and prevent Windows driver resets.
General |**Threaded Operations Allowed**	checkbox	If this is enabled then operations requiring threaded OpenGL can be executed.	
General |**Fence Synchronization Allowed**	checkbox	For modern OpenGL (4.4 and later), this preference controls if fences are used when transferring UV maps.	
General |**Cubemap Arrays Allowed**	checkbox	Enabling cubemap arrays allows the use of this OpenGL feature that the Environment Light requires.	
General |**Persistent Mapping Allowed**	checkbox	Creates a persistent framebuffer for rendering.	
General | **Bindless Textures Allowed**	checkbox	Enables the use of bindless textures for hardware that can support them. This allows a greater number of textures to be used on a graphics card.	
General |**Texture Manager Slots**	text input, slider	The maximum number of images on a shader.	
General | **Texture Manager Slots**	text input, slider	Sets the internal Texture Manager Slot count. If a scene has a large number of image based nodes/layers (such as Tiled or Triplanar), and the number of images causes a rendering failure, then increasing this number may	

Setting	Type	What it does	Notes
		help prevent errors. **Note:** Increasing the texture slot number may also incur errors; therefore, it is recommended to leave this preference at the default value.	
Shaders | **Use Cache**	checkbox	If this is enabled, the preference uses already-compiled shaders a second time, otherwise Mari compiles every time shaders, nodes, or layer stacks are changed.	
Shaders | **Tessellation Allowed**	checkbox	Enables the use of tessellation shaders for hardware able to support them.	
Shaders | **Compute Allowed**	checkbox	Enables the use of compute shaders for hardware that can support them. This can speed up rendering by pre-filtering data.	
Shaders | **Compilation Mode**	dropdown	Specifies the strategy to use when compiling GLSL shader programs: **Legacy** - uses the **Optimized** strategy with an older OpenGL technique for passing data into the shader. This matches how Mari operated prior to Mari 3.3v1. **Dynamic** - compiles shaders less frequently and at a faster speed. This may decrease the FPS.	

Setting	Type	What it does	Notes
		Optimized - compiles shaders more frequently and at a slower speed. This may increase the FPS. **Automatic** - uses the **Dynamic** shader while the **Optimized** is being compiled.	
Shadow Maps | **Allowed**	checkbox	Allows the use of shadow rendering in Mari. If the box is ticked, shadow-related features are available in the **Lights** and **Objects** palettes. By default, this preference is disabled.	Depending on your hardware and project requirements, shadow processing may cause a noticeable drop in UI responsiveness. Disabling this option disables shadow processing entirely and improves rendering throughput.
Shadow Maps | **Use Variance Maths**	checkbox	Sets whether variance map maths are used for shadows. If this box is ticked, the boundary of the shadows are softer and less prone to aliasing. For a sharper edge to shadows, unselect this box.	If **Allow Depth Projections** is unselected, shadows do not display. The **Use Variance Maths** checkbox should be unselected as it depends on shadows being used.
Shadow Maps | **Resolution**	dropdown	Sets the texture resolution of each of the array of depth textures used for shadow rendering. The **Depth Projection Texture Array Size** is linked to this resolution and impacts other resolution sizes set in Mari.	

Setting	Type	What it does	Notes
Shadow Maps | **Layer Count**	dropdown	Sets the number of depth projection textures to use for shadow rendering. Increasing the array size increases the graphics card memory made available for shadow rendering. The total graphics memory used for these textures is 8 multiplied by the depth projection resolution, multiplied by the depth projection texture array size. The array size controls how many angles each light of the 4 lights uses. The maximum array size is 24, which allows for 6 angles per light.	Options are available for 25-32, but at this time Mari does not use these additional angles. Selecting options over 24 causes wasted cache space.
Virtual Texture | **Type**	dropdown	Sets the pixel type of the virtual texture that Mari uses. Setting this to **Half** or **Float** gives more precision when painting or previewing displacement but requires significantly more GPU memory.	
Virtual Texture | **Resolution**	dropdown	Sets the size of the virtual textures Mari uses. Larger sizes may give better results but require significantly more GPU memory.	
Virtual Texture | **Layer Count**	text field, slider	Controls the number of 2D texture layers used in the virtual texture Mari uses for display. Increasing this may resolve issues with flickering textures.	This value is dependent on the graphics card capabilities. If you want to enable this preference for Mac OS X platform, go to the Mari User Guide in the Environment

Setting	Type	What it does	Notes
			Variables That Mari Recognizes section.
Virtual Texture | **Bake Resolution**	dropdown	Controls the size of the virtual texture being used for the display. Increasing this setting may resolve issues, such as flickering, but takes up more memory. This is a graphics card-dependent setting.	
Virtual Texture | **Gather Scale**	text field, slider	Sets the buffer size Mari uses when gathering information on which image tiles are required for rendering a scene. The default value of 1.000 halves the canvas resolution and works well in most situations, saving processing time. However, if glitches are appearing in the canvas, you may need to increase this value to fix the problem (at the cost of reduced painting and navigation speed).	
Virtual Texture | **Gather Type**	dropdown	The pixel format the virtual texture system uses to complete a 'meta' render, in order to verify which textures it needs during a full render. The formats are: • **Short** - a short/small integer value; 16 bits. • **Float** - a floating point value; 32 bits.	Note: With some Intel cards on Linux, the **Short** option does not always behave as it should, and it's recommended that you use **Float**.
Virtual Texture | **Thin Triangle Quality**	text field, slider	Controls the texture preview quality of thin triangles. The default value is 0. In cases where the geometry has	

Setting	Type	What it does	Notes
		long thin triangles, increasing this value results in the use of higher resolution textures.	

Misc. Tab

Setting	Type	What it does	Notes
Misc			
Appearance | **Use Custom Font Size**	checkbox	Enabling this allows you to change the font used in Mari from the default setting.	
Appearance | **Custom Font Size**	text field, slider	Adjusts point size of the custom font.	**Use Custom Font** must be enabled in order for this to have an affect.
Appearance | **Tool Bar Size**	text field, slider	Adjusts the tool bar sizes in Mari.	
Appearance | **Slider Precision**	text field, slider	Adjusts how many decimal places are represented by Mari's sliders. Increasing this gives you finer control over slider values.	
Appearance | **Sub-Menu Timeout**	text field, slider	Defines how long before a sub-menu closes (in milliseconds).	
Appearance | **Locator Size**	text field, slider	Adjusts the locator size on the canvas in Mari.	
Default Settings| **Lights Fixed**	incrementor		
Default Settings| **Channel Preset**	dropdown	Choose between: • Last Used	

Setting	Type	What it does	Notes
Category		• 3Delight Principled - Metallic * VFX • 3 Delight Principled - Non-Meta l lie - VFX • Arnold Standard Surface • VFX • Autodesk Standard Surface (1.0.2.x) - VFX • BRDF • Games • BRDF - VFX • General • Principled BRDF - Metallic - Games • Principled BRDF - Metallic - VFX • Principled BRDF * Non-Metallic * Games • Principled BRDF • Non-Metallic - VFX • Unreal * Metallic * Games • Unreal - Metallic - VFX Unreal - Non-Metallic * Games • Unreal * Non-Metallic * VFX • VRayMtl (V-Ray 6) - VFX • VRayMtl - VFX	
Image| **Remember Paths**	checkbox	If enabled, Mari automatically remembers paths previously set for importing and exporting, and offers them again next time you attempt to import or export.	
Image | **Show Import Summary**	checkbox	Sets whether Mari displays an import summary dialog after importing an image.	The import summary dialog displays information such as how many images were imported, how

Setting	Type	What it does	Notes
			long the operation took, and if patches were resized - then how many.
Image | **Show hidden Images**	checkbox	Show images that are hidden by default in the Image Manager.	
Input | **Use Hi-Res Positions**	checkbox	If this option is enabled, Mari uses high resolution position values from tablets and other devices. Most often this is only noticeable using **Paint** and **Vector Paint** on large patch sizes.	In some cases, having this option enabled can hinder painting entirely. Disabling it resolves the issue.
Input | **Zero Pressure Allowed**	checkbox	Enabling this fixes any pressure issues you might experience with the Wacom Intuos 3.	If you aren't experiencing any issues, don't enable this option, as it can introduce other problems under Windows 8.1.
Input | **Use Mouse-Mapping Mode**	checkbox	When using a tablet, determine whether or not to set your pen to behave as a mouse, so that positions are relative and incremental. If this is enabled, you can access screen areas outside of your tablet's reach.	
Named Files | **Patch Cutoff**	text field, slider	Sets the maximum number of patches objects are allowed to have for the **Named Files** tab to appear on the **Import** and **Export Channel** dialogs. You can override this	

Setting	Type	What it does	Notes
		setting using the MARI_NAMEDFILEIMPORT_SEQUENCELIMIT environment variable. You can use the **Named Files** tab to import or export individual files rather than a sequence of patches with the UDIM number (1001, 1002, etc.) in the file names.	
Shelf | **Show Labels**	checkbox	Specifies whether to show item labels in shelves (for example, names for color swatches).	
Shelf | **Shelves Include Presets**	checkbox	Specifies whether to show Mari's preset shelves in the **Shelf** palette. If off, only the **Menu**, **Personal**, **Project** and custom shelves are displayed.	
Shelf | **Double-click to Activate in Pop-over shelf**	checkbox	When enabled, double-click to switch from a floating shelf in the canvas to a brush. When disabled, this action only requires a single click.	
Shelf | **Remake Icons**	button	Regenerates Mari's icons for shelf items. If these have become corrupt (for example, if your brush icons are all blank), clicking this forces Mari to re-create all the icons.	
Toggle Tools | **Tool on Key Held**	checkbox	Sets how the shortcut keys for the **Zoom Paint Buffer** (**Z**), **Eye Dropper** (**C**), and **Select** (**S**) tools work. If **On**, these shortcut keys only apply	

Setting	Type	What it does	Notes
		while you're holding them down. Mari switches back to your previous tool when you let go of the shortcut key. If this is **Off**, tapping the shortcut key quickly switches permanently to the tool, and holding it down for more than a second or two switches to the tool while you're holding the shortcut key (and switches back as soon as you let it go).	
Toggle Tools | **Key Held Threshold Time**	text field, slider	If **Tool on Key Held** is **On**, this property allows you to specify the time, in milliseconds, after which a key is considered to be held for tool toggling purposes.	
Window Corner Dock Areas | **Top Left Corner**	dropdown	Set the main window's top left corner option. • **Top Dock Area** - The top left corner of the main window is owned by palettes docked in the top dock area. • **Left Dock Area** - The top left corner of the main window is owned by palettes docked in the left dock area.	For example, if you want palettes in the left dock area to touch the top and bottom of the Mari window, set **Top Left Corner** to **Left Dock Area**, and **Bottom Left Corner** to **Left Dock Area**.
Window Corner Dock Areas | **Top Right Corner**	dropdown	Set the main window's top right corner option. • **Top Dock Area** - The top right corner of the main window is owned by palettes docked in the top dock area. • **Right Dock Area** - The top right corner of the main window is	For example, if you want palettes in the right dock area to touch the top and bottom of the Mari window, set **Top Right Corner** to **Right Dock Area**,

Setting	Type	What it does	Notes
		owned by palettes docked in the right dock area.	and **Bottom Right Corner** to **Right Dock Area**.
Window Corner Dock Areas | **Bottom Left Corner**	dropdown	Set the main window's bottom left corner option. • **Bottom Dock Area** - The bottom left corner of the main window is owned by palettes docked in the bottom dock area. • **Left Dock Area** - The bottom left corner of the main window is owned by palettes docked in the left dock area.	For example, if you want palettes in the left dock area to touch the top and bottom of the Mari window, set **Top Left Corner** to **Left Dock Area**, and **Bottom Left Corner** to **Left Dock Area**.
Window Corner Dock Areas | **Bottom Right Corner**	dropdown	Set the main window's bottom right corner option. • **Bottom Dock Area** - The bottom right corner of the main window is owned by palettes docked in the bottom dock area. • **Right Dock Area** - The bottom right corner of the main window is owned by palettes docked in the right dock area.	For example, if you want palettes in the right dock area to touch the top and bottom of the Mari window, set **Top Right Corner** to **Right Dock Area**, and **Bottom Right Corner** to **Right Dock Area**.

Navigation Tab

Setting	Type	What it does	Notes
Navigation			
General | **Momentum Enabled**	checkbox	If this is enabled, the model has momentum, so when you let go of the mouse button, it slowly coasts to a	

Setting	Type	What it does	Notes
		stop. If this is disabled, the model only moves when you are holding the mouse button down.	
General | **Minimum Flick Speed**	text field, slider	Sets the minimum speed at which you can move your model around. The higher this is set, the more sensitive the movement controls are.	
General | **Friction**	text field, slider	Affects the momentum of your model as you move it around the canvas.	Decreasing this value allows movement to retain its momentum for longer. Increasing the value arrests momentum faster.
General | **Control Type**	dropdown	Sets up the Mari controls on the canvas to match the controls of another application, specified in the dropdown list. Applications include: 3ds Max, Houdini, Lightwave, Mari, Maya, Modo, and Nuke.	
Orbit | **Lock To World Up**	checkbox	When enabled, the y-axis is locked to world up view.	
Orbit | **Center Mode**	dropdown	Specifies the center point to orbit around from the choice of: • **Look At** - what is currently being viewed in the display. • **Objects** - the objects in a scene. • **Origin** - the scene origin point. • **Selection** - the current selection.	
Roll | **Circular Motion Enabled**	checkbox	If enabled, rolling is achieved by making a circular motion in the center of the canvas. If this option is disabled, rolling is only	

Setting	Type	What it does	Notes
		done with a horizontal motion across the canvas.	
Trackball | **Center Mode**	dropdown	Specifies the center point to orbit around from the choice of: • **Look At** - what is currently being viewed in the display. • **Visible Object center** - the center of the visible object in a scene. • **Origin** - the scene origin point. • **Selection center** - the center of the current selection.	
Trackball | **Sensitivity**	text field, slider	The sensitivity of the trackball.	
Trackball | **Size**	text field, slider	The size of the trackball as a percentage.	
Transform | **Stepping Angle**	text field, slider	Specifies the stepping angle for the rotation of objects and locators in degrees. **Tip:** To snap rotation while the **Transform Selected Objects** tool is active, hold **CTRL** and rotate.	

Node Graph Tab

Setting	Type	What it does	Notes
Node Graph			
General |**Use Cache for**	checkbox,	If enabled, Mari uses the cached	For more information

Setting	Type	What it does	Notes
Bake	reset	result of a node/layer for baking process such as "Flatten", "Merge Layers" and "Bake to Paintable". Mari does not process layers below the cached layer or upstream nodes of the cached node, but uses the cached result.	about caching nodes, see Working with Node Properties.
General | **Control Type**	dropdown	Set whether you want to zoom and pan in the Node Graph with the **Mari** or **Nuke** navigation style.	
General | **Default Port Display**	dropdown	Allows you to specify how the node ports connect in the Node Graph. Nodes can be read from left to right or top to bottom. You can select from the following navigation types: • **Show Port List (Horizontal)**: Mari uses the left-to-right navigation. • **No Port List (Vertical)**: Mari uses the top-to-bottom navigation.	Switching between these navigation types does not alter existing nodes, it only changes the nodes you create after having switched the navigation. You can switch the navigation type of a selected node or multiple selected nodes by using **Shift**+**S**.
General | **Auto Insert Node**	checkbox, reset	If checked, when you drag a node over an existing pipe, Mari auto-inserts the node between nodes. If unchecked, when you drag a node over an existing pipe, Mari does not auto-insert the node between nodes.	
General | **Zoom To**	dropdown	Allows you to set whether to zoom according to the **Cursor**'s position or	

Setting	Type	What it does	Notes
		the **Center** of the Node Graph.	
General | **Update Blending Colorspace**	dropdown	Control whether to automatically change the **Use Blending Colorspace** attribute to **Enable** for Merge nodes upstream of newly connected 8bit (byte) Channels, or Bake Points configured for color paint data. **Note:** When Merge nodes have this attribute set to **Disable** for an 8-bit (byte) color paint data channel, precision issues in the blending math could occur, which could result in visual banding artifacts in your paint. Choose between: • Prompt • Change • Leave	
Grid | **Show Grid**	checkbox	Enables/Disables the visibility of the node graph grid.	Unchecked by default.
Grid | **Snap to Grid**	checkbox	Enables/Disables node snapping on the node graph grid.	Unchecked by default.
Grid | **Grid Height**	numeric field	Determines the height of grid cells in the node graph grid.	Defaults to 50.
Grid | **Grid Width**	numeric field	Determines the width of grid cells in the node graph grid.	Defaults to 50.

Setting	Type	What it does	Notes
Grid | **Grid Lines Color**	swatch	Specify the color of the node graph grid lines using a color picker.	
Grid | **Grid Lines Width**	numeric field	Determines the line weight of the node graph grid.	Defaults to 1.
Grid | **Node Graph Background Color**	swatch	Specify the background color of the node graph grid using a color picker.	
Multi-Channel | **Default Collapsed State**	dropdown	Allows you to select a default collapsed state for Multi-Channel nodes to be displayed with in the Node Graph.	
Node Category Colors | **Basic**	color swatch	Allows you to set a color for all new and existing **Basic** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Environment**	color swatch	Allows you to set a color for all new and existing **Environment** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Filter**	color swatch	Allows you to set a color for all new and existing **Filter** nodes. The default color is: ■ **RGB : (0.533, 0.624, 0.788)**	
Node Category Colors | **Geometry**	color swatch	Allows you to set a color for all new and existing **Geometry** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors |	color swatch	Allows you to set a color for all new	

Setting	Type	What it does	Notes
Layer		and existing **Layer** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Lighting**	color swatch	Allows you to set a color for all new and existing **Lighting** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Math**	color swatch	Allows you to set a color for all new and existing **Math** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Misc**	color swatch	Allows you to set a color for all new and existing **Misc** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Procedural**	color swatch	Allows you to set a color for all new and existing **Procedural** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Surface**	color swatch	Allows you to set a color for all new and existing **Surface** nodes. The default color is: □ **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **System**	color swatch	Allows you to set a color for all new and existing **System** nodes.	

Setting	Type	What it does	Notes
		The default color is: **RGB : (0.918, 0.918, 0.918)**	
Node Category Colors | **Paint**	color swatch	Allows you to set a color for all new and existing **Paint** nodes. The default color is: **RGB : (0.976, 0.475, 0.412)**	For more information about the **Paint** node type, see Paint Node.
Node Category Colors | **Teleport Broadcast**	color swatch	Allows you to set a color for all new and existing **Broadcast** nodes. The default color is: **RGB : (0.933, 0.612, 0.102)**	For more information about the **Teleport** nodes, see Teleport Broadcaster and Teleport Receiver Nodes.
Node Category Colors | **Teleport Receiver**	color swatch	Allows you to set a color for all new and existing **Receiver** nodes. The default color is: **RGB : (0.976, 0.475, 0.412)**	
Node Category Colors | **Paint**	color swatch	Allows you to set a color for a Paint node that isn't currently active. The default color is: **RGB : (249, 121, 105)**	
Node Category Colors | **Current Paint**	color swatch	Allows you to set a color for the Paint node that is currently active. The default color is: **RGB : (1.000, 0.698, 0.553)**	For more information about the **Paint** node type, see Paint Node.
Node Category Colors | **Multi-Channel Paint**	color swatch	Allows you to set a color for a Multi-Channel Paint node that isn't currently selected.	

Setting	Type	What it does	Notes
		The default color is: **RGB : (201, 107, 254)**	
Node Category Colors | **Current Multi-Channel Paint**	color swatch	Allows you to set a color for the Multi-Channel Paint node that is currently selected. The default color is: **RGB : (219, 151, 255)**	
Quick Node Create Menu | **Number of Most Recent nodes**	Text field, incrementer	Defines how many recent nodes appear in the quick Node create menu .	
Shake to Extract | **Shake to Extract Selected Nodes**	checkbox	Turn on the ability to shake selected nodes to extract them from the node graph. You extract a node by shaking it **Shakes to Trigger times** within **Shakes in Time** milliseconds. On by default.	
Shake to Extract | **Shakes to Trigger**	text field	Set how many complete shakes are needed to extract. Default is 4.	
Shake to Extract | **Shake to Extract Selected Nodes**	text field	Set the time limit for the shakes to take effect in milliseconds. Default is 500.	

Nuke Tab

Setting	Type	What it does	Notes
Nuke			
General | **Use Flat Lighting**	checkbox, reset	Modifies the Nuke workflow to use flat lighting.	The reset icon resets the flat lighting option back to default.
Server | **Nuke Host**	text field, reset	Allows you to specify the host name for the location of Nuke.	
Server | **Port**	text field, slider, reset	Allows you to specify the port number, or modify it with the slider.	
Server | **Nuke Launch Path**	text field	Allows you to browse and load a launch path for accessing Nuke files.	The path can be written manually or inserted using the browse and load icon.

Painting Tab

Setting	Type	What it does
Painting		
Cursor | **Shape**	dropdown	Allows you to set whether the brush cursor in Mari is displayed as an outline of the selected brush tip or as a standard round paint cursor.
Cursor | **Opacity**	text field, slider	Allows you to set the opacity of the brush cursor. This only affects how opaque the cursor design is, not the opacity of the paint.
Pressure | **Mode**	dropdown	Allows you to choose the method used to apply pressure sensitivity to brush strokes when painting.

Setting	Type	What it does
		• **Tablet** - if you are using a tablet, this option allows you to control the pen pressure by the pen tip of the tablet being used. Pressing harder increases the percentage of pressure applied. If your tablet does not provide pressure control then 100% pressure is applied. • **Key** - this option allows you to increase the pen pressure using the **Spacebar** on your keyboard. As you are painting, hold the **Spacebar** down to increase the pressure, and release the **Spacebar** to decrease the pressure. • **Wheel** - pressure is applied at a fixed value that can be increased and decreased by scrolling the mouse wheel, or using the touch ring on your tablet. As you are painting, scroll the mouse wheel up to increase the pressure, and scroll down to decrease the pressure.
Pressure | **Wheel Scale**	text filed, slider	This option is enabled when the pressure **Mode** is set to **Wheel**. This value determines the number of rotations of the mouse wheel it takes to go from minimum to maximum pressure, or from maximum to minimum pressure. For example, if you decrease the value to **0.100**, you only need to scroll the wheel one tenth of a rotation for the pressure to increase or decrease fully. If you increase this value to **2.000**, you need to scroll the mouse wheel two rotations for the pressure to increase or decrease fully.
Pressure | **Key Interval**	text field, slider	This option is enabled when the pressure **Mode** is set to **Key**. This value determines the number of seconds it takes to go from minimum to maximum pressure, or from maximum to minimum pressure, while holding or releasing the **Spacebar**. For example, if you decrease the value to **0.100**, you only need to hold the **Spacebar** for one tenth of a second for the pressure to increase or decrease fully. If you increase this

Setting	Type	What it does
		value to **180**, you need to hold the **Spacebar** for 180 seconds for the pressure to increase or decrease fully.
Pressure | **Key Curve**	spline editor	This option is enabled when the pressure **Mode** is set to **Key**, and allows you to define the gradient curve followed when changing the pressure. Click the **Modify the curve using the spline editor** button to open the **Curve Editor** and adjust the curve controls to set how the pressure falls off between the Start and End points. The start point at the left of the curve represents the minimum pressure, and the end point at the right represents the maximum pressure. The amount of pressure is on the vertical axis.
Projection | **Default Color Depth**	dropdown	Sets the default color depth Mari uses: 8, 16, or 32-bit.

Paths Tab

Setting	Type	What it does	Notes
Paths			
Channel Presets | **Search Paths**	list, button	Adds or removes directories that Mari checks for channel preset configuration files.	See the *Mari User Guide* for details on writing your own channel preset files.
Environments | **HDR Paths**	list, button	Adds or removes directories that Mari checks for environment light images. For each directory provided, a Thumbnails sub-directory within it is looked for that should contain smaller versions of the images, with	When Mari scans the path, it also searches for image thumbnails to use from the **Thumbnails** directory.

Setting	Type	What it does	Notes
		matching filenames, for use by the UI.	
Materials | **Ingest Template Paths**	list, button	Adds or removes directories that Mari checks for **.mma** files to be used as material ingest templates. Any found **.mma** files in the given paths appear in the **Material Template** dropdown of the **Material Ingest Tool**.	See Creating Materials with the Material Ingest Tool. Any paths defined in the **Ingest Template Paths** can be overridden by the following environment variable: **MARI_MATERIAL_INGEST_TEMPLATES_SEARCHPATHS**.
Megascans| **Search Paths**		Adds the directory that Mari checks for Megascans assets.	See Texture Sets Palette.
Shaders | **Search Paths**	list, button	Adds or removes directories that Mari checks for compositing shaders.	

Ptex Tab

Setting	Type	What it does	Notes
Ptex			
Face Textures | **Minimum Size**	dropdown	Sets the minimum face texture size. The options provide increments in powers of two, and allow you to clamp face texture sizes during project creation and face resizing.	The default state is 1 for quadrangular textures, and 2 for triangular textures.

Scripts Tab

Setting	Type	What it does	Notes
Scripts			
Console | **Font Size**	text field, slider	The point size of the default Mari font in the Python console.	
Mari Command Port | **Port**	text field, slider	Sets the port number that Mari receives commands from Nuke through. The default value is 6100.	
Mari Command Port | **Enabled**	checkbox	If checked, Mari can receive commands through the specified port.	The default state is disabled.
Mari Command Port | **Local Host Only**	checkbox	If unchecked, Mari can listen for connections to its command port from any machine. If this is checked, then Mari only listens for connections from the local machine. Only allowing connections from the local host is more secure but prevents you from using features, such as the Nuke<>Mari Bridge.	The default state is enabled.

Thumbnail Tab

Setting	Type	What it does	Notes
Geo-Channel Properties			
Channels	button	Adds a new geo-channel to the mesh object defined in **Mesh File**. The geo-channel drives defined mesh maps such as ambient occlusion, and	

		curvature into the object in the exported material thumbnail.	
I	button	Imports an image set to the geo-channel. **Note:** If the imageset was produced using the Modo Bake Presets, information about the bake preset is displayed in a tooltip when hovered over the imageset.	
R	button	Resets the imageset assigned to the geo-channel.	
-	button	Removes the geo-channel.	
Scene			
Mesh File	text field, button	Specifies the mesh file to be used in generating material thumbnails during material export.	
Env Light File	text field, button	Specifies the environment light to be used in generating material thumbnails during material export.	
Projector File	text field, button	Specifies the projector to be used in generating material thumbnails during material export.	

USD Tab

Setting	Type	What it does	Notes
General | **Load USD Plugin**	checkbox	Defines whether to load the USD plugin	

Display Properties Dialog

The **Display Properties** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to set how Mari displays the project - the background behind the model, the overlay used on locked patches, whether to show wireframes on the model, and so on.

How to Access It

• *Right-click* | main canvas > **Display Properties**

Display Properties Dialog Fields

Control	Type	What it does	Notes
Canvas > HUD			
HUD	checkbox	Shows the Heads Up Display information in the background of the canvas (tool tips, etc).	
Canvas > Background			
Image	text field, button	Select the file to display as the project background in the canvas. If not set, the colors in the **Bottom** and **Top** fields are used for the background.	
Bottom	swatch	Sets the color to display at the bottom of the screen. To use a single color in the background, set both the **Bottom** and **Top** to the same value.	

Control	Type	What it does	Notes
Top	swatch	Sets the color to display at the top of the screen.	
Canvas > Environment			
Show Environment	checkbox	If an **Environment** light image is set to display as a background on the canvas, this control applies as a global setting for the canvas so that it can be turned on or off separately.	
Canvas > Grid			
Color	swatch	Sets the color for the onscreen reference grid.	
Size	text field	How large (in squares) the onscreen reference grid is.	
Visible	checkbox	Shows the reference grid in the background of the onscreen display.	This does not affect the UV view.
Canvas > Wireframe			
Color	swatch	The color of the wires displayed in the wireframe overlay.	
Visible	checkbox	If this is selected, the model in the 3D views are displayed with the wireframe visible on top of the paint.	
Canvas > Camera Mask			
Opacity	text field, slider	The opacity of the black bars that indicate the aspect ratio of the perspective camera in the Canvas.	By default, this is set to 0, which means the bars are disabled.
Canvas > Mirror Plane			
Color	swatch	The color of the Mirror Plane or Mirror	

Control	Type	What it does	Notes
		Line.	
Painting Style	dropdown	Specifies the appearance of the Mirror Plane: **Mirror Plane** or **Mirror Line**.	
Canvas > Linked Patches			
Color	swatch	The color of the patches that are linked together.	
Canvas > Rendering			
Checker Color A	swatch	Sets color A of the transparency checkerboard. This can be selected using the color picker.	
Checker Color B	swatch	Sets color B of the transparency checkerboard. This can be selected using the color picker.	
Checker Size	text field, slider	The size of the individual checkerboard squares. This can be adjusted using either the slider or the text field.	By default this is set to 10.
Render Camera	checkbox	Show the cameras on the canvas.	
Render Light	checkbox	Show the lights on the canvas.	
Render UV Image	checkbox	Show the patches in the UV view as they appear after export, including the overspill areas.	
Canvas > UV Label			
Text Color	swatch	The color for the onscreen text on the UV view, used to show the UV patch information.	Only visible if you have opened the dialog box from within the UV view.
Visible	checkbox	Whether to show the onscreen UV	Only visible if

Control	Type	What it does	Notes
		patch information text over each patch in the UV view.	you have opened the dialog box from within the UV view.
Canvas > Selection			
Fill Render	dropdown	When to show the selection fill color. This appears in the middle of selected areas. Options are: • **Always** - show the selection fill the whole time the area is selected. • **Select** - show the selection fill while the area is being selected. Once you've finished making the selection, the fill disappears. • **Never** - do not show the selection fill.	
Fill Color	swatch	The color to use to fill selected areas. Click on the swatch to set the color.	
Outline Render	dropdown	When to show the selection outline. This appears around the outside of selected areas. The options are the same as for **Fill Render**, above.	
Outline Color	swatch	The color to use to outline the selected areas. Click on the swatch to set the color.	

Screenshot Settings Dialog

The **Screenshot Settings** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to specify the settings for any screenshots that you take of the Mari canvas. If these settings have never been adjusted before, the dialog appears the first time you attempt to take a screenshot from the **View** menu. However, you can change these settings at any time.

How to Access It

• *Menubar* | **View** > **Screenshot Settings**

Screenshot Settings Dialog Fields

Control	Type	What it does	Notes
Output	text field, dropdown	The directory where the screenshots are to be saved. Type the directory or click the file browser button to navigate to it directly.	
	button	Browse to the directory where you want the screenshots to be saved.	
Incremental	dropdown	If **Enabled**, multiple screenshots are saved with incremental numbers appended to the filename. If **Disabled**, every screenshot overwrites the previous one.	
Background	dropdown	Sets whether the background of the screenshot is **Gray** or **Transparent**.	

Image Dialogs

Select Color Dialog

The **Select Color** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

Note: **Value** can go beyond 100% if picking colors from an HDR image and **Saturation** can go beyond 100% if picking colors from a Channel in a colorspace with a wider gamut than the color picking colorspace.

Note: Picking values for HDR images outside of the range as defined by the color picking OCIO colorspace, may result in clamped values.

What It Does

The **Select Color** dialog box lets you view and change colors. The controls on this palette all work together - as you change the color through one control, the other controls update to show your change.

How to Access It

- *Tools toolbar* | **Foreground** color swatch
- *Tools toolbar* | **Background** color swatch
- *Colors palette* | **Foreground** color swatch
- *Colors palette* | **Background** color swatch
- *Add Channel dialog* | **Color** color swatch

Select Color Dialog Fields

Control	Type	What it does	Notes
Swatches	swatch	These swatches store the 6 most	The current foreground color is the

Control	Type	What it does	Notes
		recently selected colors and the current foreground color (the one at the top). When a new color is selected, it is added to the top of the list. Clicking on a swatch selects it as the current color (foreground color).	color that Mari uses when you paint using the **Paint** tool.
	button	Adds any of the following tabs: • **Color History** • **Picker** • **Colorspace** • **Sliders** • **Values** • **Image** • **Blend** • **Gray** • **Analyzer**	
	button	Opens tabs in separate windows.	
	button	Deletes the currently displayed tab.	
Picker			
Color field	field	Click to select a color.	
Color slider	vertical slider	Click and drag up and down to select a color.	
Transparency/Alpha slider	vertical slider	Click and drag up and down to select the level of transparency.	
H S V	radio buttons	Select whether to display **Hue**, **Saturation**, **Values**, **Red**, **Green**, or **Blue** in the Color field and the Color slider.	

Control	Type	What it does	Notes
R G B			
Color swatch	swatch	The current foreground color.	The current foreground color is the color that Mari uses when you paint using the **Paint** tool.
Intensity	input box, horizontal slider	Enter a value in the input field or click and drag from left to right to adjust the intensity of colors to select HDR colors. HDR colors are considered to be any color with an intensity that is greater than 1.	From 0 to 4096; defaults to 4096. The maximum range of the **Intensity** slider can be adjusted in the Mari Preferences Dialog under **Color** > **Color Management Defaults** | **Color Picker Maximum**.
Values			
Source (sRGB)			
	text field, slider	Specify **Hue** values in degrees.	Range: 0 to 360°
	text field, slider	Specify **Saturation** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Value** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Red** values.	Range: 0 to 255
	text field,	Specify **Green** values.	Range: 0 to 255

Control	Type	What it does	Notes
	slider		
	text field, slider	Specify **Blue** values.	Range: 0 to 255
	text field, slider	Specify **Alpha** values.	Range: 0.0 to 1.0
Target (linear)			
	text field, slider	Specify **Hue** values in degrees.	Range: 0 to 360°
	text field, slider	Specify **Saturation** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Value** values in percentage.	Range: 0 to 100%
	text field, slider	Specify **Red** values.	Range: 0 to 255
	text field, slider	Specify **Green** values.	Range: 0 to 255
	text field, slider	Specify **Blue** values.	Range: 0 to 255
	text field, slider	Specify **Alpha** values.	
Image			

Control	Type	What it does	Notes
Image field	field	Drag-and-drop an image in the image field then click the image to select a foreground color. You can also load an image by right-clicking and selecting **Load Image**, then selecting a file from the **Load** dialog box. To delete the image, right-click it and select **Clear Image**.	**Note:** You can drag-and-drop an image from the Image Manager Palette or from a file browser.
Gray			
Gradient, presets	tab: sliders, buttons	Use this to pick a gray value from the gradient, or one of the presets.	
Intensity	input box, slider	Enter a value in the input field or click and drag from left to right to adjust the intensity of colors to select HDR colors. HDR colors are considered to be any color with an intensity that is greater than 1.	From 0 to 4096; defaults to 4096. The maximum range of the **Intensity** slider can be adjusted in the Mari Preferences Dialog under **Color** > **Color Management Defaults** | **Color Picker Maximum**.
Blend			
Color field	field	This displays a field with four colors blending into each other. Click any of the color swatches (foreground color or recently used colors) and drag-and-drop them to any corners of the color field. Then click in the color field to pick	

Control	Type	What it does	Notes
		from the blend.	
Analyzer			
Current	swatch	Displays the color of the pixel selected with the **Eye Dropper** tool.	
Minimum	swatch	Displays the minimum color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	
Average	swatch	Displays the average color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	
Maximum	swatch	Displays the maximum color of the pixels that have been selected with the **Eye Dropper** tool (when **Accumulate** is enabled).	
Accumulate	checkbox	When selected, keeps data on all pixels selected (until **Reset**), and calculates **Minimum**, **Average** and **Maximum** colors.	
Reset	button	Clears all accumulated pixel selections, to just the Current selection.	
Values | **R, G, B, A**	information	Values for Red, Green, Blue, and Alpha.	
Values | **H, S, V, L**	information	Values for Hue, Saturation, Value, and Luminosity.	
Values | **X, Y E, I**	information	Values for position of pixel on x and y axis, and the pixel Exposure	

Control	Type	What it does	Notes
		and Intensity.	
Colorspace			
Colorspace	dropdown	The colorspace in which the colors are interpreted. **Same as 8 bit Data** is determined by the **Color Picking** control in the **Project Settings**, or the same control set at project creation. OCIO colorspace roles are also displayed in this list.	This colorspace can be reset to the default option by clicking R. The **Colorspace** menu is linked to the **Source** column in the **Values** tab. See **Color Swatches and Pickers** preference in the Mari Preferences Dialog
Range	dropdown	Set color values to either **Byte** (8 Bit), **Half** (16 Bit), or **Float** (32 Bit) .	
H S V	input box, slider	Enter a value in the input field or click and drag from left to right to select Hue, Saturation, and Value.	Values can go beyond 100% if picking colors from a HDR image and Saturation can go beyond if picking colors from a Channel in a colorspace with a wider gamut than the color picking colorspace.Values can go beyond 100% if picking colors from a HDR image and Saturation can go beyond if picking

Control	Type	What it does	Notes
			colors from a Channel in a colorspace with a wider gamut than the color picking colorspace.
R G B	input box, slider	Enter a value in the input field or click and drag from left to right to select Red, Green, and Blue.	
Alpha	input box, slider	Enter a value in the input field or click and drag from left to right to select the level of transparency.	
	button	Using the Eye Dropper tool, pick a color from within Mari, as well as from your desktop or other windows	
Canvas	dropdown	This controls how pixels in the Canvas are picked, pre- or post-shaded, lit, and color managed. **Pick raw pixels:** Resolves the color of the picked pixel within the Canvas, to the raw color value in the displayed Channel. **Pick screen pixels:** Picks the color of the picked pixel from the desktop's 8-bit screen buffer, which includes shading, lighting, and the color managed view transform.	All pixels picked outside of Mari's Canvas shall pick from the desktop's 8bit screen buffer. This includes previews of floating point images.

Control	Type	What it does	Notes
			Tip: To pick colors from 16/32-bit depth images, open the **Image** tab in the Colors Palette and click on the image.

Advanced Blending Mode

The **Advanced Blending** mode pop-up from within the **Layers** palette is described below with functions for how to access it. A list of the controls in the pop-up can be found in the table below.

What It Does

The **Advanced Blending** mode pop-up lets you adjust the blended colors before the blending mode is applied to the selected layers and layered shaders. These modes can be applied to each layer independently, regardless of whether it already has other adjustments applied to it.

How to Access It

- *Button* | Layers Palette > ■
- *Button* | Shaders Palette > **Shader Layers** tab> ■
- *Button* | Mask Stack > ■
- *Button* | Adjustment Stack > ■

Advanced Blending Mode Fields

Control	Type	What it does	Notes
Mode	dropdown	Select the blend mode that you want to modify with the advanced settings.	
Amount	text field, slider	Adjusts the blend mode amount.	
Amount Enabled	dropdown	**Enable** or **Disable** the ability to change the blend mode amount.	If **Amount** has already been modified and **Amount Enabled** is set to **Disable**, then the amount is not reset to the default. It is locked to the amount entered when **Disable** was set.
Blending	dropdown	Choose whether to modify the blending mode with **Basic** or **Advanced** settings. If you select **Basic**, the **Component** field and the curve editor fields cannot be modified.	
Component	dropdown	Select either the **Grayscale**, **Red**, **Green**, **Blue**, or **Luminance** components to adjust when modifying the curve editors and applying the blend mode.	
This Layer	curve editor	Maps between input and output values for the selected layer.	Drag to move points on the graph. Add points by clicking on the curve. Edit a point by right-clicking it.
Below layer	curve editor	Maps between input and output values for the layer stack below the selected layer.	Drag to move points on the graph. Add points by clicking on the curve. Edit a point by right-clicking it.
Swizzle Red	dropdown	Select the component - **R**, **G**, **B**, **A**, **One**, or **Zero** - that is output from the Red color	By default, the Red component is set to output red color values.

Control	Type	What it does	Notes
		component. This allows you to re-arrange the RGBA values.	
Swizzle Green	dropdown	Select the component - **R**, **G**, **B**, **A**, **One**, or **Zero** - that is output from the Green color component. This allows you to re-arrange the RGBA values.	By default, the Green component is set to output green color values.
Swizzle Blue	dropdown	Select the component - **R**, **G**, **B**, **A**, **One**, or **Zero** - that is output from the Blue color component. This allows you to re-arrange the RGBA values.	By default, the Blue component is set to output blue color values.
Swizzle Alpha	dropdown	Select the component - **R**, **G**, **B**, **A**, **One**, or **Zero** - that is output from the Alpha component. This allows you to re-arrange the RGBA values.	By default, the Alpha component is set to output alpha values.

Add Channel Dialog

The **Add Channel** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

This allows you to add a single channel to your project. The dropdown at the top of the dialog box lets you select from recently created channels. If you select a channel from the dropdown, Mari updates the rest of the fields with the details you used last time. You can change any of the values if required.

How to Access It

- *Menubar* | **Channel** > **Add Channel**
- *Right-click* | in Channels Palette > **Add Channel**
- *Button* | Channels Palette >

• *Button* | Shaders Palette >

Add Channel Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
<Previous channel names>	dropdown	This holds the names of recently created channels. Select a channel from the list, and the rest of the dialog box updates with the details from the last time the channel was created.		
Name	information, dropdown	Name of the channel to create.		
Size	dropdown	Patch size for the new channel. Either **256x256**, **512x512**, **1024x1024**, **2048x2048**, **4096x4096**, **8192x8192**, **16384x16384**, or **32768x32768**.		
Depth	dropdown	Color depth for the new channel. Either **8bit (Byte)**, **16bit (Half)**, or **32-bit (Float)**.		
File Space	dropdown	The type of file space used for vector painting: **Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**.		
Color Data | **Colorspace**	dropdown	The colorspace the channel data is converted to. **Automatic** is determined by the file name, size, and type of data in the channel. OCIO colorspace roles are also displayed in this list.		
Color	swatch	You can select the leftmost Color square to spawn color picker, or use any of the preset colors (white, black, mid grey, normal, transparent alpha). Any saved Preset Colors	Select Color Dialog	

Control	Type	What it does	Opens this dialog box	Notes
		can also be accessed here via the button.		

Add Paint/Bake Point Dialog

The **Add Paint/Bake Point** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

This allows you to add an image set for your Paint node.

How to Access It

- *Right-click* | Node Graph > **Create Paint node**
- *keyboard* | **P**

OR

- *Right-click* |**Nodes** > **Misc** > **Bake Point**.

Add Paint/Bake Point Fields

Control	Type	What it does	Opens this dialog box	Notes
Size	*dropdown*	Patch size for the new channel. Either		

Control	Type	What it does	Opens this dialog box	Notes
		256x256, **512x512**, **1024x1024**, **2048x2048**, **4096x4096**, **8192x8192**, **16384x16384**, or **32768x32768**.		
Depth	*dropdown*	Color depth for the new channel. Either **8bit (Byte)**, **16bit (Half)**, or **32-bit (Float)**.		
File Space	*dropdown*	The type of file space used for vector painting: **Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**.		
Color	*swatch*	You can select the leftmost Color square to spawn color picker, or use any of the preset colors (white, black, mid grey, normal, transparent alpha). Any saved Preset Colors can also be accessed here via the button.	Select Color Dialog	
Colorspace				
Color Data | **Colorspace**	*dropdown*	The colorspace the channel data is converted to. **Automatic** is determined by the file name, size, and type of data in the channel. OCIO colorspace roles are also displayed in this list.		
Color Data | **Raw Data**	*checkbox*	When checked, the image-set/channel is treated as raw numbers, so no color conversion is applied upon processing the data.		
Color Data | **Scalar Data**	*checkbox*	When checked, the data in the image-set/channel is considered as non-color data such as masks, heights, normals, and depths, so Mari stops applying the viewport colorspace LUT when such channel/image-		See Color Data and Scalar Data for more information.

Control	Type	What it does	Opens this dialog box	Notes
		set is displayed in isolation. For example, when you are viewing a scalar channel through "Current Channel" shader, Mari applies no viewport colorspace LUT.		
Mask Data | **Colorspace**	*dropdown*	The colorspace the channel mask data is converted to. **Automatic** is determined by the file name, size, and type of data in the channel. OCIO colorspace roles are also displayed in this list.		
Mask Data | **Raw Data**	*checkbox*	When checked, the image-set/channel is treated as raw numbers, so no color conversion is applied upon processing the data.		
Mask Data | **Scalar Data**	*checkbox*	When checked, the data in the image-set/channel is considered as non-color data such as masks, heights, normals, and depths, so Mari stops applying the viewport colorspace LUT when such channel/image-set is displayed in isolation. For example, when you are viewing a scalar channel through "Current Channel" shader, Mari applies no viewport colorspace LUT.		

Channel Presets Dialog

The **Channel Presets** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to set up preferences for quickly creating channels through the **Channels** palette or menu. These presets include options such as the color depth size, patch size, file space, and colorspace. This gives you the opportunity to quickly set up a channel template for channel types you know are going to be created numerous times. You can also choose from a few Mari channel presets.

How to Access It

- *Menubar* | **Channels** > **Channel Presets**
- *Right-click* | in Channels Palette > **Channel Presets**

Channel Presets Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Root Path	text field, dropdown, button	Specify the file path for the creation of new channel templates.	Choose Base Path dialog	The browse and scan buttons also allow you to search for the file location.
Scan	button	Scans the **Root Path** for any existing texture files.		After scanning the **Root Path**, you should always set the **Size** of the texture files manually. The default values Mari sets may not be correct as channels can contain many different patch sizes.
Import	dropdown	Select which patches to import if a channel is being imported for the		

Control	Type	What it does	Opens this dialog box	Notes
		creation of the template. Options are: **All Patches**.		
Resize	dropdown	Select which files should be resized when patches are being imported for channel template creation. Options are: **Patches to fit images** or **Images to fit patches**.		
Category	dropdown	Selects a Mari channel preset template from the list of available shaders.		**3Delight Principled - Metallic - VFX**, **3Delight Principled - Non-Metallic - VFX**, **General**, **Principled BRDF - Metallic - Games**, **Principled BRDF - Metallic - VFX, Principled BRDF - Non Metallic - Games, Principled BRDF - Non Metallic - VFX**, **Unreal - Metallic - Games**, **Unreal - Metallic - VFX**, **Unreal - Non-Metallic - Games**, **Unreal - Non-Metallic - VFX**, **VRayMtl - VFX**
Template	dropdown	Specifies a channel template to control the format of the filenames when Mari looks for existing textures on disk.		For more information about channel templates, see Defining Custom Channel Templates, Reviewing the Custom Channel

Control	Type	What it does	Opens this dialog box	Notes
				Template File Format, and Example File: DefaultTemplates.mtt.
Prefix	text field	Specifies the filename prefix for existing texture files, if applicable.		
Create	checkbox	When enabled, creates a shader and connects channels to the shader on project creation.		Most presets contain enabled channels by default.
Import	checkbox	Imports existing textures into the channel after creating it. Import settings were selected from the earlier option in the dialog.		
Name	information	Specifies the name of the channel to create. Mari sets a default value, but you can change this.		For example, color, specRough, dispFine, dispBroad, colorDetail, colorDetailMask.
Size	dropdown	Specifies the patch size for the new channel, from the options: **256x256**, **512x512**, **1024x1024**, **2048x2048**, **4096x4096**, **8192x8192**, **16384x16384**, **32768x32768**. Mari sets a default value, but you can change this.		Channels can be resized later from the **Channels** menu or palette if necessary.
Colorspace	dropdown	The colorspace in which the colors within your project are interpreted. Either select from the shipped configuration files or allow Mari to		

Control	Type	What it does	Opens this dialog box	Notes
		set an **Automatic** colorspace.		
		Tip: You can limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function or the MARI_COLORSPACE_OCIO_UI_ALLOWLIST environment variable. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus or Environment Variables That Mari Recognizes for more details.		
Scalar	checkbox	When **Colorspace** is set to **Automatic**, checking **Scalar** converts the resolved colorspace to scalar data. Scalar data in Mari is used to define the properties of a texture such as bumpiness, roughness, or shininess. See Color Data and Scalar Data for more information. **Note:** Automatically resolved 32-bit channels cannot be converted to scalar data. Only 8- and 16-bit channels are supported.		**Scalar** is disabled by default.
File Space	dropdown	Specifies the file space used when generating vectors. Mari sets a default but you can change this.		**Normal**, **Vector**, **Vector (flipped Y)**, or **Normal (flipped Y)**
Fill	swatch	Specifies the color to fill the new channel. Click the swatch to change the color.	Select Color Dialog	

Control	Type	What it does	Opens this dialog box	Notes
		Mari sets a default value, but you can change this.		
Depth	dropdown	Select a color depth size for the created channel template. Options are: **8bit (Byte)**, **16bit (Half)**, **32bit (Float)**.		Mari sets a default value, but you can change this. The color depth size for channels can also be resized later from the **Channels** menu or palette if necessary.
Files	text field	Specifies the filename template used to pick existing textures to import into the channel. Mari sets a default value, but you can change this, or you can use the button to open a dialog allowing you to manually pick the files.		This option is ignored if the Import checkbox is disabled.
	button	Opens a dialog box allowing you to pick the textures for Mari to import into the new channel. This corresponds to the File text field above.	Import Dialog	
	icon	Indicates that Mari has not found any textures to import.		If there are textures to import, Mari displays the number of patches available.

Masks Formats Dialog

The **Masks Formats** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

This allows you to set up a designated material channel to use when you are creating a layered shader.

How to Access It

- *Menubar* | **Shading** > **Add New Shader** > **Layered**
- *Right-click* | in Shaders Palette > **Add New Shader** > **Layered**
- *Button* | Shaders Palette > > **Layered**

Mask Formats Dialog fields

Control	Type	What it does	Notes
Size	dropdown	Set the channel size when creating a mask to be: • **256x256** • **512x512** • **1024x1024** • **2048x2048** • **4096x4096** • **8192x8192** • **16384x16384** • **32768x32768**	
Depth	dropdown	Sets the channel depth when creating a mask to be: • **8bit (Byte)** • **32bit (Float)** • **16bit (Half)**	

Convert Type Dialog

The **Convert Type** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Converts the color depth of the selected channel into a different bit depth. The color depth determines the number of pixels per RGB channel. Existing channels set to another color depth can be converted by simply selecting them and opening the **Convert Type** dialog box.

How to Access It

- *Menubar* | **Channels** > **Convert Channel**
- *Right-click* | in Channels Palette > **Convert Channel**
- *Button* | in Channels Palette >

Convert Type Dialog Fields

Control	Type	What it does	Notes
Type	dropdown	The color depth options for converting the selected channel. Depth sizes are: **8bit (Byte)**, **16bit (Half)**, and **32bit (Float)**.	You should only see two options in the dropdown menu, as it does not show the current depth size of your channel.

Copy Multiple Textures Dialog

The **Copy Multiple Textures** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you copy the textures from one layer to another. If you have multiple objects in your project, you can copy the textures between objects.

How to Access It

Menubar | **Patches** > **Copy Textures**

Copy Multiple Textures Dialog Fields

Control	Type	What it does	Notes
Source Object	dropdown	Select the object holding the source layer.	
Source Layer	dropdown	The layer to copy the patches from.	
Select By	dropdown	Select how to identify the patches on the source layer.	
Source Range	text field	The patch numbers of the patches to copy textures from. Enter the patch numbers. You can separate patch numbers with commas, or enter ranges. For example: 1001,1003,1005-1008	
Target Object	dropdown	The object to copy the patches onto.	
Target Layer	dropdown	The layer to copy the patches onto. If required, this can be the same as the source layer.	
Offset	text field	A numeric offset for the patch numbers. For example, if this is set to 100, the source patch 1001 is copied to 1101. Use this to copy textures between sides of symmetrical models.	The default for the Offset is zero.

Channel Transfer Dialog

The **Channel Transfer** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you transfer channels from one object or version to another.

How to Access It

- *Menubar* | **Channels** > **Transfer**
- *Right-click* | in Channels Palette > **Transfer**

Channel Transfer Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Source | **Object**	dropdown	Select the object you want to transfer channels from.		
Source | **Version**	dropdown	Select the version of the object you want to transfer channels from.		
Source | **Patches**	dropdown, text field, incrementation field	Select **All**, **Selected**, or **Range** to specify which patches are transferred from. You can only set **Patches** to **Selected** if you have already selected patches on the model prior to opening the **Channel Transfer** dialog.		
Destination | **Object**	dropdown	Select the object you want to transfer channels to.		

Control	Type	What it does	Opens this dialog box	Notes
Destination | **Version**	dropdown	Select the version of the object you want to transfer channels to.		
Destination | **Patches**	dropdown, text field, incrementer	Select **All**, **Selected**, or **Range** to specify which patches are transferred to. You can only set **Patches** to **Selected** if you have already selected patches on the model prior to opening the **Channel Transfer** dialog.		
	button	Moves all the channels that appear in the source channel list to the destination channel list.		
	button	Moves only the selected channel in the source channel list to the destination channel list.		Multiple selections in the source list can be moved to the destination list.
	button	Moves only the selected channel in the destination channel list back to the source channel list.		Multiple selections in the destination list can be moved to the source list.
	button	Moves all the channels that appear in the destination channel list back to the source channel list.		
Transfer Method	dropdown	Specifies the method for transferring the texture onto the destination UV map.		

Control	Type	What it does	Opens this dialog box	Notes
		Choose **UV Remap From Bakery** to use the Bakery to generate the UV coordinate mappings from the source object to the destination object. Or select the **UV Remap From File**option if you want to use an import-like template to remap pre-generated textures based on UV coordinates. When **UV Remap From File** is selected, import the template in the **UV Remap** column. For example, if you have generated the UV map beforehand, you can load it here.		
UV Remap *(when UV Remap from File is selected)*.		Uses an import-like template to use pre-generated textures of UV coordinate remap. For example, if you have generated the UV map beforehand, you can load it here.		
Size	dropdown	Specifies the texture resolution of the destination channel. You can set the **Size** to be the **Same As Source** or to any other supported channel sizes, see Channels Palette.		The texture resolution can only be set for UV maps.
Bleed Edges	dropdown	Choose **Yes** or **No** to specify whether or not to bleed patch edges when applying the transfer.		
Fill Color	dropdown	Sets the fill color for new patches to either **Transparent**, **Black**, **White**, **Mid-gray**, **Magenta**, **Foreground**, or **Background**.		

Control	Type	What it does	Opens this dialog box	Notes
R	button	Resets the dropdown fields in the **Advanced** > **Sampling** tab to the default options.		
Ok	button	Accepts the channel transfer parameters and starts the transfer.		
Cancel	button	Cancels the channel transfer and closes the **Channel Transfer** dialog.		

Layer Transfer Dialog

The **Layer Transfer** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you transfer layers from one object or version to another.

How to Access It

- Menubar | **Layers** > **Transfer**
- Right-click | layer in Layers Palette > Transfer

Layer Transfer Dialog Fields

Control	Type	What it does	Notes
Source | **Object**	dropdown	Select the object you want to transfer layers from.	

Control	Type	What it does	Notes
Source | **Version**	dropdown	Select the version of the object you want to transfer layers from.	
Source | **Patches**	dropdown, text field, increment field	Select **All**, **Selected**, or **Range** to specify which patches are transferred from. You can only set **Patches** to **Selected** if you have already selected patches on the model prior to opening the **Layer Transfer** dialog.	
Source | **Channel**	dropdown	Select the channel you want to transfer layers from.	
Destination | **Object**	dropdown	Select the object you want to transfer layers to.	
Destination | **Version**	dropdown	Select the version of the object you want to transfer layers to.	
Destination | **Patches**	dropdown, text field, increment field	Select **All**, **Selected**, or **Range** to specify which patches are transferred to. You can only set **Patches** to **Selected** if you have already selected patches on the model prior to opening the **Layer Transfer** dialog.	
Destination | **Channel**	dropdown	Select the channel you want to transfer layers to.	
	button	Moves all the layers that appear in the source layer list to the destination layer list.	
	button	Moves only the selected layer in the source layer list to the destination layer list.	Multiple selections in the source list can be moved to the destination list.
	button	Moves only the selected layer in the destination layer list back to the source layer list.	Multiple selections in the destination list

Control	Type	What it does	Notes
			can be moved to the source list.
	button	Moves all the layers that appear in the destination layer list back to the source layer list.	
Transfer Method	dropdown	Specifies the method for transferring the texture onto the destination UV map. Choose **UV Remap From Bakery** to use the Bakery to generate the UV coordinate mappings from the source object to the destination object. Or select the **UV Remap From File**option if you want to use an import-like template to remap pre-generated textures based on UV coordinates. When **UV Remap From File** is selected, import the template in the **UV Remap** column. For example, if you have generated the UV map beforehand, you can load it here.	
UV Remap *(when UV Remap from File is selected)*.	dropdown, file search window	Uses an import-like template to use pre-generated textures of UV coordinate remap. For example, if you have generated the UV map beforehand, you can load it here.	
Bleed Edges	dropdown	Choose **Yes** or **No** to specify whether or not to bleed patch edges when applying the texture transfer.	
Fill Color	dropdown	Sets the fill color for new patches to either **Transparent**, **Black**, **White**, **Mid-gray**, **Magenta**, **Foreground**, or **Background**.	
	button	Resets the dropdown fields in the **Advanced** > **Sampling** tab to the default options.	
Ok	button	Accepts the layer transfer parameters and starts the transfer.	

Control	Type	What it does	Notes
Cancel	button	Cancels the layer transfer and closes the **Layer Transfer** dialog.	

Mask from Selection Dialog

The **Mask from Selection** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What it does

The **Mask from Selection** dialog box enables you to create a mask from an active selection or Selection Group. This is useful for obtaining fine-tuned control to paint on and applying specific effects to certain sections of your model.

How to access it

- *Menu bar* | **Selection > Mask > Create Mask from Selection**
- *Right-click* | **Canvas** > **Selection** > **Mask** > **Create Mask from Selection**
- *Right-click* | **Node Graph** > **Selection** > **Mask** > **Create Mask from Selection**

Mask from Selection Dialog fields

Control	Type	What it does	Notes
Size	*dropdown*	Defines the size of the mask in pixels	
Depth	*dropdown*	Defines the bitdepth of the mask	
Selection	*dropdown*	Select whether to create the mask from the current selection, or from an existing Selection Group.	
Paint node	*text box*	Defines the name of the paint node to be created alongside the baked mask. If left empty, no paint node will be created.	

Control	Type	What it does	Notes
Geo-Channel	*dropdown, text box*	The name of the Geo-Channel to bake into. Will overwrite any existing Geo-Channel image sets. If left empty, Geo-Channels will remain untouched.	

Mask from Selection Group Dialog

The **Mask from Selection Group** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What it does

The **Mask from Selection** dialog box enables you to create a mask from a Selection Group. This is useful for obtaining fine-tuned control to paint on and applying specific effects to certain sections of your model.

How to access it

• *Right-click on selection group* | Selection Groups Palette > **Create Mask from Selection Group**

Mask from Selection Dialog fields

Control	Type	What it does	Notes
Size	*dropdown*	Defines the size of the mask in pixels	
Depth	*dropdown*	Defines the bitdepth of the mask	
Paint node	*text box*	Defines the name of the paint node to be created alongside the baked mask. If left empty, no paint node will be created.	
Geo-Channel	*dropdown, text box*	The name of the Geo-Channel to bake into. Will overwrite any existing Geo-Channel image sets. If left empty, Geo-Channels will remain untouched.	

ID Mask from Selection Group Dialog

The **ID Mask from Selection Group** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What it does

The **ID Mask from Selection Group** dialog box enables you to create a color ID masks from selection groups, which you can then use for painting and mask workflows.

How to access it

- *Menu bar* | **Selection** > **Mask** > **Create ID Masks from Selection Groups**
- *Right-click* | **Canvas** > **Mask** > **Create ID Masks from Selection Groups**
- *Right-click* | **Node Graph** > **Selection** > **Mask** > **Create ID Masks from Selection Groups**

ID Mask from Selection Group Dialog fields

Control	Type	What it does	Notes
Size	*dropdown*	Defines the size of the mask in pixels	
Depth	*dropdown*	Defines the bitdepth of the mask	
Create paint nodes	*checkbox*	Defines whether a paint node should be created for the baked mask. The name of the paint node will match the Single Map name field if **Create as single map** is checked. If **Create as single map** is unchecked, the name of the paint node(s) will match the selection groups.	

Control	Type	What it does	Notes
		If unchecked, no paint node(s) will be created.	
Create Geo-Channels	*checkbox*	Defines whether a Geo-Channel node should be created for the baked mask. The Single Map name will be used for the Geo-Channel name when **Create as Single Map** is checked. When **Create as Single Map** is unchecked, the name of the Geo-Channel(s) will match the selection groups. In both cases, if a Geo-Channel already exists with the same name, it will be overwritten.	
Create as single map	*checkbox*	If checked, masks are grouped into a single map. If unchecked, separate maps are made for each selection group.	
Single Map name	*text box*	Use this option to name the single map paint node and Geo-Channel, if applicable.	
Filter	*search bar*	Use this search bar to find specific selection groups by their name .	
Table			
Fill	*checkbox*	Defines which selection groups will have masks baked out. Check the box to bake out the Color mask.	
Color	*Color picker*	The randomly-assigned color for the selection group mask. Click to open up the Select Color Dialog if you want to edit the color. If you have selected multiple rows in the table (through **Ctrl** or **Shift** click), then editing one value will edit the other selected rows too.	

Control	Type	What it does	Notes
Selection Group	*Label*	The name of the selection group.	

Node Transfer Dialog

The **Node Transfer** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Lets you transfer nodes from one object or version to another.

How to Access It

• *Right-click* | on selected node in Node Graph > **Selection** > **Transfer**

Channel Transfer Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Source | **Object**	dropdown	Select the object you want to transfer nodes from.		
Source | **Version**	dropdown	Select the version of the object you want to transfer nodes from.		
Source | **Patches**	dropdown, text field, incrementer	Select All, Selected, or Range to specify which patches are transferred from. You can only set Patches to Selected if you have already selected patches on the model prior to opening the Node Transfer		

Control	Type	What it does	Opens this dialog box	Notes
		dialog.		
Destination | **Object**	dropdown	Select the object you want to transfer nodes to.		
Destination | **Version**	dropdown	Select the version of the object you want to transfer nodes to.		
Destination | **Patches**	dropdown, text field, incrementer	Select **All**, **Selected**, or **Range** to specify which patches are transferred to. You can only set **Patches** to **Selected** if you have already selected patches on the model prior to opening the **Node Transfer** dialog.		
Transfer Method	dropdown	Specifies the method for transferring the texture onto the destination UV map. Choose **UV Remap From Bakery** to use the Bakery to generate the UV coordinate mappings from the source object to the destination object. Or select the **UV Remap From File**option if you want to use an import-like template to remap pre-generated textures based on UV coordinates. When **UV Remap From File** is selected, import the template in the **UV Remap** column. For example, if you have generated the UV map beforehand, you can load it here.		
UV Remap (when UV Remap from File is	Dropdown, file search window	Uses an import-like template to use pre-generated textures of UV coordinate remap. For example, if you have generated the UV map beforehand, you		

Control	Type	What it does	Opens this dialog box	Notes
selected).		can load it here.		
Size	dropdown	Specifies the texture resolution of the destination node. You can set the **Size** to be the **Same As Source** or to any other supported channel sizes.		The texture resolution can only be set for UV maps.
Bleed Edges	dropdown	Choose **Yes** or **No** to specify whether or not to bleed patch edges when applying the transfer.		
Fill Color	dropdown	Sets the fill color for new patches to either **Transparent**, **Black**, **White**, **Mid-gray**, **Magenta**, **Foreground**, or **Background**.		
R	button	Resets the dropdown fields in the **Advanced** > **Sampling** tab to the default options.		
Ok	button	Accepts the node transfer parameters and starts the transfer.		
Cancel	button	Cancels the node transfer and closes the **Node Transfer** dialog.		

Open an Image Dialog

The **Open Image** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to choose an image from somewhere on your machine or network for use in environment lighting and the **Image Manager**. Image files can be opened in Mari using this dialog, so long as they are of a valid file format, shown in the **File of type** field in the table below.

How to Access It

- *Right-click* | Image Manager Palette > **Open**
- Image Manager Palette >
- *Right-click* | Lights Palette > **Environment** > **Image** >

Open Image Dialog Fields

Control	Type	What it does	Notes
Look in	dropdown	The directory containing the files to import. Select a recent directory from the dropdown, or use the navigation controls described below.	Make sure you check the path and name carefully. Mari defaults to the last place you imported from. Don't risk overwriting your textures!
	button	Goes back to the previous directory.	
	button	Goes forward to the next directory that you have browsed.	Only available if you have already browsed back using the back button.
	button	Goes up one level in the directory hierarchy.	
	button	Creates a new folder in the current directory.	
	button	Displays the contents of the directory as icons and names only.	

Control	Type	What it does	Notes
	button	Displays the contents of the directory as a details view, with a sort option at the top of the listing.	
Name	information	Lists the name of the files or folder in the selected directory.	The files that are shown are only those that are of a valid file format.
Size	information	Lists the file or folder size.	
Type	information	Displays what kind of file the item is, for example, a folder or a **.jpg** file.	
Date Modified	information	Lists the date that the file was last opened or modified.	
File name	text field	Type a file name into the field to open the image with a specific name in the **Image Manager**.	
File of type	dropdown	Lists the file formats that are able to be opened in Mari or brought into the **Image Manager**.	
Advanced Options			
Colorspace	dropdown	The colorspace the image data is converted to. **Automatic** is determined by the file name, size, and type of data in the channel. OCIO colorspace roles are also displayed in this list.	
Alpha	dropdown	• **Premultiply** - If there is an alpha channel, divides color data by the alpha before converting from the colorspace, and then multiplies by the alpha afterwards. This corrects the color of the partially-transparent pixels produced by most renderers. • **Leave** - Does not premultiply.	

Control	Type	What it does	Notes
		• **Remove** - Imports only the RGB data, discarding the alpha channel altogether.	

Save As Dialog

The **Save As** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to save images in the **Image Manager** palette in your Mari project to somewhere on your machine or network.

How to Access It

• *Right-click* | image in the Image Manager Palette > **Open**

• Image Manager Palette >

Save As Dialog Fields

Control	Type	What it does	Notes
Look in	dropdown	The directory where the image was imported from. Select a recent directory from the dropdown, or use the navigation controls described below.	
	button	Goes back to the previous directory.	
	button	Goes forward to the next directory that you have browsed.	Only available if you have already browsed back using

Control	Type	What it does	Notes
			the back button.
	button	Goes up one level in the directory hierarchy.	
	button	Creates a new folder in the current directory.	
	button	Displays the contents of the directory as icons and names only.	
	button	Displays the contents of the directory as a details view, with a sort option at the top of the listing.	
Name	information	Lists the name of the files or folder in the selected directory.	The files that are shown are only those that are of a valid file format.
Size	information	Lists the file or folder size.	
Type	information	Displays what kind of file the item is, for example, a folder or a **.jpg** file.	
Date Modified	information	Lists the date that the file was last opened or modified.	
File name	text field	Type a file name into the field to save the image with a specific file name outside of Mari.	
File of type	dropdown	Lists the file formats that are able to be saved out of Mari.	
Advanced options			
Colorspace	dropdown	The colorspace the image data is converted to. **Automatic** is determined by the file name, size, and type of data in the channel. OCIO colorspace roles are also displayed in this list.	

Control	Type	What it does	Notes
		In addition to the OCIO roles and colorspace options, there are also **Same as Native** and **Same as Working** options available in the dropdown menu. These allow you to maintain the same colorspace as the one that was native to the image, or as the one you used within Mari when you were working with the image.	

Select Patches By Range Dialog

The **Select Patches By Range** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Selects a range of patches according to their index or UDIM value number. Objects or scenes with many patches may benefit from being able to quickly select patches by their range value as opposed to selecting them manually from the **Patches** palette.

How to Access It

• *Menubar* | **Patches** > **Select Range**

Select Patches By Range Dialog Fields

Control	Type	What it does	Notes
Select By	dropdown	Gives you the choice of selecting a range of patches by either the UDIM value or the index	This is set to UDIM by default.

Control	Type	What it does	Notes
		value.	
Value	text field	Allows you to specify the UDIM or index values that you wish to select. You can select a range of consecutive values by listing them with a hyphen (for example, 1-5) and include non-consecutive values by separating them with a comma (for example, 1-5, 8).	

Assign Channel Dialog

The **Assign Channel** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to specify which Mari channels and which Modo effects should map to one another.

How to Access It

• *Button* | Modo Render Palette > **Add**

Assign Channel Dialog Fields

Control	Type	What it does	Notes
Mari Channel	dropdown	Specify the channel from your project that you want to map to a Modo effect.	
Modo Effect	dropdown	Specify the Modo effect that you want to map to a channel in Mari.	

Material Ingest Tool Dialog

The **Material Ingest Tool** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

The **Material Ingest Tool** dialog lets you ingest texture files and create material presets.

> **Note:** To find out how to create material presets using the **Material Ingest Tool** dialog, refer to Creating Materials with the Material Ingest Tool.

How to Access It

• *Menubar* | **Tools** > **Material Ingest Tool**

Material ingest Tool Dialog Fields

Control	Type	What it does	Notes
Presets	dropdown	Specifies the material preset to use. Once a preset is selected, Mari updates the current values in the **Material Ingest Tool** dialog. The **Presets** dropdown list contains the names of all preset files found in **C:\Program Files\Mari 8.0v1\Bundle\Media\Settings\MaterialIngestPresets**	

Control	Type	What it does	Notes
		Note: To load custom presets from another directory, use the **MARI_MATERIAL_INGEST_PRESETS_SEARCHPATHS** environment variable. See Environment Variables That Mari Recognizes.	
Save Presets	button	Saves all the current values in the **Material Ingest Tool** dialog as an **.mip** preset file.	
Load Presets	button	Loads a preset (**.mip** file). Once loaded, the preset name is added to the **Presets** dropdown menu and set as the current preset.	
Clear Presets		Clears the preset values (Shader Model patterns, case sensitive and the file template) and resets them to their default.	
Stream Configuration			
Material Template	dropdown	Allows you to pick an existing **.mma** file.	Locks the **Shader Model** to the one defined in the **.mma**. Once an **.mma** is picked, disables the **Procedural Type** field, and **Stream Name** missing streams are disabled if not all streams are present in the **.mma**. You can define paths to material templates in the Mari Preferences Dialog under the **Paths** tab.
	button	Allows you to pick an **.mma** file from disk.	Locks the **Shader Model** to

Control	Type	What it does	Notes
			the one defined in the **.mma**. Once an **.mma** is picked, disables the **Procedural Type** field, and **Stream Name** missing streams are disabled if not all streams are present in the **.mma**.
Shader Model	dropdown	Lists the shader models of the shaders loaded in your project. This updates the table containing the streams. This table is used to match the texture files' naming patterns to the shader input or stream they are targeted for. **Note:** You can access this list through Python using the following method: **mari.shader_models.list()**	
Case Sensitive	dropdown	Evaluates whether the stream patterns are case sensitive, in case texture files follow different capitalization conventions, for example: if set to **Yes** the pattern '**ao**' would not match a file with '**AO**' as the stream name. **Note: Case Sensitive** default to **Yes**.	
Stream Name	checkbox	Indicates each stream name of the current shader model set in the **Shader Model** dropdown. Toggles whether or not the stream should build a procedural image node when creating a material.	
Color	color swatch	Opens the **Select Color** dialog to set a color for the streams in case a matching texture file cannot be found. Double-click the color swatch to open the **Select Color** dialog.	

Control	Type	What it does	Notes
		Note: If the stream type is **Scalar**, indicated by the icon, and if no texture file can be found, a Constant node is added if the color is gray, otherwise a Vector node is added.	
Naming Patterns	text field	Specifies a list of naming patterns separated by a comma ',' in the order of importance, to identify an appropriate texture file for this stream. **Note:** Double-click the row to change the naming patterns.	
Ingestion Options			
Filename Template	text field	Specifies the naming pattern for texture files using the following templates: Required: • **$NAME** - specifies the name of your material. You can use any set of alphanumeric characters including underscores, spaces, dashes, and periods. • **$STREAM** - indicates where in the file name the pattern specified per stream should be matched. Optional: • **$EXT** - specifies any supported image extensions. • **$CHAR** - ignores any arbitrary characters. • **$NUM** - ignores any numbers. • **$ANY** - ignores any alphanumeric characters including underscores, spaces, dashes, and periods. • **$SEP** - specifies the system's path separator, forward slash **/** or backslash ****. • **NUMCHAR** can be used together to match against textures with their resolution in the filename, for example: **myMaterial_color_4k**	

Control	Type	What it does	Notes
		Note: The default value is **$NAME_$STREAM.$EXT** To specify multiple templates, use the pipe '**\|**' symbol to separate them, for example: **$NAME_$STREAM_NUMCHAR.$EXT|$NAME_$STREAM_$NUM$CHAR_$NAME.$EXT** To search for texture files in subfolders, you can use the backslash **** symbol, for example, to match **myMaterial\PNG\diffuse.png** use the template **$NAME\$CHAR\$STREAM.$EXT**	
Search Root Path	dropdown, button	Sets the root path that the Material Ingest Tool searches for collections of files matching the specified naming patterns. Either enter the path or browse to the path using the folder button. **Note:** The default value is **mari.resourses.DEFAULT_IMAGE**	
Ingest Method	dropdown	Specifies the ingestion method: • **Export Materials to Disk** - this exports materials in **.mma** files to the specified location set in **Material Export Path**. • **Build Materials in Project** - this adds materials as Tiled or Tri Planar Projection nodes in the Node Graph. • **Display Search Results Only** - opens the **Texture Set Search Results** window which displays all the found texture sets used to create Materials in the specified **Search Root Path**.	
Thumbnail Template	text field	Specifies the naming convention for a thumbnail file using a set number of templates: • **$NAME** - any set of alphanumeric characters that represent the name of your material. You can use any set of alphanumeric characters including underscores, spaces, dashes, and periods. • **$EXT** - this is replaced by any supported image type. • **$CHAR** - this can be used to ignore any arbitrary characters. • **$NUM** - this can be used to ignore any numbers.	

Control	Type	What it does	Notes
		• **$ANY** - this can be used to ignore any alphanumeric characters, including underscores, spaces, dashes, and periods. • **$SEP** - this is replaced by the system's path separator (****).	
Material Export Path	dropdown, button	Sets the path to where the Material Ingest Tool saves the generated materials. Either enter the path or browse to the path using the folder button. **Note:** The default value should be **mari.resourses.DEFAULT_EXPORT**	
Procedural Type	dropdown	Specifies which image based procedural node to generate materials with: • **Tiled** - uses one image to tile across the model. • **Tri Planar Projection** - projects the texture map on your model from three different directions (Top, Front, and Right).	
Add To Shelf	dropdown	Specifies in which shelf to add the generated materials. You can add them to the **Menu**, **Personal**, or **Project** shelves or you can create a new shelf. Select **Do Not Add to Shelf**, if you don't require to add them to a shelf.	
New Shelf	text field	Specifies the name of the new shelf to which generated materials are added. In the **Add to Shelf** dropdown, select **Add to New Shelf** to enable the **New Shelf** field.	
Create Materials	button	Creates the materials using the information you provided in the **Material Ingest Tool** dialog.	
Close	button	Closes the **Material Ingest Tool** dialog.	

Control	Type	What it does	Notes
		Note: The **Material Ingest Tool** dialog remembers the last settings used.	
Texture Set Search Results			
Show Relative Paths	checkbox	When unchecked, shows the full path to the found texture asset in the **Matched Filename** column. When checked, shows the relative path to the found texture asset.	
Show Unmatched Streams	checkbox	When unchecked, removes any stream names from the **Stream Name**column that have no found texture assets. When checked, keeps all stream names from the selected **Shader Model** even if no texture assets were found for that stream name.	
Material Names	list	Lists the found texture set materials using the **$NAME** naming pattern from the **Filename Template**.	
Stream Name	list	Lists the stream names in the specified **Shader Model**. If **Show Unmatched Streams** is toggled off, only stream names with found texture assets are displayed.	
Matched Filename	list	Lists the texture filenames for a given stream name found using the naming pattern specified in **Filename Template**. If no texture assets are found for a given stream name, **No images found matching patterns** is displayed next to the stream name.	
Close	button	Closes the **Texture Set Search Results** dialog.	

Create Shader Dialog

The **Create Shader** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

The **Create Shader** dialog box allows you to create shaders that are not listed in the **Shading** menu or **Shaders** palette. These shaders are primarily those that do not use Lambertian diffuse shading. However, options for
Lambertian diffuse shading are still included in the **Create Shader** dialog, so you can customize your shaders however you want.

How to Access It

- *Menubar* | **Shading** > **Add New Shader** > **Choose Diffuse and Specular**
- *Right-click* | Shaders Palette > **Choose Diffuse and Specular**
- Shaders Palette | > **Choose Diffuse and Specular**

Create Shader Dialog Fields

Control	Type	What it does	Notes
Diffuse	dropdown	Select either Lambertian or Minnaert for Lighting/Diffuse options. Selecting either option does not limit the options you can select from the **Specular** dropdown menu.	Selecting Lambertian and a Specular options gives you the same result as the named options for **Add New Shader** on the menubar.
Specular	dropdown	Select Phong, Cook-Torrance, Beckman, or Blinn for the Lighting/Specular options.	

Object Dialogs

Add Object Settings Dialog

The **Add Object Settings** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

Specifies the settings of the imported object that is being added to the Mari project. This relates primarily to whether selected objects are merged, and how, and how the object mapping is handled.

How to Access It

- *Menubar* | **Objects** > **Add Object** > select object and **Open**
- *Right-click* | in Objects Palette > **Add Object** > select object and **Open**
- *Right-click* | object in Objects Palette > **Add Child Object** > select object and **Open**
- *Button* | Objects Palette > > select object and **Open**

Add Object Settings Dialog Fields

Control	Type	What it does	Notes
Mapping Scheme	dropdown	If there is a UV map available, and you would like to use it**,** select **UV if available, Ptex otherwise**. If you prefer to use Ptex, select **Force Ptex**. This option loads the selected geometry and replaces the existing UV map with a Ptex map.	Only available when choosing **.obj**, **.abc**, or **.fbx** files in the file browser.
Selection Groups	dropdown	Selection groups have two options, to either: • **Create From Face Groups** - this is the default behavior and allows the use of selection groups in Mari.	Only available when choosing **.obj** files in the file browser.

Control	Type	What it does	Notes
		• **Don't Create** - if you choose not to create a selection group, Mari does not create any from the file.	
Multiple Geometries Per Object	dropdown	If your **.obj** consists of multiple geometries, you have the option to **Merge Geometries Into One**, which is the default behavior, or **Create Separate Geometries (Ptex Only)**. If you select **Merge Geometries Into One**, Mari combines the individual geometries into a single geometry, while selecting **Create Separate Geometries (Ptex Only)** creates individual geometries.	Only available when choosing **.obj** files in the file browser. Even if you select **Create Separate Geometries (Ptex Only)**, Mari only displays a single geometry on the canvas. However, when you export a channel you have the option to export individual **.ptx** files for each geometry.
Merge Type	dropdown	For Ptex files, there are two choices for the geometry mesh options: • **Separate Meshes** - creates only one object with one geometry. • **Single Mesh** - creates only one object, but has multiple geometries within it.	Only available when choosing **.ptx** files in the file browser. The option **Single Mesh** is essentially the inverse of the Ptex channel export process, and allows project reconstruction.
Merge Type	dropdown	Choose whether to merge selected items from the object hierarchy tree as: • **Single Mesh** - merges selected geometry into one single mesh in a GeoEntity. Note that this implies that the information about the original geometries cannot be accessed once the merge is done. • **Just Merge Nodes** - merges selected geometry into one GeoEntity while retaining the metadata and mesh information about the original individual geometries. Note, this	Only available when choosing **.abc** or **.fbx** files in the file browser.

Control	Type	What it does	Notes
		information can only be accessed by more technical users using the C API.	
Take	dropdown, information	If an object has been exported with animation takes, the takes are shown in the dropdown menu. Otherwise, information regarding the only take available is shown.	Only available when choosing **.fbx** files in the file browser. When adding an object with animation takes, the take can only be selected on import and cannot be changed at a later point.
Scene	information	In the object hierarchy, shows all the objects, cameras, and lights that are part of the Alembic file.	Only available when choosing **.abc** or **.fbx** files in the file browser.
Merge	checkbox	Select the checkbox to determine if a geometry is to be merged with another on project creation.	Only available when choosing **.abc** or **.fbx** files in the file browser. For more information on merging Alembic or FBX geometry, see *Create a New Project* under the *Alembic and FBX* chapter in the *Mari User Guide*.
Start Frame and **End Frame**	text field	Set the frame range from the source file to import.	Only available when choosing **.obj**, **.abc**, and **.fbx** file sequences in the file browser.
Single Frame	button	Sets the start and end frame fields to only import the first frame from the animation in the object file.	Only available when choosing **.obj**, **.abc**, and **.fbx** file sequences in the file browser.
All Frames	button	Sets the start and end frame fields to import the entire animation from the object file.	Only available when choosing **.obj**, **.abc**, and **.fbx** file sequences in the file browser.
Frame Offset	text field	Set by how many frames to offset the loading of an animated object. You can	Only available when choosing **.abc** files in the file browser.

Control	Type	What it does	Notes
		specify a frame range, or simply a number of frames.	
Ok	button	Accepts the object settings and imports the selected objects.	
Cancel	button	Cancels the object import and closes the **Add Object Settings** dialog.	

Add Version Settings Dialog

The **Add Version Settings** dialog is described below with functions for how to access it. A list of the controls on the dialog box can be found in the table below.

What It Does

Specifies the settings of the imported object version that is being added to the Mari project.

How to Access It

• *Right-click* | on object in Objects Palette > **Add Version** > select object and **Open**

Add Version Settings Dialog Fields

Control	Type	What it does	Notes
Mapping Scheme	dropdown	If there is a UV map available, and you would like to use it**,** select **UV if available, Ptex otherwise**. If you prefer to use Ptex, select **Force Ptex**. This option loads the selected geometry and replaces the existing UV map with a Ptex map.	

Control	Type	What it does	Notes
Merge Type	dropdown	Choose whether to merge selected items from the objecthierarchy tree as: **Single Mesh** - merges selected geometry into one single mesh in a GeoEntity. Note that this implies that the information about the original geometries cannot be accessed once the merge is done. **Just Merge Nodes** - merges selected geometry into one GeoEntity while retaining the metadata and mesh information about the original individual geometries. Note, this information can only be accessed by more technical users using the C API.	Select **Just Merge Nodes** to preserve selection groups between versions. See Adding and Removing Object Versions .
Merge Selection Groups	dropdown	You can choose to **Merge selection groups with the same name** or choose the **Do not merge selection groups** option.	Select **Merge selection groups with the same name** to preserve selection groups between versions. See Adding and Removing Object Versions .
Take	dropdown, information	If an object has been exported with animation takes, the takes are shown in the dropdown menu. Otherwise, information regarding the only take available is shown.	Only available when choosing **.fbx** files in the file browser. When adding an object with animation takes, the take can only be selected on import and cannot be changed at a later point.
Scene	information	In the object hierarchy, shows all the objects, cameras, and lights that are part of the Alembic file.	
Merge	checkbox	Select the checkbox to determine if a geometry is to be merged with another on	For more information on merging Alembic or FBX

Control	Type	What it does	Notes
		project creation.	geometry, see *Create a New Project* under the *Alembic and FBX* chapter in the *Mari User Guide*.
Start Frame and **End Frame**	text field	Set the frame range from the source file to import.	
Single Frame	button	Sets the start and end frame fields to only import the first frame from the animation in the object file.	
All Frames	button	Sets the start and end frame fields to import the entire animation from the object file.	
Frame Offset	text field	Set by how many frames to offset the loading of an animated object. You can specify a frame range, or simply a number of frames.	
Ok	button	Accepts the object settings and imports the selected object versions.	
Cancel	button	Cancels the object import and closes the **Add Version Settings** dialog.	

Subdivide Dialog

The **Subdivide** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Invoked by the object subdivision commands, the **Subdivide** dialog box lets you specify options for how an object's mesh is further divided.

How to Access It

- *Menubar* | **Objects** > **Subdivide**
- *Right-click* | Objects Palette > **Subdivide**
- Objects Palette| Subdivision| **Subdivide**

Subdivide Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Level	dropdown	Indicates how dense the mesh is, once divided, with **3** being the highest level of subdivision. The number of subdivided quads or triangles for each level depends on the Subdivision **Scheme** used.		
Snap to Limit Surface	checkbox	Snaps the last level of subdivision to the limit surface.		
Use Source	checkbox	Uses subdivision options (Scheme and rules) from the source geometry file.		
Force Subdivision	checkbox	When enabled, Mari forces subdivision of the geometry no matter how the subdivision flags are set.		
Scheme	dropdown	The subdivision method that you want to use when subdividing the object: • **Catmull Clark** - Mari uses the Catmull-Clark subdivision method when subdividing a mesh. • **Loop** - Mari uses the Loop subdivision method when subdividing a mesh. Loop		

Control	Type	What it does	Opens this dialog box	Notes
		subdivision supports meshes consisting of only triangles. • **Bilinear** - Mari uses the Bilinear subdivision method when subdividing a mesh.		
Boundary Interpolation	dropdown	Specifies how the boundary edges and vertices are interpolated. • **None** - no boundary edge interpolation should occur, instead boundary faces are tagged as holes so that the boundary edge-chain continues to support adjacent interior faces but is not considered to be part of the refined surface. • **Sharpen edges** - all the boundary edge-chains are sharp creases, boundary vertices are not affected. • **Sharpen edges and corners** - all the boundary edge-chains are sharp creases and boundary vertices with exactly one incident face are sharp corners.		
Face Varying Linear Interpolation	dropdown	Specifies the face varying linear interpolation rule: • **None** - smooths everywhere the mesh is smooth. • **Corners only** - sharpens (linearly interpolate) corners only. • **Corners plus1** - **Corners only** + sharpening of junctions of 3 or more regions.		

Control	Type	What it does	Opens this dialog box	Notes
		• **Corners plus2** - **Corners plus1** + sharpening of darts and concave corners. • **Boundaries** - linear interpolation along the boundary edges and corners • **All** - linear interpolation of everywhere (boundaries and interior).		
Creasing Method	dropdown	Interpolates the sharpness of incident edges: • **Uniform** - applies regular semi-sharp crease rules. • **Chaikin** - applies Chaikin semi-sharp crease rules. Chaikin's curve subdivision algorithm improves the appearance of multi-edge semi-sharp creases with varying weights.		
Triangle Subdivision	dropdown	Rule added to the Catmull-Clark scheme that can be applied to all triangular faces: • **Default** - default Catmull-Clark scheme weights. • **Smooths** - smooths triangle weights.		
OK	button	Click to subdivide the object and display a subdivision progress bar. The subdivision **Level** chosen is applied in the Viewer when the progress bar is finished.		
In Background	button	Click to subdivide the object using a		

Control	Type	What it does	Opens this dialog box	Notes
		background process, no progress bar is displayed. The subdivision **Level** must be applied manually in the **Objects** palette when the running man icon disappears.		
Cancel	button	Click to close the **Subdivide** dialog.		

Gaussian Blur Dialog

The **Gaussian Blur** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to apply an object-based blur effect across the whole object, not just on specific patches. This eliminates seaming because it can apply the effect across 3D boundaries.

How to Access It

- *Menubar* | **Objects** > **Generate** > **Gaussian Blur**
- *Right-click* | Objects Palette > **Generate** > **Gaussian Blur**

Gaussian Blur Dialog Fields

Control	Type	What it does	Notes
General			
Geometry Version	dropdown	Select the object version to which you want to apply the blur.	
Frame	text field, slider	Set the frame to which you want to apply the blur.	
Samples	text field, slider	Set the sample size for the blur effect.	
Sampling			
Center Positions	checkbox	Choose whether or not to center the sampling position for the blur effect. This is where the **Samples** size starts from.	
Bleed Edges	checkbox	Choose whether or not the blur effect bleeds over the edges of patches.	
Range Amount	text field, slider	The range (distance) to which the blur effect extends. The units in which this is measured are determined in **Range Units**.	
Range Units	dropdown	The unit of measurement for the **Range Amount** field: • **Infinite** - the range amount is set in the context of infinite XYZ space. • **World Units** - the range amount is set in worldspace. • **Object Size Percentage** - the range amount is determined as a percentage of the total size of the object.	

Height Dialog

The **Height** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to generate a height map, using a high polygon source model and a low polygon destination model to calculate the height.

How to Access It

- *Menubar* | **Objects** > **Generate** > **Height**
- *Right-click* | Objects Palette > **Generate** > **Height**

Displacement Dialog Fields

Control	Type	What it does	Notes
Source			
Object	dropdown	Select the high polygon object from which you want to calculate the height map.	
Version	dropdown	Select the high polygon object version from which you want to calculate the height map.	
Size	dropdown	Set the size of the selected channel on the low polygon object for which you are creating the displacement map.	
Frame	text field, slider	Set the frame from which you want to calculate the height map.	
Destination			
Version	dropdown	Select the low polygon object version to	

Control	Type	What it does	Notes
		which you want to apply the height map.	
Frame	text field, slider	Set the frame to which you want to apply the height map.	
Sampling			
Interpolation	dropdown	Interpolation is similar to Photoshop's sampling functionality. You can select the method Mari uses to average out pixels from either: • **Nearest Neighbor**- selects the nearest pixel for sampling. • **Inverse Distance Weighted** - weights the average on the closest pixels. • **Inverse Distance Weighted (Smoother)** - weights the average on the closest pixels with a priority to providing smoother sampling. • **Inverse Distance Weighted (Sharper)** - weights the average on the closest pixels with a priority to providing sharper sampling.	
Sample Count	text field, slider	Set the number of points the search finds and combines when sampling.	
Center Positions	checkbox	Choose whether or not to center the sampling position for the height map. This is where the **Samples** size starts from.	
Bleed Edges	checkbox	Choose whether or not the height map bleeds over the edges of patches.	
Range Amount	text field, slider	The range (distance) to which the height map extends. The units in which this is measured are determined in **Range Units**.	
Range Units	dropdown	The unit of measurement for the **Range**	

Control	Type	What it does	Notes
		Amount field: • **Infinite** - the range amount is set in the context of infinite XYZ space. • **World Units** - the range amount is set in worldspace. • **Object Size Percentage** - the range amount is determined as a percentage of the total size of the object.	

Displacement Dialog

The **Displacement** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Allows you to generate a displacement map, using a high polygon source model and a low polygon destination model to calculate the displacement.

How to Access It

• *Menubar* | **Objects** > **Generate** > **Displacement**

• *Right-click* | Objects Palette > **Generate** > **Displacement**

Displacement Dialog Fields

Control	Type	What it does	Notes
Source			
Object	dropdown	Select the high polygon object from which you want to calculate the displacement map.	

Control	Type	What it does	Notes
Version	dropdown	Select the high polygon object version from which you want to calculate the displacement map.	
Size	dropdown	Set the size of the selected channel on the low polygon object for which you are creating the displacement map.	
Frame	text field, slider	Set the frame from which you want to calculate the displacement map.	
Destination			
Version	dropdown	Select the low polygon object version to which you want to apply the displacement map.	
Frame	text field, slider	Set the frame to which you want to apply the displacement map.	
Sampling			
Interpolation	dropdown	Interpolation is similar to Photoshop's sampling functionality. You can select the method Mari uses to average out pixels from either: • **Nearest Neighbor**- selects the nearest pixel for sampling. • **Inverse Distance Weighted** - weights the average on the closest pixels. • **Inverse Distance Weighted (Smoother)** - weights the average on the closest pixels with a priority to providing smoother sampling. • **Inverse Distance Weighted (Sharper)** - weights the average on the closest pixels with a priority to providing sharper sampling.	

Control	Type	What it does	Notes
Sample Count	text field, slider	Set the number of points the search finds and combines when sampling.	
Center Positions	checkbox	Choose whether or not to center the sampling position for the displacement map. This is where the **Sample Count** starts from.	
Bleed Edges	checkbox	Choose whether or not the displacement map bleeds over the edge of patches.	
Range Amount	text field, slider	Sets the range for the search distance. The units in which this is measured are determined in **Range Units**.	
Range Units	dropdown	The unit of measurement for the **Range Amount** field: • **Infinite** - there is no limit; Mari attempts to find a pixel in the source image data. • **World Units** - the maximum search distance expressed directly in world space. • **Object Size Percentage** - the maximum search distance is expressed as a percentage of the destination object's size.	

Import and Export Dialogs

Export Manager Dialog

The **Export Manager** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

The **Export Manager** dialog lets you specify options for exporting, in batch, channels and Bake Point nodes as flattened from the current Mari project. The Export Manager retains the export settings throughout your project's life.

Note: In order to help you export successfully, cells containing overrides are highlighted in yellow, warning messages are highlighted in blue and cells containing errors are highlighted in red to point out potential risks of export failures.

Tip: Once you have selected one or more items, use the **Export Manager** dialog right-click context menu to easily check or uncheck and export or remove items from the export list. You can also right-click on the **Size**, **Colorspace**, **Depth**, **Set File Name**, **File Options**, and **Post Process Command** columns to open their corresponding dialog from the context menu.

Tip: You can export multiple variations of the same source channel. For example if you want to export a 16-bit channel to full sized EXR and a half sized proxy JPG.

How to Access It

- *Menubar* | **Channels** > **Export Manager**
- *Right-click* | in Channels Palette > **Export Manager**
- *Right-click* | on canvas > **Export Manager**

Export Manager Dialog Fields

Control	Type	What it does	Notes
Save	button	Saves your export settings to a file on disk.	This only saves export

Control	Type	What it does	Notes
Export Settings			settings for the currently selected object(s).
Load Export Settings	button	Uploads a previously exported settings file for you to export channels and Bake Point nodes.	This only loads export settings for the currently selected object(s). The channel and Bake Point node's names have to match for the settings file to be exploitable.
Export Root Path	text field, dropdown	Selects the file location to export to.	
	button	Navigates to the file location to export to.	
Export Items			
Object Name	text field, dropdown	Selects the object of your current project for which you want to export channels and Bake Point nodes.	
Add New Export Items	button	Adds one or more items (channels or Bake Point nodes) to the export list.	All existing or added channels (not Bake Point nodes) to your project are displayed in the export list by default.
Remove Export Item	button	Removes the selected items (channels or Bake Point nodes) from the list.	
Source	checkbox	Selects the channels and Bake Point nodes for which to export settings.	
Size	text field, dropdown	Specifies the resolution of the item from the following options: **Same as source**	You can also right-click in the **Size** cell and select **Set Size** from the dropdown menu. This

Control	Type	What it does	Notes
		256 x 256 **512 x 512** **1024 x 1024** **2048 x 2048** **4096 x 4096** **8192 x 8192** **16384 x 16384** **32768 x 32768**	opens the **Set Size** dialog where you can set the resolution.
Colorspace	text field, dropdown	Specifies the colorspace to which the output values are converted. When the Colorspace option is set to **Same as source**, the colorspace is determined by the channel's file name, size, and type of data in file. The available options depend on your config file. See Color Management.	You can also right-click in the **Colorspace** cell and select **Set Colorspace** from the dropdown menu. This opens the **Set Colorspace** dialog where you can set the colorspace.
Depth	text field, dropdown	Changes the color depth of the channel or Bake Point node to either **8bit (Byte)**, **16bit (Half)**, or **32bit (Float)**.	You can also right-click in the **Depth** cell and select **Set Depth** from the dropdown menu. This opens the **Set Depth** dialog where you can set the depth.
File Name	text field	Specifies the default file name value and format, as well as the sub-folder of the export item.	You can also right-click in the **File Name** cell and select **Set File Name** from the dropdown menu. This opens the **Set File Name** dialog where you can set the default

Control	Type	What it does	Notes
			file name.
File Options	text field	Double click the File Options field to open the **Set File Options** dialog where you can specify the file format options: **Compression** - Set the export OpenEXR compression type. The default is zip compression. This option is only available if your file is an EXR. **Type** - Set the type of EXR file to export.This option is only available if your file is an EXR. • **Tiled**: Default. EXR files will be saved in the tiled format without mipmaps. The tile size is 256x256. • **Renderman**: EXR files will be saved in the Renderman compliant format. The tile size is 32x32 and mipmaps will be generated. • **Scanline**: EXR files will be saved in the scanline format. Currently only available via PythonAPI with an opt-in flag. **Alpha Channel** - Select **Keep** or **Remove** alpha channels when exporting textures. This can help decrease the resulting file size when exporting images that don’t need alpha channels. **Full Edge bleed** - Enable or disable to specify whether or not to bleed patch edges when exporting channels or Bake Point	You can also right-click in the **File Options** cell and select **Set File Options** from the dropdown menu. This opens the **Set File Options** dialog where you can set the file options for your channel. Enabling **Small Textures** saves disk space if you're exporting a channel that hasn't been fully painted. Disable **Small Textures** if you want Mari to always export patches at their full size (even if they haven't been painted).

Control	Type	What it does	Notes
		nodes. **Small Textures** - With this option enabled, Mari exports any patches filled with a constant color as 8x8 pixel textures, no matter what resolution the patches are in Mari.	
Post Process Command	text field, dropdown	Through a command line, specifies what actions to take after exporting the channel or Bake Point node export settings file. You can use the following commands: **$ROOT** - the name set in the **Export Root Path** field. **$EXT** - the name of the extension set in the **File Options** field. **$UDIM** -applies the post process command to each UDIM **$EXPORTED** - a placeholder for the full export path, which allows you to run one command on each exported file **$CWD** - the location of the Current Working Directory. **$FILENAME** - the name specified in the **File Name** field. For example on Linux - To copy exported files to a pre-existing backup folder. This can be useful if you want to back up any exported files on a separate network location or multiple network locations, to be sure no data is lost.	You can also right-click in the **Post Process Command** cell and select **Set Post Process Command** from the dropdown menu. This opens the **Post Process Command** dialog where you can enter your command line. You can right-click to set the same post process command across all selected cells.

Control	Type	What it does	Notes
		rsync -avc $EXPORTED /home/<username>**/MyBackup** To compress exported images to a 'tar' archive called 'dailies'. **tar -rvf $ROOT/dailies.tar -C $ROOT $FILENAME** For example on Windows - To copy the exported images to a backup folder called 'MyBackup'. **Robocopy $ROOT** %HOMEPATH% **\Documents\Mari\MyBackup $FILENAME** To print the size on disk of each image exported, using a Python script. **python** %HOMEPATH% **\Documents\Mari\Exports\postScript.py $EXPORTED $FILENAME $ROOT $EXT**	
Status	information	Indicates if there are some errors in your export settings or anything that requires your attention.	
Export Overrides			
Size	text field, dropdown	Overrides the resolution of all the channels or Bake Point nodes, set in the channel/Bake Point node's **Size** field, with one of the following options:	

Control	Type	What it does	Notes
		Same as source **256 x 256** **512 x 512** **1024 x 1024** **2048 x 2048** **4096 x 4096** **8192 x 8192** **16384 x 16384** **32768 x 32768**	
Depth	text field, dropdown	Overrides the color depth of all the channels or Bake Point nodes, set in the channel/Bake Point node's **Depth** field, with either **8bit (Byte)**, **16bit (Half)**, or **32bit (Float)**.	
Patches	dropdown	Overrides whether Mari exports all or selected patches of the object only.	
Colorspace	text field, dropdown	Overrides the colorspace of all the channels or Bake Point nodes, set in the channel/Bake Point node's **Colorspace** field. The available options depend on your config file. See Color Management.	
Post Process	text field, dropdown	Overrides the Post Process Command line of all the channels or Bake Point nodes, set in the channel/Bake Point node's **Post Process Command** field.	
Export Current	button	Exports the channels or Bake Point nodes for the object currently selected in the **Export Manager** dialog.	

Control	Type	What it does	Notes
Export All	button	Exports the channels or Bake Point nodes for all objects, and closes the **Export Manager** dialog.	You can export the same channel or Bake Point node several times as long as the settings or the export path differs.
Close	button	Closes the **Export Manager** dialog.	

Export Dialog

The **Export** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Invoked by the export channel functions, export flattened channel functions, export layer functions, or export flattened layers functions, the **Export** dialog box lets you specify options for exporting channels or layers from the current Mari project, in either flattened or non-flattened states.

If you export with **Small Textures** set to **Enabled** then any patches that are entirely filled with one color (such as gray, before anything else has been painted on the patch,) are exported as small 8x8 images to save space. Mari
supports loading these 8x8 images when importing. These images are scaled up to fill the entire patch.

How to Access It

- *Menubar* | **Channels** > **Export** > any of the dropdown options
- *Menubar* | **Channels** > **Export Flattened** > any of the dropdown options
- *Menubar* | **Layers** > **Export** > any of the dropdown options
- *Menubar* | **Layers** > **Export Flattened** > any of the dropdown options
- *Right-click* | in Channels Palette > **Export** > any of the dropdown options

- *Right-click* | in Channels Palette > **Export Flattened** > any of the dropdown options
- *Right-click* | in Layers Palette > **Export** > any of the dropdown options
- *Right-click* | in Layers Palette > **Export Flattened** > any of the dropdown options

Export Dialog Fields

Control	Type	What it does	Notes
Named Files - The **Named files** tab does not display if your model contains more than 20 patches.			
U	information	Lists the U texture coordinate for the left edge of the patch.	
V	information	Lists the V texture coordinate for the bottom edge of the patch.	
UDIM	information	Lists the UDIM number for the patch.	
File	text field	Specify the file location to export to.	
	button	Browse to the file location you want to export the texture to.	
Sequence			
Layer	information	The name of the object, channel, and layer being exported.	
Path	text field, dropdown	The directory to export the files to. Type the directory, select a recent directory from the dropdown, or use the navigation controls below.	
Reset To Defaults	button	Resets the path to the default location.	
	button	Goes back to the previous directory.	
	button	Goes forward to the next directory that you have browsed.	Only available if you have already browsed back using the back button.

Control	Type	What it does	Notes
	button	Goes up one level in the directory hierarchy.	
	button	Creates a new folder in the current directory.	
	button	Displays the contents of the directory as icons and names only.	
	button	Displays the contents of the directory as a details view, with a sort option at the top of the listing.	
Filter	text field, dropdown	Lets you filter the image sequences displayed in the dialog box. As you enter the filter text, Mari updates the display to only show the image sequences that match your text.	
Full	Image	Indicates whether the selected image sequence has a full set of images for the current model. That is, for each patch on the model, there is a corresponding image in the image sequence. Green if a full set of images is present, otherwise the number found vs number required (for example, 4/5) displays.	
Images	information	Lists the image sequences found in the selected **Path**.	
Start	information	First UDIM number in the selected image sequence.	
End	information	Last UDIM number in the selected image sequence.	
Patch Count	information	Number of image patches in the image sequence.	

Control	Type	What it does	Notes
Frames	Information	How many frames in the image sequence.	
Modified	information	Last modified date for the images in the sequence.	
Template	text field, dropdown	Sets the **Template** for Mari to use when gathering the names of images in the selected sequence. Variables available are: • **$ENTITY** - the name of the object • **$CHANNEL** - the name of the channel • **$LAYER** - the name of the layer • **$NODE** - the name of the node • **$UDIM** - the UDIM patch value • **$FRAME** - the animation frame number • **$NUMBER** - number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28") • **$COUNT** - total number of files in the channel • **$COLORSPACE** - the chosen colorspace for the selected images. • **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").	Don't use numbers in your texture name - if you do, the Import and Export dialog boxes won't recognize the sequence.
File Example	information	Displays an example of the selected template path and filename.	
Formats	Information	Displays a list of supported export file formats.	
Use Template For	dropdown	Specifies whether the template should be used for **Only this layer** or **Everything**	

Control	Type	What it does	Notes
		that is being exported.	
Export Dialog: General Controls			
Colorspace	dropdown	The colorspace to which the output values are converted. **Automatic** is determined by the file name, size, and type of data in file. In addition to the OCIO roles and colorspace options, there are also **Same as Native** and **Same as Working** options available in the dropdown menu. These allow you to maintain the same colorspace as the one that was native to the data, or as the one you used within Mari when you were working with the data.	
Small Textures	dropdown	With this option enabled, Mari exports any patches filled with a constant color as 8x8 pixel textures, no matter what resolution the patches are in Mari. Enabling **Small Textures** saves disk space if you're exporting a channel that hasn't been fully painted. Disabling **Small Textures** lets Mari always export patches at their full size (even if they haven't been painted).	Enabling **Small Textures** saves disk space if you're exporting a channel that hasn't been fully painted. Disable **Small Textures** if you want Mari to always export patches at their full size (even if they haven't been painted).
Alpha Channels	dropdown	Select **Keep** or **Remove** alpha channels when exporting textures. This can help decrease the resulting file size when exporting images that don’t need alpha channels.	
Export All	button	Exports images from all patches in the	

Control	Type	What it does	Notes
Patches		project.	
Cancel	button	Cancels the export and closes the dialog box.	

Import Dialog

The **Import** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Invoked by the import channel functions and import layer functions, the **Import** dialog box lets you specify options for importing channels or layers. This can be done by importing textures into existing layer stacks, layers, or as new channels, however, importing textures as a new channel only opens the **Import** dialog box after first selecting options for the new channel from the Add Channel Dialog.

How to Access It

- *Menubar* | **Channels** > **Import** > any of the dropdown options
- *Menubar* | **Layers** > **Import** > any of the dropdown options
- *Right-click* | Channels Palette > **Import** > any of the dropdown options
- *Right-click* | Layers Palette > **Import** > any of the dropdown options

Import Channel Dialog Fields

Control	Type	What it does	Notes
Named Files - The **Named files** tab does not display if your model contains more than 20 patches.			

Control	Type	What it does	Notes
U	information	Lists the U texture coordinate for the left edge of the patch.	
V	information	Lists the V texture coordinate for the bottom edge of the patch.	
UDIM	information	Lists the UDIM number for the patch.	
File	text field	Specify the file location to import from.	
	button	Browse to the file location you want to export the texture to.	
Sequence			
Channel/ Layer	information	The name of the channel or layer that is being imported.	This name is only displayed if a name has been assigned to the selected layer.
Path	text field, dropdown	The directory containing the files to import. Type the directory, select a recent directory from the dropdown, or use the navigation controls below.	Make sure you check the path and name carefully, especially if you have been swapping between assets - Mari defaults to the last place you imported from. Don't risk overwriting your textures!
Reset To Defaults	button	Resets the path to the default location.	
	button	Goes back to the previous directory.	
	button	Goes forward to the next directory that you have browsed.	Only available if you have already browsed back using the back button.
	button	Goes up one level in the directory hierarchy.	

Control	Type	What it does	Notes
	button	Creates a new folder in the current directory.	
	button	Displays the contents of the directory as icons and names only.	
	button	Displays the contents of the directory as a details view, with a sort option at the top of the listing.	
Filter	text field, dropdown	Lets you filter the image sequences displayed in the dialog box. As you enter the filter text, Mari updates the display to only show the image sequences that match your text.	
Full	Image	Indicates whether the selected image sequence has a full set of images for the current model. That is, for each patch on the model, there is a corresponding image in the image sequence. Green if the selected image sequence has a full set of images for the current model, yellow if the selected image sequence has a partial set of images for the current model, or red if the selected image sequence has no images for the current model.	
Images	information	Lists the image sequences found in the selected **Path**.	
Start	information	First UDIM number in the selected image sequence.	
End	information	Last UDIM number in the selected image sequence.	
Patch Count	information	Number of image patches in the image sequence.	

Control	Type	What it does	Notes
Frames	information	How many frames in the image sequence.	
Modified	information	Last modified date for the images in the sequence.	
Template	text field, dropdown	Specify or select a preset format for automatically naming and numbering files on import. Variables available are: • **$ENTITY** - the name of the object • **$CHANNEL** - the name of the channel • **$LAYER** - the name of the layer • **$NODE** - the name of the node • **$UDIM** - the UDIM patch value • **$FRAME** - the animation frame number • **$NUMBER** - number of the file within the batch (for example, "$NUMBER of $COUNT" gives strings like "22 of 28") • **$COUNT** - total number of files in the channel • **$COLORSPACE** - the chosen colorspace for the selected images. • **$[METADATA VALUE]** - any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project1", Mari replaces the variable "$PROJECT" with "Project1").	Don't use numbers in your texture name - if you do, the **Import** and **Export** dialog boxes won't recognize the sequence.
File Example	information	Displays an example of the selected template path and filename.	
Formats	Information	Displays a list of supported import file formats.	
Import Dialog: General Controls			
Colorspace	dropdown	The colorspace to which the input values are converted. **Automatic** is determined by the	

Control	Type	What it does	Notes
		file name, size, and type of data in file.	
Resize	dropdown	How Mari handles patch/image size mismatches (for example, attempting to import a 1024x1024 image to a 2048x2048 patch). You can select either to resize the **Images** to match the patches, or to resize the **Patches** to match the images.	
Alpha Channels	dropdown	Choose to either **Keep** or **Remove** alpha channels when importing textures. This can help decrease the resulting file size when importing images that don't need alpha channels.	
Import All Patches	button	Imports textures onto all patches in the project.	
Import Selected Patches	button	Appears instead of **Import All Patches** when at least one patch is selected.	
Cancel	button	Closes the dialog box without importing.	

Mari to Maya Export Dialog

The **Mari to Maya Export** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

The **Mari To Maya Export** dialog box lets you specify the file type and destination for exporting a channel from the current Mari project to a Maya **.ma** file. Exported textures are saved to a user-selected directory and the created **.ma** file is set up to reference those texture files. The **.ma** file is created with the correct UDIM offsets for the textures so that Maya can import and apply them to the model correctly.

How to Access It

• *Menubar* | **Python** > **Examples** > **Export for Maya.**

Mari to Maya Export Dialog Fields

Control	Type	What it does	Opens this dialog box	Notes
Output Folder	text field	Displays the selected filepath for the exported files. You can also manually type in the file location.		
Browse	button	Allows you to select the file location for the exported files from the **Select Directory for Export** dialog.	**Select Directory for Export** dialog box.	
Channels	list	Displays the channel(s) for the currently selected object.		Multiple channels can be selected at once.
+	button	Adds the selected channel(s) and destination directory to the **For Export** pane.		
-	button	Removes the selected channel(s) and destination directory from the **For Export** pane.		
For Export	information	Displays the channel(s) to be exported.		
8-bit Files	dropdown	Selects the bit depth of the exported images.		Available formats: **bmp, jpg, jpeg, png, ppm, psd, tga, tif**, **tiff**, **xbm,** and **xpm.**
16/32-bit Files	dropdown	Selects the bit depth of the exported images.		Available formats: **exr**, **psd**, **tif**, and **tiff.**

Control	Type	What it does	Opens this dialog box	Notes
Force Overwrite	checkbox	Forces Mari to overwrite previously exported files to the same location, effectively replacing them.		
OK	button	Accepts the current channel export selections and exports them to the designated file location.	**Mari > Maya Progress** dialog box.	
Cancel	button	Cancels the current channel export selections and closes the **Mari To Maya Export** dialog box.		

Load Camera Dialog

The **Load Camera** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

The **Load Camera** dialog box comes up whenever you attempt to load or import a camera from outside the current Mari project. The camera file types are **.abc** and **.fbx** only.

How to Access It

• *Menubar* | **Camera** > **Load Camera**

Load Camera Dialog Fields

Control	Type	What it does	Notes
Look in	text field, dropdown	The directory of the camera file that is to be loaded into the current project.	
	button	Goes back to the previous directory.	
	button	Goes forward to the next directory that you have browsed.	Only available if you have already browsed back using the back button.
	button	Goes up one level in the directory hierarchy.	
	button	Creates a new folder in the current directory.	
	button	Displays the contents of the directory as icons and names only.	
	button	Displays the contents of the directory as a details view, with a sort option at the top of the listing.	
File Name	text field	Displays the name of the selected file to be imported into the project.	This field is automatically filled in when a file is selected from the above directory location.
File Type	text field, dropdown	Displays the file formats available to import the selected file.	By default this should display **.abc**, **.fbx** or **.FBX**.

Render Turntable Dialog

The **Render Turntable** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Takes the scene as you can see it through the selected shader, and creates a series of images showing the model rotating through an axis. The axis of rotation is a vertical line through the current camera view, centered in the middle of the visible portion of the model. You can include custom text or thumbnails of reference images in the turntable.

How to Access It

• *Right-click* | Projector in Projectors Palette > **Render Turntable**

Render Turntable Dialog Fields

Control	Type	What it does	Notes
Settings			
Size	dropdown	Selects the size of the generated images.	
Antialias	dropdown	How much anti-aliasing to use in the generated images.	
Background	dropdown	Selects the background for the images (Gray, Transparent, or From Canvas).	From Canvas uses the current background in Mari.
Frames	text field, slider	Sets the number of frames to generate - how many images should be in the sequence.	Range: 1 to 1000. Default: 360
Play Animation	checkbox	If you are using animated models or textures, lets you render the turntable with the animation.	When you check this, the **Frames** control above becomes disabled. This is because the frame range is taken from the Play Controls Toolbar.

Control	Type	What it does	Notes
Offset	text field	If you are combining multiple turntables, lets you chain together sequences by offsetting the image numbers.	For example, with an offset of 0, the first image is MariTurntable_1.jpg. With an offset of 360, the first image is MariTurntable_361.jpg. You can use this to generate one sequence of frames for the x axis, then another for the y axis.
Filters	dropdown	If this is checked, the turntable images are produced with the LUT and any other applicable view filters applied. If this is not checked, no filters are applied to the images produced.	
Lighting	dropdown	Specify whether you want to render a turntable with **Flat**, **Basic**, or **Full** lighting.	
Info			
Comment	text field, dropdown	Sets a comment to include this at the bottom of the turntable.	
Text Size	text field, slider	Sets the text size for the comment.	Range: 0.10 to 10.00. Default: 1.00
Reference	text field, dropdown, button	Selects a reference image to include at the top right of your model.	Second dropdown specifies whether the colorspace is sRGB or Linear.
Output			
Path	text field, dropdown, button	Specifies where Mari should create the images.	Default: /local1
Template	text field,	Sets a template for the filenames.	This must include the

Control	Type	What it does	Notes
	dropdown		**$FRAME** variable, so Mari assigns the frame numbers correctly to the files. Mari can export turntables as either **.tif**, **.png**, **.jpg**, or **.tga** files - change the file extension in the template to set the file type. Default: MariTurntable_$FRAME.jpg
Command	text field, dropdown	Specifies any commands to run on files after creating them.	
Preview	button	Generates the first frame of the turntable, enabling you to check the placement of reference images and text.	

Diagnostic Turntable Dialog

The **Diagnostic Turntable** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Creates a series of images showing the scene rotating through an axis. The axis of rotation is a vertical line through the current camera view, centered in the middle of the visible portion of the scene. You can export multiple channels at once. Mari creates all the images using the default shader and flat lighting.

How to Access It

• *Right-click* | Projector in Projectors Palette > **Diagnostic Turntable**

Diagnostic Turntable Dialog Fields

Control	Type	What it does	Notes
Channel	list	Selects which channels to export turntables for.	
Settings			
Size	dropdown	Selects the size of the generated images.	
Antialias	dropdown	How much anti-aliasing to use in the generated images.	
Background	dropdown	Selects the background for the images (Gray, Transparent, or From Canvas).	From Canvas uses the current background in Mari.
Frames	text field, slider	Sets the number of frames to generate - how many images should be in the sequence.	Range: 1 to 1000. Default: 360
Play Animation	checkbox	If you are using animated models or textures, lets you render the turntable with the animation.	When you check this, the **Frames** control above becomes disabled. This is because the frame range is taken from the Play Controls Toolbar.
Filters	dropdown	If this is checked, the turntable images are produced with the LUT and any other applicable view filters applied. If this is not checked, no filters are applied to the images produced.	
Info			
Comment	text field, dropdown	Sets a comment to include this at the bottom of the turntable.	

Control	Type	What it does	Notes
Text Size	text field, slider	Sets the text size for the comment.	Range: 0.10 to 10.00. Default: 1.00
Reference	text field, dropdown, button	Selects a reference image to include at the top left of your model.	Second dropdown specifies whether the colorspace is sRGB or Linear.
Output			
Path	text field, dropdown, button	Specifies where Mari should create the images.	Default: /local1
Template	text field, dropdown	Sets a template for the filenames.	This must include the $FRAME variable, so Mari assigns the frame numbers correctly to the files. If you're generating turntables for multiple channels, this should include the $CHANNEL variable, as otherwise the later channels overwrite the earlier ones. Mari can export turntables as either **.tif**, **.png**, **.jpg**, or **.tga** files - change the file extension in the template to set the file type.
Command	text field, dropdown	Specifies any commands to run on files after creating them.	
Preview	button	Generates the first frame of the turntable, enabling you to check the placement of reference images and text.	

Quick Projection Dialog

The **Quick Projection** dialog is described below with functions for how to access it. A list of the controls on the dialog can be found in the table below.

What It Does

Sets the details that the quick projection tools use - the filename, file size, and lighting model for the quick project files.

How to Access It

- *Toolbars* | **Quick Projection**
- *Menubar* | **Camera** > **Quick Projection Settings**

Quick Projection Dialog Fields

Control	Type	What it does	Notes
Resolution	dropdown	Sets the resolution for the projected file.	
Color Depth	dropdown	Sets the color depth for the projected file.	
Clamp	checkbox	If enabled, Mari clamps the RGB values for the projected file.	
Path	text field, button	Specifies the location and filename for the projected file.	
Lighting	dropdown	Sets whether the projected file contains lighting information: • **None** - no lighting information. • **Separate File** - lighting information is stored in a separate file. For example: diffuse.lighting.psd. • **Layer** - lighting information is stored in a separate layer (**.psd** files only).	

USD Preview Export Dialog

As part of Mari's integrated support for the USD format, there is now a **USD Export** tab available in the **Export Manager**.

What It Does

The **USD Export** tab allows USD Preview Surface Shaders created in Mari to be flattened for export and used in other Hydra powered digital content creators, such as Katana.

How to Access It

- *Channels* | **Export Manager**
- *Menubar* |

USD Preview Export Dialog Fields

Control	Type	What it does	Notes
Object	checkbox	Enables and disables selection of objects for export.	Checked by default.
Shader	checkbox	Enables and disables selection of shader for export.	Shaders listed in this tab correspond with the object that it has been assigned to.
Shader/Export Item	checkbox	Select individual shader channels for export.	All channels toggled by default.
Size	dropdown	Provides a set list of dimension sizes on a per-shader channel basis.	Size dimension is automatically set to the same as source.

Control	Type	What it does	Notes
Color Space	dropdown	Provides a set list of color spaces on a per-shader channel basis.	Color space is automatically set to the same as source.
Depth	dropdown	Provides a set list of bit depth options on a per-shader channel basis.	Depth is automatically set to the same as source.
Format	dropdown	Provides a set list of export formats per-shader channel.	Defaults to **tif** if the channel has a bit depth of 8-bit. Defaults to **exr** format if the channel is 16-bit.
Target Directory	text field	Specifies a target destination for texture exports.	Maps to the user profile of your machine by default.
Texture File Name	text field	Specify a naming convention for texture exports.	Maps to the user profile of your machine by default. Defaults to Mari's naming variables. See Exporting Textures from Channels for more details.
USD Look File	text field	Specify a target destination for a USD Look File export.	Maps to the user profile of your machine by default.
USD Assembly File	text field	Specify a target destination for a USD Assembly export.	Maps to the user profile of your machine by default.
Override Depth	dropdown	Allows bit depth settings to be overridden by a single specified value.	Defaults to no override.

Control	Type	What it does	Notes
Override Resolution	dropdown	Allows resolution settings to be overridden by a single specified value.	Defaults to no override.
Root Name	text field	Allows for a root name to be specified.	Any root name specified to this parameter must have a **/** (forward slash) prefix.
USD Payload	text field	Specifies the file location of a USD payload.	To export USD Assembly files, you must specify an associated USD payload file.

Toolbars

The toolbars that are available in Mari are described in the following pages, along with example images, the controls that can be found on each toolbar, and a list of descriptions of both the toolbar and the controls on it.

Use the left menu to see the details for Mari's toolbars.

Project Toolbar

The Project Toolbar performs common project functions, such as creating, closing, and saving projects.

Project Toolbar Controls

Control	Type	What it does	Opens this dialog box	Notes
	button	Create a new project.	New Project Dialog	Name the project, select options, and add geometry.
	button	Close the current project.	**Save Changes Before Closing?** (if unsaved changes)	
	button	Save the current project.		
	button	Undo the last action.		See also History View Palette.
	button	Redo the last action.		See also History View Palette.
	button	Opens the **Export Manager** dialog for you to export channels and Bake Point nodes.	Project Toolbar	

Palettes Toolbar

The Palettes toolbar allows quick access to your Mari palettes.

Palettes Toolbar Controls

Controls	What it does	Notes
	Click the Pin Palette mode button to activate the Pin Palette mode. The Pin Palette icon shows it is activated. Open palettes, channel layer stacks, mask stacks, and adjustment stacks are automatically pinned to the UI and do not disappear when the cursor is moved away from them. Click the Float Palette mode button to activate the Float mode. The Float Palette icon shows it is activated. Open palettes, channel layer stacks, mask stacks, and adjustment stacks float on the UI, and disappear from the UI when the cursor is moved away from them.	
	Click the Hide button to display the **Palettes** toolbar with its icons only. Click the Show button to display the **Palettes** toolbar with its icons and palette names.	
Channels	Each object in a Mari project has its own set of channels. The **Channels** palette displays: A list of channels in the project. Buttons to add, duplicate, convert, or remove channels. Options for creating quick channels. Channel attribute information. See Channels Palette for more information.	
Colors	The **Colors** palette lets you view and change colors. The controls on this palette all work together - as you change the color through one control, the other controls update to show your change. See Colors Palette for more information.	

Controls	What it does	Notes
History View	The **History View** palette displays a list of actions performed in the current session. Click an action to step back to that version of the project. See History View Palette for more information.	
Image Manager	The **Image Manager** lets you store and work with images in your project, including using them for brushes and painting through them onto your model. For organization purposes, you can also add custom tabs to the palette by clicking the add tab button. See Image Manager Palette for more information.	
Layers	The **Layers** palette lets you view, create and edit layers. You can also add the following to individual layers, or layer groups: masks, adjustments, procedurals, and shader layers. See Layers Palette for more information.	
Lights	The **Lights** palette displays the lights in your project, and allows you to: • Customize the lights. • Turn each light on or off. • Move the lights around. • Reposition a light to the current camera position. • Render shadows for individual lights. See Lights Palette for more information.	
Modo Render	The **Modo Render** palette allows you to render a final image of your scene in Modo with your Mari channels mapped to a Modo shader, bake into Mari channels, preview render and bake, and save out the Mari scene as an **.lxo** file. See Modo Render Palette for more information.	
Multi-Paint	The Multi-Paint palette enables you to define your Multi-Paint streams and settings.	

Controls	What it does	Notes
	See Multi-Paint Palette for more information.	
Node Graph	The **Node Graph** palette doesn't have any controls as such, rather it's a dedicated area designed to hold nodes. See Node Graph Palette for more information.	
Node Properties	The **Node Properties** palette doesn't have any controls as such, rather it's a dedicated area designed to hold the controls associated with nodes in the Node Graph. See Node Properties Palette for more information.	
Objects	The **Objects** palette displays the current objects in the project, and allows you to: • Show and hide the objects. • Add and remove objects and child objects from the project. • Lock and unlock the objects. • Select the object to edit. • Add versions to existing objects. • Add locators to objects. See Objects Palette for more information.	
Painting	The **Painting** palette displays information and controls for managing painting features. See Painting Palette for more information.	
Patches	The **Patches** palette displays a selectable list of patches in the project, with icons and buttons to toggle whether they are visible, or whether they are locked. You can group patches into sets. See Patches Palette for more information.	
Projectors	The **Projectors** palette displays the projectors in your project, and allows you to: • Create, load, save, and delete projectors.	

Controls	What it does	Notes
	• Use a projector to take a snapshot of the current view. • Use a projector to project an edited snapshot back onto the model's surface. See Projectors Palette for more information.	
Python Console	The **Python Console** palette is divided into three sections: A **Script Path** text field that loads Python scripts from a defined location. An input pane that is used to enter and execute Python statements in the lower half of the palette. An output pane that displays statements and their outputs in the upper half of the palette. See Python Console Palette for more information.	
Selection Groups	The **Selection Groups** palette lets you switch between saved selection groups. You can also create new selection groups, lock and unlock groups, and show or hide groups. See Selection Groups Palette for more information.	
Shaders	The **Shaders** palette displays the lighting shaders and allows you to toggle between the shader setup for the current channel, layer stack, layer, and paint target, as well as custom-made shaders. See Shaders Palette for more information.	
Shelf	The **Shelf** palette stores sets of brushes and colors. These include: • **Menu** shelf - items that you can select from the **F9** pie selection control menu. • **Personal** shelf (with customized items) - selected items you use regularly. • **Basic Brushes** shelf - a set of predefined basic brushes. • **Hard Surface Brushes** shelf - a set of predefined hard	

Controls	What it does	Notes
	surface brushes. • **Organic Brushes** shelf - a set of predefined organic brushes. • **Brad's New Brushes** - a set of predefined brushes. • **Project** - shelf items for just the current project. See Shelf Palette for more information.	
Snapshots	The **Snapshots** palette displays: • A list of snapshots for the entire project or currently selected channel. • Buttons to take, revert, or delete channel snapshots. • Information about snapshots, such as name, when they were created and by what artist. See Snapshots Palette for more information.	
Texture Sets	The **Texture Sets** palette manages texture sets or images (for example Megascans assets) that you can use for painting. You can browse and import texture sets to the **Image Manager** palette to work with images in your project. See Texture Sets Palette for more information.	
Tool Properties	The **Tool Properties** palette displays information about the selected tool. See Tool Properties Palette for more information.	

Paint Buffer Toolbar

The **Paint Buffer** toolbar clears all not yet baked paint and reset the paint buffer transform.

Paint Buffer Toolbar Controls

Controls	What it does	Notes
■	Clears all painting not yet baked into the model (still in the paint buffer).	
■	Resets the paint buffer transforms.	

Tools Toolbar

The Tools toolbar selects a tool for painting.

Note: The Tools toolbar contains group buttons. Click and hold the buttons that show the little down arrow to display other available tools, then hover over the tool you want to select and release the cursor. This selects the tool. The tools that show on the toolbar are the last ones used. The default layout is restored once you open a new Mari session or if you select **View** > **Default Layout**.

Tools Toolbar Controls

Control		Type	What it does	Opens this dialog box	Notes
Objects tools		button	Activate the **Select** tool. This tool lets you select areas on the model(s) in your project.		You can select whole objects, patches, or areas on the

Control		Type	What it does	Opens this dialog box	Notes
					surface of the objects. Once you have made a selection, you can choose to show or hide the selected areas, or to lock them for editing.
		button	Activate the **Transform Selected Objects** tool. This tool lets you move objects and lights around on the canvas by either clicking and dragging them or using the transform handles.		
		button	Activate the **Marquee Select** tool. This tool lets you select areas on the screen, to control where you can apply paint.		When you've made a selection, paint only bakes down within the selected areas.
Painting tools		button	Activate the **Paint** tool. This tool lets you draw paint strokes in the paint buffer.		
		button	Activate the **Blur** tool. This tool lets you blur paint already baked on a model.		If you want to blur large areas of the surface, you may want to use the blur Filter Functions

Control		Type	What it does	Opens this dialog box	Notes
					instead. These apply a controlled blur to entire patches or objects. Mari includes several blur filters, including a controllable Gaussian blur.
		button	Activate the **Paint Buffer Eraser** tool. This tool lets you erase paint from the paint buffer. It does not affect any underlying paint baked onto the model.		To clear the entire paint buffer, click the button in the **Paint Buffer** toolbar.
		button	Activate the **Vector Paint** tool. This tool lets you draw vector paint strokes in the paint buffer.		
Image Painting tools		button	Activate the **Paint Through** tool. This tool lets you position an image over an object, and then as you paint, paints the image "through" onto your model.		
		button	Activate the **Gradient** tool. This lets you create a color gradient floating over your model, which you can paint through to stamp onto your		

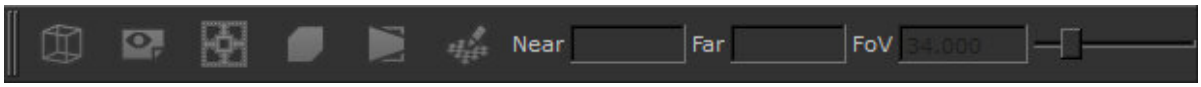
Control		Type	What it does	Opens this dialog box	Notes
			model (like the **Paint Through** tool).		
		button	Activate the **Clone Stamp** tool. This tool lets you clone from an existing image, shown in a separate window. You set a source point on the image. As you paint, Mari copies the texture around that source point onto your model.		You can also clone stamp from painting already on an object, in the paint buffer, or in another channel.
Paint Transformation tools		button	Activate the **Warp** tool. This tool lets you warp a selected area within your paint buffer. You create a warp box - a rectangle with between 4 and 64 control points.		
		button	Activate the **Slerp** tool. This tool works like the Liquify tool in Adobe® Photoshop®. It lets you "pull" the contents of the paint buffer around.		As well as pulling, **Slerp** lets you grow, shrink, and rotate the paint. You can also use it to selectively erase distortions you've already applied.
		button	Activate the **Pinup** tool. This tool works like the **Warp** tool, but lets you set your own control points (pins) wherever you want.		**Shift**-click to create a pin.

Control		Type	What it does	Opens this dialog box	Notes
		button	Activate the **Towbrush** tool. This tool lets you select areas of paint and drag them around the surface of your model. As you tow the paint around, it blends the edges of your selection with the surrounding paint. It works like the **CloneStamp** tool, but blends in the edges of the cloned selection.		
Paint Buffer tools		button	Activate the **Marquee Select** tool. This tool lets you select areas on the screen, to control where you can apply paint.		When you've made a selection, paint only bakes down within the selected areas.
		button	Activate the **Transform Paint Buffer** tool. This tool lets you resize, rotate, and move the paint buffer (and any unbaked paint in the buffer). It works on the entire paint buffer at once.		You can also make these transformations by manually editing the values under **Transform** in the **Paint Buffer** section of the Painting Palette.
		button	Activate the **Zoom Paint Buffer** tool. This tool lets you zoom in and out of the scene with the paint buffer locked to the object and without		You can also enable this tool by pressing **Z**. Zooming can be set to be reset

Control		Type	What it does	Opens this dialog box	Notes
			baking the paint.		when paint is baked.
		button	Activate the **Vector Inspector** tool. This tool lets you control the appearance of the directional markers for painted vectors.		You can opt to enable or disable the visibility of the directional markers in the **Vector Inspector** toolbar. If the markers don't appear when you select the tool, ensure that this option is set to enabled.
		button	Activate the **Color Picker/Pixel Analyzer** tool. This tool lets you pick a color from the paint on the model.		
Color swatch		button	Select a foreground color.	Select Color	
		button	Swap the foreground and background colors.		
Color swatch		button	Select a background color.	Select Color	
		button	Set the foreground color to white and the background color to black.		

Canvas toolbar

The Canvas Toolbar lets you set the view options for the UV and 3D views.



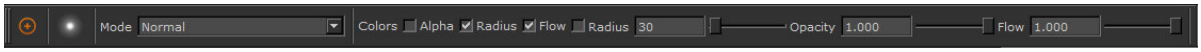
Canvas Toolbar Controls

Control	Type	What it does	Notes
	button	Toggle whether to show the wireframes on the model.	You can also press **Shift**+**W** to do this.
	button	Toggle between showing the patches in the UV view: • as they appear as part of the model, and • as they appear when exported (that is, as square images, with over-paint areas around the corners of the patches).	
	button	Toggle whether paint can be projected onto UV patches located outside the shells of the original geometry.	This control is only available in UV view.
	button	Switch to the orthographic camera view.	
	button	Switch to the perspective camera view.	In this view, you can set the near and far clipping planes and the field of view.
	button	Switch to the UV camera view (the **UV** tab). This gives you a “flat” view of the patches in the model.	

Control	Type	What it does	Notes
Near	text field	Set the value for the near clipping plane. Mari doesn't display parts of the scene that are closer than the Near plane.	This control is only available in the perspective camera view.
Far	text field	Set the value for the far clipping plane. Mari doesn't display parts of the scene that are further than the Far plane.	This control is only available in the perspective camera view.
FoV	text field, slider	Set the value for the field of vision. This controls how much Mari distorts the view when applying perspective. At 0, the perspective camera gives exactly the same view as the ortho camera. As the value increases, the distortion increases.	This control is only available in the perspective camera view.

Tool Properties Toolbar

The Tool Properties toolbar displays information about the selected tool.



Tool Properties Toolbar Control

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Facing	dropdown	Set whether the selection only applies to the **Front**		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			of the model as you can see it, or if the selection goes **Through** the model and includes the other side.		
		button	Activate **Rectangle Selection** mode. This lets you draw a rectangular selection.		This control only works in **Face** selection mode.
		button	Activate **Lasso Selection** mode. This lets you draw the outline of your selected area. You can draw any shape you want.		This control only works in **Face** selection mode.
		button	Activate the **Polygonal Lasso Selection** mode. This lets you draw the outline of your selected area by setting points between polygon sides until you close the selection.		You can close the selection by double-clicking on the canvas, pressing **Enter**, or clicking on the first point.
		button	Activate **Smart Selection** mode. This lets you create selections based on the surface of the model.		Smart selection only works in **Face** selection mode.

Tool	Control	Type	What it does	Opens this dialog box	Notes
			Tip: You can also make a smart selection by double-clicking a geometry while any selection tool shape is selected (square, lasso, or polygonal lasso). This way you can combine standard selection and smart selection together.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Type	dropdown	Choose whether to base your smart selection on: • **Connectedness Mesh** - selects the areas on the model connected in 3D view. • **Connectedness UV** - selects the areas on the model connected to the selected face. When you look at the area in the UV view, this selects the whole connected area within the current UV patch. If the UV patch holds several different areas, only those areas you click on are selected. • **Edge Angle** - selects based on the degree of change between neighboring faces. The selection goes until it hits an angle higher than the **Angle**. • **Orientation** - selects the area that faces in the same direction as your initial selection. • **Selection Group** - selects the whole		This control only works in **Smart Selection** mode.

Tool	Control	Type	What it does	Opens this dialog box	Notes
		• **Connectedness Mesh** - selects the areas on the model connected in 3D view. • **Connectedness UV** - selects the areas on the model connected to the selected face. When you look at the area in the UV view, this selects the whole connected area within the current UV patch. If the UV patch holds several different areas, only those areas you click on are selected. • **Edge Angle** - selects based on the degree of change between neighboring	selection group that the selected face belongs to. If the selected face does not belong to a selection group, nothing happens. If a face belongs to more than one selection group, clicking repeatedly cycles through the other selection groups to which it belongs.	**Smart Selection** mode.	

Tool	Control	Type	What it does	Opens this dialog box	Notes
		faces. The selection goes until it hits an angle higher than the **Angle**. • **Orientation** - selects the area that faces in the same direction as your initial selection. • **Selection Group** - selects the whole selection group that the selected face belongs to. If the selected face does not belong to a selection group, nothing happens. If a face belongs to more than one selection group, clicking repeatedly cycles through the other selection groups to which it belongs.			

Tool	Control	Type	What it does	Opens this dialog box	Notes
Angle	text field, slider	Set the maximum angle for the smart selection. • For **Edge Angles**, this sets the maximum angle between areas on the model surface. For example, if the angle is set to 30, this selects areas up to a 30 degree change of facing. • For **Orientation**, this sets how far from the facing of your original selection the selection goes to. For example, if the angle is set to 30, this selects areas up to 30 degrees away from the facing of your original selection.		This control only works in **Smart Selectio n** mode.	

Tool	Control	Type	What it does	Opens this dialog box	Notes
All	button	Select all items.			
None	button	Unselect all items.			
Invert	button	Invert the current selection. Unselected areas become selected, and the other way around.			
Grow	button	Grow the current selection.		This control only works in **Face** selection mode.	
Shrink	button	Shrink the current selection.		This control only works in **Face** selection mode.	
Grow/ Shrink by	dropdown	Choose whether to grow or shrink the current selection by **Vertex** or **Edge**.		This control only works in **Face** selection mode.	
	No				

Tool	Control	Type	What it does	Opens this dialog box	Notes
	properties.				
	Mode	dropdown	The blending mode used when the paint bakes down onto the channel surface. The default is **Normal** (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.		
	Colors	checkbox	Whether more pressure causes the color to vary (from slightly darker to the target color).		
	Alpha	checkbox	Whether more pressure increases the opacity.		
	Radius	checkbox	Whether more pressure increases the radius.		
	Flow	checkbox	Whether more pressure increases the flow.		
	Radius	text field	Set how big the brush tip is (in pixels).		
	Opacity	text field	How opaque the paint is. This is a multiplier on the paint buffer contents. At 1.0, the paint bakes down to the surface with the same		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			opacity as it is in the buffer. At 0.5, the paint applied to the surface is half as transparent as when it's in the buffer.		
	Flow	text field	Mimics how quickly paint is applied, by setting the maximum opacity in a splat.		
	Same as Paint, plus:				
	Blur	text field, slider	Set the blur strength. The higher the value, the more blur is produced.		
	Same as Paint.				
	Same as Paint				
	Mode	dropdown	Select the layer and painting blend modes.		
	Image Clone	checkbox	Instead of painting through an image, the **Image Clone** option clones from the image. **Ctrl**+click the image to set a clone source point before painting.		
	Stamp	button	Stamp the image straight		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			onto the model.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Toggle Repeat	button	Toggle source image tiling on and off. If this is on, when you paint off the edge of the image, Mari repeats the image (so you always have source data).		
	Tint	button	Tint combines the foreground color with the paint through texture.		
	Stencil	dropdown	Stencil uses the **Paint Through** texture's alpha channel as a mask and paints the foreground color. There are three options available: • **No Stencil** - if this is selected, Mari ignores the stencil color and follows the alpha value. • **Stencil** - if this is selected, paint is applied to the foreground color. • **Inverted Stencil** - if this is selected, paint is applied in everything that is transparent.		
	Luminance	dropdown	Luminance uses the Paint Through		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			texture's luminance instead of the alpha, but works similarly to **Stencil.** The following options exist within the same dropdown menu as the Stencil options: • **Luminance** - Works well for dirt, noise, and other similar images by using the luminance value of an image instead of the alpha. • **Inverted Luminance** - Similar to the luminance option, but the calculated luminance value is inverted before use.		
	Reset Image	button	Reset any transformations (such as moving, resizing, or rotating) you have made to the source image that you're cloning.		
	Pan Lock	checkbox	Lock the image position relative to the model. If you pan the model, the image that you're painting through pans too.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Scale Lock	checkbox	Lock the image size relative to the model. If you zoom the model in or out, the image that you're painting through zooms in or out too.		
	Same as Paint, plus:				
	Color swatch		Select the start color for the gradient.	Select Color	
	Color swatch		Select the end color for the gradient.	Select Color	
	In Point	text field	How far through the gradient the start color lasts (before it starts graduating into the end color).		
	Out Point	text field	How far through the gradient the end color reaches.		
	Type	dropdown	Select between **Linear** or **Radial** gradient. A linear gradient has a color at one end grading into the other color. A radial gradient has a color in the middle, radiating out		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			in a circle to the other color on the outside.		
	Same as Paint, plus:				
	Source	dropdown	Clone from: • **Current Paint Target** - the model's surface (baked paint) in the active channel. • **Painting** - the current (unbaked) paint in the paint buffer. • **Image** - an external image (selected from the **Image Manager**, optionally zoomed in or out). • Any of the other channels in the project.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Clear	button	Remove the current warp.		
	+	button	Increase the number of control points (up to 64) in the warp grid.		
	-	button	Decrease the number of control points in the warp grid.		
	Toggle Grid	button	Toggle between hiding and displaying the grid lines connecting the control points.		
	Mode	dropdown	Select how to use the Slerp tool: • **Pull** - pulls paint around. • **Grow** - makes the paint in a specific area larger. • **Shrink** - makes the paint in a specific area smaller. • **Rotate** - rotates paint around. • **Erase** - lets you undo the distortion in a specific area of the painting. You can also: • press ' (apostrophe) to		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			undo all the distortions you've applied, or • press **;**(semicolon) to apply your changes to the paint buffer (that is, make them so you can't erase the distortions).		
	Radius	text field, slider	Set the slerp radius (that is, how big the brush tip is).		
	Opacity	text field, slider	Set the strength of the Slerp effect.		
	Reset Pins	button	Undo all the distortions you've applied, moving the pins back to their original positions.		
	Clear Pins	button	Remove all the current pins.		
	+	button	Increase the strength of the currently selected pin. Stronger pins pull more of the surrounding texture towards them.		
	-	button	Decrease the strength of the currently selected pin.		
	Apply	button	Apply your changes to the paint		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			buffer and remove all current pins.		
		button	Clone from the selected region to another part of the surface.		
		button	Overwrite the selected region with another part of the surface.		
		button	Swap the contents of the selected region with another area on the surface.		
	Falloff	text field, slider	Controls how the texture blends in from the edges of the patch. Higher values have sharper transition to the surrounding texture, lower values blend in more smoothly. Use { and } to lower and raise the falloff.		
	Value	text field, slider	How much of the color of the original patch to preserve. Higher values keep more of the original color, lower values blend the whole patch in with the background as you move it. Use [and] to lower and raise the value.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
	Local	dropdown	Set the gimbal on the Transform Selected Objects tool to use the object's **Local** space for translation, rotation, scale, and free movement.		Local space uses the selected object's pivot as the origin for the movement for the object. The gimbal's axes move with the pivot of the object.
	Global	dropdown	Set the gimbal on the Transform Selected Objects tool to use **Global** (world) space for translation, rotation, scale, and free movement.		Global (or world) space uses the entire scene's origin as the movement for the object. The gimbal's axes always align with the scene's origin.
		button	Activate **Rectangle Selection** mode. This lets you draw a rectangular selection.		
		button	Activate **Lasso Selection** mode. This lets you draw the outline of your selected area. You can draw any shape you want.		
		button	Activate the **Polygonal Lasso Selection** mode.		You can close the selection by

Tool	Control	Type	What it does	Opens this dialog box	Notes
			This lets you draw the outline of your selected area by setting points between polygon sides until you close the selection.		double-clicking on the canvas, pressing **Enter**, or clicking on the first point.
		button	Activate **Ellipse Selection** mode. This lets you draw an elliptical selection.		
		button	Activate **MagicWand Selection** mode. This lets you select by color. When you click, Mari creates a selection around the point you clicked. The selection is based on color – areas with a similar color to the original point are selected. You can change the color threshold for the selection.		
		button	How drawing a new selection affects the current marquee selection: • **Replace** - the new selection replaces the existing selection. • **Transform** - this lets you move your selection area around		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			(move, rotate, or scale the selection). In **Replace** mode (the default), you can: • hold down **Shift**, click and drag to add to the current selection, or • hold down **Ctrl**/**Cmd**, click and drag to remove from the current selection. In **Transform** mode, you can: • hold down **Shift**, click and drag to move the current selection, • hold down **Ctrl**/**Cmd**, click and drag to rotate the current selection, or • hold down **Shift**+**Ctrl**/**Cmd**, click and drag to scale the current selection.		
		button	Fill the selected paint buffer area with the current foreground color.		
		button	Invert the current		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			selection. Unselected areas become selected, and the other way around.		
		button	Lift the selected texture and store it in the paint buffer.		
		button	Clear the current selection.		
		text field	Feathering - Soften the edges of the selection. Higher values give the selection a softer, fuzzier edge. At 0, the selection has a hard edge.		
	Reset Buffer	button	Reset any transformations (moves, rotations, scales) applied to the paint buffer (and any unbaked paint in the buffer).		
	Reset on Bake	checkbox	Controls whether Mari resets the paint buffer to its defaults after baking. If enabled, any transformations you've made to the paint buffer revert to the defaults. That is, after baking you get a new paint buffer (once again slightly larger		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			than the visible screen). If you want to preserve your transformations after baking, change this to disabled.		
	Reset	button	Reset the zoom level of the paint buffer.		
	Zoom In	button	Zoom in on the paint buffer.		
	Zoom Out	button	Zoom out of the paint buffer.		
	Zoom	information	Displays the paint buffer's zoom percentage.		
	Reset on Bake	Checkbox	Controls whether Mari resets the paint buffer to its defaults after baking. If enabled, any transformations you've made to the paint buffer revert to the defaults. That is, after baking you get a new paint buffer (once again slightly larger than the visible screen). If you want to preserve your transformations after baking, change this to disabled.		
	Reset	button	Reset the settings of the vector inspector.		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			Can also be done by using the ' (apostrophe) shortcut when the vector inspector is active.		
	Enabled	checkbox	Enable (or disable) the visible directional markers for the vector paint tool. Can also be done by using the **Shift**+**I** shortcut.		
	Spacing	text field, slider	Adjust the spacing between the directional markers. Can also be done by **Shift**+dragging the mouse when the vector inspector is active.		
	Length	text field, slider	Adjust the length of the directional markers.		
	Start Color	button	Change the color at the base of the directional markers.		
	End Color	button	Change the color at the tip of the directional markers.		
	Vector Space	dropdown	Set whether the vector inspector		

Tool	Control	Type	What it does	Opens this dialog box	Notes
			operates in **ScreenSpace**, **TangentSpace**, or **WorldSpace**.		
	Sample Size	text field, slider	Radius in pixels that the eyedropper uses when sampling colors off the surface. The eyedropper takes the average value of all the pixels in this area, and sets this as the foreground color.		
Color swatch	**button**	Select a foreground color.	Select Color		Color swatch
	button	Swap the foreground and background colors.			
Color swatch	**button**	Select a background color.	Select Color		Color swatch
	button	Set the foreground color to white and the background color to black.			

Ptex Toolbar

The Ptex toolbar displays a number of options for managing Ptex faces within your project. The toolbar is only available for Ptex channels.

Ptex Toolbar Controls

Control	Type	What it does	Notes
	button	Double the selected face resolution.	
	button	Halve the selected face resolution.	
	button, text field	Set the world space face resolution for the selected face.	Enter the **Density** and then click .
	button, dropdown	Set the face resolution for the selected face.	Select the **Size** and then click .
	dropdown	Fill the selected faces with the foreground color.	
	button	Fill the selected faces with the background color.	

View Transform Toolbar

The **View Transform** Toolbar displays a number of options for managing your monitor's colorspace in Mari.

> **Note:** You can toggle Color Management in the Project Settings Dialog. When Color Management is disabled, you can still enable the View Transform toolbar by clicking the /Color/Scalar icon.

View Transform toolbar when Color Management is turned on:

View Transform toolbar when Color Management is turned off:

Select a channel in the Channels Palette and use the toolbar to choose the view transform and edit the LUT etc. The color icon on the left is gray if the channel contains scalar data, otherwise it is color to indicate color data. (For a more detailed explanation, see Color Data and Scalar Data).

You can change the view transform's canvas colorspace using the dropdown list, and revert to the default colorspace using the reset button.

View Transform Controls

Control	Type	What it does	Notes
	button	Toggles on and off the View Transform. Also indicates whether you are viewing color or scalar data. - The data is considered as color data and is being color managed. - The data is considered as non-color data, specifically scalar data, and has no color management applied. **Note:** The View Transform scalar mode is automatically enabled when viewing nodes that connect to scalar streams of Multi-Channel nodes, Material nodes, or Shader nodes.	Click to toggle. See Color Data and Scalar Data for more information.
	button	Select an OCIO configuration file.	
Input	dropdown	Set the colorspace of the current	

Control	Type	What it does	Notes
Colorspace		channel or painting in the **Input Colorspace** dropdown. (This is remembered for each channel.)	
View Transform	dropdown	The View Transform operation is applied to the canvas before it is displayed on the screen. It is typically used to convert from the working colorspace to your monitor's colorspace.	
	button	Reset the View Transform colorspace back to the default as set in the Project Settings Dialog. If it's a color channel, the **Color Monitor** project setting is used. If it's scalar, the **Scalar Monitor** project setting is used.	
	button	• Select a custom LUT to apply to the scene. • Clear the current LUT applied to the scene. • Set whether the GPU extrapolates the grid values or clamps to the maximum value of the LUT.	
Component	dropdown	View the individual channels for the scene.	
F-Stop, Gain	incrementer, text field, slider	Set the number of f-stops or the amount of gain correction applied before the view transform.	You can boost or reduce gain by entering a multiplier (exposure value), dragging on the slider, or adjusting the F-Stop value.
Gamma	text field,	Set the amount of gamma	You can boost or reduce gamma

Control	Type	What it does	Notes
	slider	correction applied after the gain viewer process.	by entering a gamma level or dragging on the slider.

Navigation Toolbar

The Navigation toolbar displays a number of options for managing navigation in Mari.

Navigation Toolbar Controls

Control	Type	What it does	Notes
	button	Resets all the options in the **Navigation toolbar** to the default settings.	
	button, dropdown	Provides the following options for panning: • **Pan All** - allows panning in any direction. • **Pan X** - allows panning on the x axis only. • **Pan Y** - allows panning on the y axis only. • **Pan Disabled** - disables all panning.	The button changes to reflect the selected option.
	button, dropdown	Provides the options to either enable or disable zooming.	The button changes to reflect the selected option.
	button, dropdown	Provides the options to either enable or disable rolling.	The button changes to reflect the selected option.

Control	Type	What it does	Notes
	button, dropdown	Provides the following options for orbiting: • **Orbit All** - allows orbiting around the model in any direction. • **Orbit X** - allows orbiting around the model on the x axis only. • **Orbit Y** - allows orbiting around the model on the y axis only. • **Orbit Disabled** - disables all orbiting.	The button changes to reflect the selected option.
	button, dropdown	Provides the following options: • **Snap 45 degrees** - when rotating the model it snaps at 45 degree angles. • **Snap 90 degrees** - when rotating the model it snaps at 90 degree angles. • **Snap angle disabled** - when rotating the model it does not snap to any angle.	The button changes to reflect the selected option.

Vector Painting Toolbar

The Vector Painting toolbar displays the option to toggle the Vector Inspector on or off, and to create default vector shaders. When the tool is active, the Vector Inspector markers are shown, regardless of whether you toggle the option on or off.

Vector Painting Toolbar Controls

Control	Type	What it does	Notes
	button	Quickly enables the vector inspector so that the directional markers are visible. To quickly turn off the directional markers, click the button again.	Can also be done by using the **Shift**+**I** shortcut.
	button	Sets up two default vector shaders with default channels and layers for painting flow or normal maps. Also changes the tool to **Vector Paint** and switches the Painting **Mode** to **Paint Flow Vectors.**	Can also be done by navigating to **Python** | **Examples** > **Setup Vector Brush**.

Project Controls toolbar

The **Project Controls** toolbar displays a set of four tools:

- **Selection** - lets you switch between three selection modes (objects, patches, and faces) when using the **Select** tool. See Selecting Objects, Patches, or Faces.
- Default Shaders - displays one of the four default shaders in the **Shaders** palette and the last-selected user shader: **Current Channel**, **Current Layer and Below**, **Current Layer**, **Current Paint Target**, and **User Shader**. Selecting a custom created shader in the **Shaders** palette won't highlight any of the shaders in the toolbar, but selecting a default shader from either the palette or the toolbar activates the shader and highlights the icon in the toolbar. See Shaders.
- **Lighting** - lets you change the lighting on the object. You can select one of the five lighting modes: **Flat**, **Basic**, **Basic with Shadows**, **Full**, or **Full with Shadows**. By default, lighting is relative to the scene, not the view. You can set the individual lights to move the view. Mari saves custom lighting with the project. If you need the same lights in another project, you need to set them up separately. See Configuring the Lighting.
- **Paint Buffer Symmetry** - displays the four **Mirror Painting** modes for the Paint Buffer Symmetry functionality. See Paint Buffer Symmetry.
- **Mirror Projection** - displays the four **Mirror Projecting** modes for the Mirror Projection functionality. See Mirror Projection.

Project Controls Toolbar Controls

Tool	Control	Type	What it does	Notes
		button	Selects whole objects.	
		button	Selects patches on objects.	You can hold down **Shift** to add to the selection, **Ctrl**/**Cmd** to remove from the selection, and (**Windows**) or **Shift**+**Cmd** to intersect the selection.
		button	Selects areas on objects.	You can hold down **Shift** to add to the selection, **Ctrl**/**Cmd** to remove from the selection, and (**Windows**) or **Shift**+**Cmd** to intersect the selection. When using the **Select** tool, double-click on the face of the model to create a smart selection based on the settings in the **Type** dropdown menu.
		button	Selects the default **Current Channel** shader from the **Shaders** palette and displays only the paint in the current channel.	
		button	Selects the default **Current Layer and Below** shader from the **Shaders** palette and displays only the paint in the current layer stack.	
		button	Selects the default **Current Layer** shader from the **Shaders** palette and displays only the paint in the current layer.	

Tool	Control	Type	What it does	Notes
		button	Selects the default **Current Paint Target** shader from the **Shaders** palette and displays only the paint of the currently selected target.	
		button	Select the **User Shader**, which is the last selected shader you added to the **Shaders** palette.	
		button	Set the lighting on the object to "Flat".	In this mode, the position of the lights has no effect.
		button	Set the lighting on the object to "Basic".	In this mode, you can move lights around on the canvas or make fine adjustments to lighting using the Lights Palette.
		button	Set the lighting on the object to "Basic with Shadows" - basic lighting with diffuse lighting, as well as shadows.	To enable shadow processing, in the Mari Preferences Dialog, in the **GPU** tab, enable the **Shadow Maps** > **Allowed** checkbox. By default this preference is disabled.
		button	Set the lighting on the object to "Full".	In this mode, you can move lights around on the canvas or make fine adjustments to lighting using the Lights Palette.
		button	Set the lighting on the object to "Full with Shadows" - full lighting with diffuse and specular lighting, as well as shadows.	To enable shadow processing, in the Mari Preferences Dialog, in the **GPU** tab, enable the **Shadow Maps** > **Allowed** checkbox. By default this preference is disabled.
		button	This is the default setting. Mirror projection is disabled.	
		button	With **Mirror Projecting X** selected, the mirror line appears	

Tool	Control	Type	What it does	Notes
			vertically. Paint strokes created on either the left or right side of the mirror plane project the same strokes on the other side.	
		button	With **Mirror Projecting Y** selected, the mirror line appears horizontally. Paint strokes created either on top or under the mirror plane project the same strokes on the other side.	
		button	With **Mirror Projecting Z** selected, the mirror line appears vertically intersecting the profile of the asset. Paint strokes created either at the front or back of the mirror plane are projected on the opposite side of the geometry.	
		button	This is the default setting. When mirroring is disabled, the paint buffer acts as normal and the paint strokes are not mirrored.	
		button	Mirroring left and right splits the paint buffer down the center vertically. Any paint strokes made on either the left or right side of the divider are mirrored on the opposite side.	This applies to paint strokes only.
		button	Mirroring top and bottom splits the paint buffer down the center horizontally. Any paint strokes made on either the top or bottom of the divider are mirrored on the opposite side.	This applies to paint strokes only.

Tool	Control	Type	What it does	Notes
		button	Mirroring four ways splits the paint buffer into quarters around the center of the canvas. Any paint strokes made in one of the quadrants are mirrored in the other three.	This applies to paint strokes only.

Status Bar

The status bar and descriptions of all the controls on it can be found in the tables below.

Note: The status bar also displays basic tool help for the current tool.

What it looks like	What it does
	Displays status icons that indicate which features are active, whether an error has been encountered, and information about the current project. This includes progress bars on running processes.

Status Bar Icons

Icon	What it does	Notes
	Displays the UDIM number of the patch you have selected. If nothing is selected, then there is nothing listed after **Udim**.	
	Notifies you that there is paint to bake.	Click on the icon to bake.
	Notifies you that background jobs are currently running.	Clicking on this icon displays a dialog for the progress bar.
	Notifies you that the project has changed and allows you to save.	Click on the icon to save changes.
	Notifies you that your painting is hidden.	
	Notifies you that you are currently projecting on the front	Clicking on this icon toggles

Icon	What it does	Notes
	of objects.	between projecting through objects and projecting on the front of objects. The icon takes the default icon's place when switching to project through.
	Notifies you that you currently have edge masking enabled.	
	Notifies you that you currently have channel masking enabled.	
	Notifies you that you currently have ambient occlusion masking enabled.	
	Notifies you that you currently have depth masking enabled.	
	Notifies you that you currently have backface masking enabled.	
	Notifies you that you currently have fractal noise masking enabled.	
	Notifies you that you currently have the global mask preview enabled.	Clicking on this icon disables the global mask preview.
	Notifies you whether you are in LDR (Low Dynamic Range) or HDR (High Dynamic Range) mode.	Clicking on this icon toggles between LDR and HDR. When in HDR mode, the icon appears as .
	Warns you that a low memory status has been detected. Low memory affects Mari's performance and may make it unstable.	Try freeing memory or closing and restarting Mari to clear this issue.
	Notifies you that there are warning messages present.	Clicking on this icon displays these warnings.

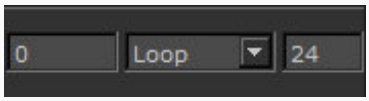
Icon	What it does	Notes
	Warns you that the cache disk is critically full.	If you don't free disk space, data loss may occur. A simple way to free disk space is to close the project and re-open it. Mari saves your session history, but closing your project clears this used disk space up for use.
	Warns you that there is no disk space left for the operation you are trying to run.	
	Warns you that channels are a higher bit-depth than what the virtual texture is configured for.	
	Notifies you that Mari is currently loading data from the disk.	
	Notifies you that Mari is currently loading data from the RAM to GPU memory.	
	Notifies you that Mari is listening for commands on [COMPUTER NAME] port [PORT NUMBER].	Hovering over this icon displays the computer name and port number on which Mari is listening for commands.
	Notifies you that there was an error while Mari was listening for commands.	
	Notifies you that Nuke is connected to send commands to Mari on [COMPUTER NAME] port [PORT NUMBER].	Hovering over this icon displays the computer name and port number from which Nuke is connected.
	Notifies you that the canvas updates are active. Click to toggle off ( ). Helpful if you don't want to wait for shader operations to compile in between tasks.	

Play Controls Toolbar

The **Play Controls** toolbar and the controls that can be found on it are described in the table below. The **Play Controls** toolbar allows you to play animations. You can set keyframes, which allows you to paint different textures in different parts of the animation sequence.

Play Controls Toolbar Controls

Control	Type	What it does	Notes
Timeline	timeline	Shows the position of the playhead. Keyframes appear as orange markers in the timeline.	
	text field	Sets the frame range: the start and end frames of the animation.	
	button	Adds a keyframe at the current frame.	
	button	Removes the selected keyframe.	
	button	Moves back a single frame.	
	button	Moves back to the beginning of the animation.	
	button	Plays the animation.	
	button	Stops playing.	
	button	Moves forward a single frame.	

Control	Type	What it does	Notes
	text field, dropdown	Skips to a particular frame. Select the play mode from the dropdown. One of: • **Loop** - plays the selected frames through, repeating again from the start when it finishes. • **Bounce** - plays the selected frames through from start to finish, then backwards from the finish to start, and then repeating. • **Stop** - plays the animation once and then stop. Sets the frame rate.	

Nuke<>Mari Bridge

Typically, visual effects involves creating digital environments, adding generated imagery and matte paintings to live action scenes, and creating clean plates by removing blemishes, wires, or rigs from existing footage. Building shots using a combination of these techniques can be a laborious task, especially if there is no effective bridge between applications.

Using the 2D and 3D environment in Nuke has helped speed up this process, but when composite projections need to be extended, blended, or recreated it becomes increasingly complex. Some projections may need to be warped to match other projections, or new projections may need to be created to fill in areas that are completely occluded by objects that have been removed from entire shots. Even when sufficient projection information exists, multiple projections may need to be blended together using paint tools to create a seamless composite.

A typical example of this is with matte painting. Once completed, a matte painting is often imported into a 3D system and projected onto a scenic model (such as a building or landscape). Unfortunately, as the 3D model stretches the painting, the projection only works well from the angle of the projection camera. Fixing this in a 3D system requires the creation of new projections where the stretching occurs and in turn, blending these projections with the original painting for every frame in a shot. Creating new projections in this manner and manipulating them on the fly is both technically and artistically difficult to achieve.

The Nuke<>Mari Bridge offers you a quick and easy way for fixing projection problems, converting projection textures to UV textures, and editing multiple projections in context using the rich and natural toolset of Mari.

Once installed, it's as simple as sending one or more **Project3D** nodes from Nuke to Mari (transferring all of the models, textures and projections in a shot) and using the Mari toolset to fix, enhance, or create projection setups. Once the work is finished in Mari, projection setups are sent back to Nuke where they appear immediately in the node tree.

The Nuke<>Mari Bridge handles all of the data management in the background and links both applications whether they are on the same machine, or on different machines on the same network.

Installation

> **Warning:** If you have any previous versions of the Nuke<>Mari Bridge scripts installed, please remove them first.

To install the Nuke<>Mari Bridge, follow these steps:

On Linux

1. Navigate to the **Media/Scripts/Nuke** sub-directory of the Mari application directory on your computer.
2. Copy the contents of this folder to the **/home/*login name*/.nuke** directory.

On Windows

1. Navigate to the **Bundle\Media\Scripts\Nuke** sub-directory of the Mari application directory on your computer.
2. Copy the contents of this folder to the **\Users*login name*\.nuke** directory.

Connection Setup

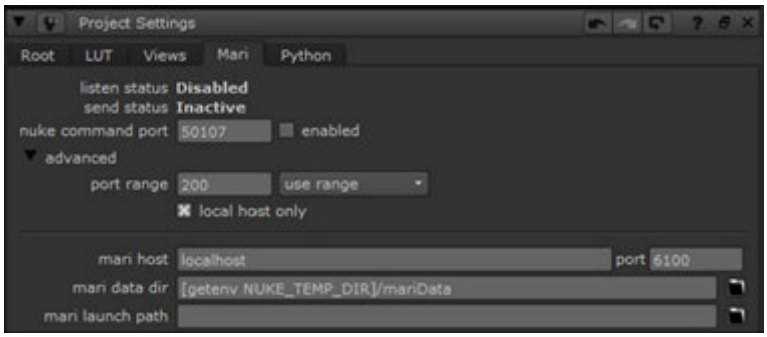
Before we can send a projection to Mari, we need to establish a connection between the two applications.

Nuke

When Mari is launched from inside Nuke or a command is sent to Mari through the **Mari** menu or toolbar, Nuke's command port is automatically enabled. To manually set this, or adjust other settings related to the **Nuke<>Mari Bridge:**

1. Click **Edit** > **Project Settings** > **Mari.**

   The **Project Settings** tab for Mari displays.



2. Check **enabled** and set the **command port**. When enabled, Nuke listens to Mari on the specified command port.

> **Note:** Only one Nuke session can listen on the given port number. If the port number is already in use, Nuke attempts to listen on another port number within a range starting at the given port number. To change this behavior, click **Edit** > **Project Settings** > **Mari** > **advanced** and adjust both the **port range** and **use range** settings.

3. If both Nuke and Mari are running on the same computer, you should leave the **mari host** field blank, and ensure **local host only** is checked. If they are running on separate machines, enter the machine name or IP address of the computer running Mari here and uncheck **local host only**. These settings are automatically set when a command is successfully sent from Mari to Nuke.

> **Note:** Checking **local host only** tells the application to only listen for connections from the local machine. This can prevent accidental connections from different machines and is therefore more secure, but this must be turned off when you want to use the Nuke<>Mari workflow across the network.

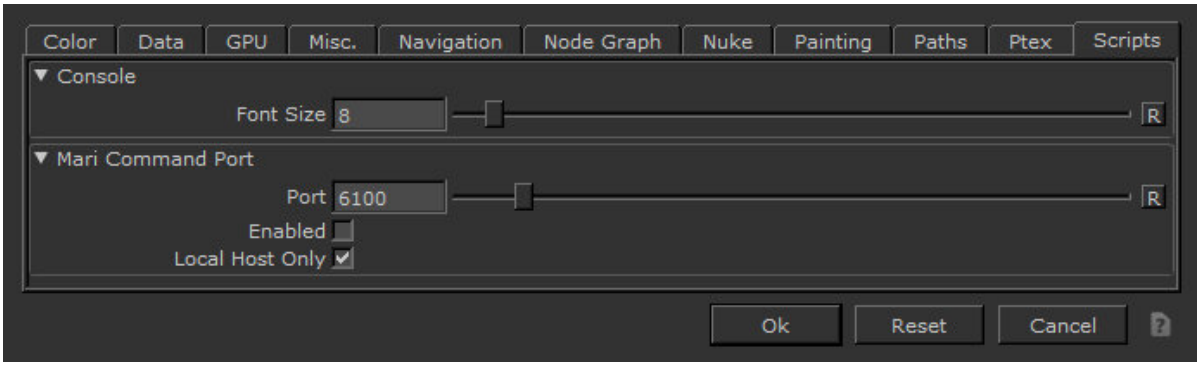
4. Enter the port number that Mari is set to receive commands on in the **port** field. This number should match the Mari **Command Port** number set in the **Scripts** section of the **Mari Preferences** dialog. This value should only be changed if the default is already in use by another application. This is automatically set when a command is sent from Mari to Nuke.
5. Set the **mari data dir.** This field specifies the location of the directory used for the data rendered and exported from Nuke and Mari.

> **Note:** The settings for connecting Nuke to Mari are stored separately in each script.

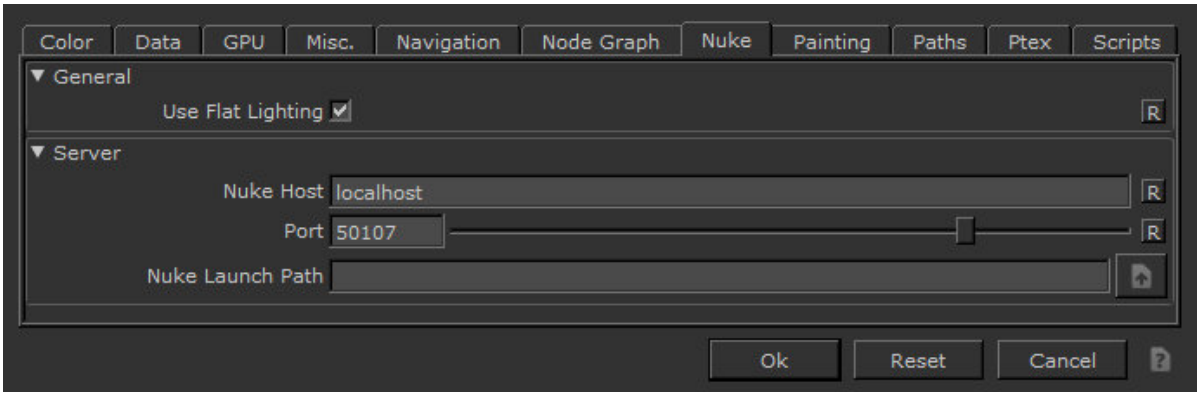
Mari

In Mari, there are a number of settings that we need to check before we can receive commands from Nuke.

1. Open the **Mari Preferences** dialog by navigating to **Edit** > **Preferences**.
2. Select the **Scripts** tab and ensure that the **Mari Command Port** is **Enabled**. The default **Port** number should only be changed if it is already in use by another application. This number should match the port number set in Nuke under **Project Settings** > **Mari**.



3. As with the Nuke connection setup, if both Nuke and Mari are running on the same computer, ensure **local host only** is checked. If they are running on separate machines, uncheck this. Refer to the note on the local host option in the Nuke section for more information.
4. In the **Mari Preferences** dialog, click on the **Nuke** tab.



5. Enter the machine name or IP address of the computer running Nuke in the **Nuke Host** field. If both Mari and Nuke are running on the same computer, you should leave this field blank. This is automatically set when a command is sent from Nuke to Mari.
6. Enter the port number that Nuke is set to receive commands on in the **Port** field. This number should match the Nuke **command port** set in Nuke under **Project Settings** > **Mari**. This value should only be changed if the default is already in use by another application. This is automatically set when a command is sent from Mari to Nuke.

   You are now ready to exchange data between Nuke and Mari.

Note: If you intend to launch Mari from Nuke from the **Mari** > **Launch Mari** menu, you can set the environment variable **MARI_INSTALL_PATH** so that you don't need to point to the Mari executable every time. For example, **MARI_INSTALL_PATH=/usr/local/Mari**<version>**/mari**.

Communicating with Mari

Mari uses connection ports to communicate with Nuke as well as other programs. Once the command port is set up in the **Mari Preferences** dialog, described in the Connection Setup section, you can connect to the port and send commands to control Mari.

The way to communicate with Mari is to:

1. Open the **Mari Preferences** dialog by navigating to **Edit > Preferences**.
2. Select the **Scripts** tab and set Mari to use port **6100**.
3. Open a python session, for instance, write 'python' in the terminal.
4. Set up the connection by typing the following:

```
>>> import telnetlib
>>> HOST = "localhost"
>>> PORT = 6100
>>> tn = telnetlib.Telnet(HOST,PORT)
```

Note: Mari does not execute the commands immediately. Instead, Mari buffers these commands until you send an EOT (end-of-transmission) character.

5. Send the EOT character by writing the following:

```
>>> tn.write("\x04")
```

For example, to create a Perlin noise node in the Node Graph, write the following:

```
>>> tn.write("ng = mari.geo.current().nodeGraph()\n")
>>> tn.write("perlin = ng.createNode('Procedural/Noise/Perlin')\n")
>>> tn.write("\x04”)
```

Workflow

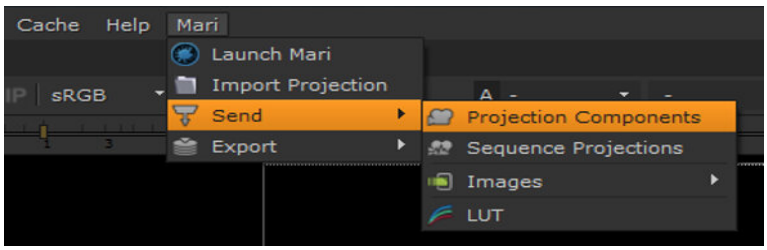
Sending a Projection Setup

To send a Nuke projection setup to Mari follow these steps:

1. Select the Project3D node and the Geometry node that the projection is applied to, or select multiple nodes at once to send to Mari.

> **Note:** If you want to merge the Nuke projection with a Mari project that is currently open, and the required geometry is already part of that project, you don't need to select the Geometry node.

2. If Mari is not already running, select **Mari** > **Launch Mari** or click on the Mari icon in the **Mari** panel to launch it.
3. In Nuke, select **Mari** > **Send** > **Projection Components** or click on the **Send** icon in the **Mari** menu.



This pre-renders the texture to the location specified in **mari data dir**. The name of the pre-rendered file is derived from the node name and the current frame. If you sent the texture to Mari before, you are asked if you want to overwrite the file. If you select **No**, the existing file is used.

Note: You can also opt to export the projection instead of sending it to Mari using the Nuke<>Mari bridge.

To export:

1. In Nuke, select **Mari** > **Export** > **Projection(s)...** or click on the export icon in the **Mari** panel.
2. Set a directory location in the **Export Mari Projection** dialog.
3. In Mari, select **Nuke** > **Import Projection** and select the file in the **Load Nuke Projection** dialog to import the projection into Mari.

This lets you send files between the two applications without a socket connection.

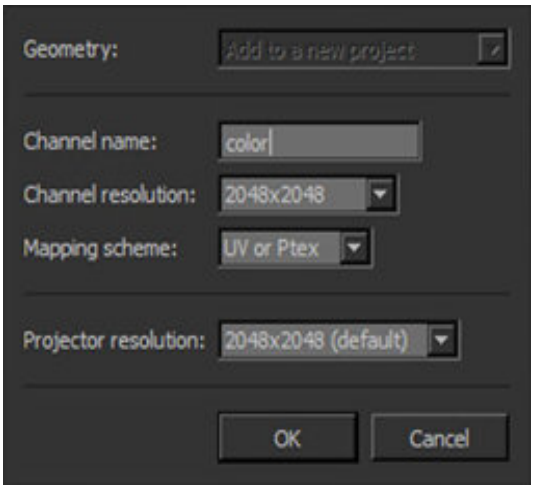
Note: When exporting files to disk in Nuke, an **.nmb** extension is assigned (for Nuke<>Mari Bridge). When exporting from Mari, an **.mnb** extension is used (for Mari<>Nuke Bridge).

Sending or Importing Projections to Mari

Once the projection has been sent or imported into Mari, follow these steps to set up the projections in your project:

1. If there is no project open in Mari, the script creates a new project and asks you to create a channel for your projection. In the **Mari - New Projection** dialog, you are prompted to enter a **Channel name**, set the **Mapping scheme**, provide a **Channel resolution** and finally, select the **Projector resolution** for the new channel.

Note: Ensure that you don't compromise the incoming image's resolution by selecting a texture size that is too low.



If you already have a project open (for example, if you are re-importing the projection), you are asked if you want to apply the incoming projection to the current project or open a new project.

2. Mari sets up:
   - objects in the **Objects** palette for each Geometry node sent or exported from Nuke,
   - projectors in the **Projectors** palette for each Project3D node sent or exported from Nuke,
   - the channel that you specified in the **Mari - New Projection** dialog in the **Channels** palette, and
   - all the sent or imported cameras, created as layers in the specified channel's layer stack, with masks to help blend projections together.

   Depending on the complexity of the projection being sent or imported into Mari, this may take some time.
3. Images are also imported into the **Image Manager** to keep track of the projection textures sent or imported into Mari.

Sending or Exporting Projections to Nuke

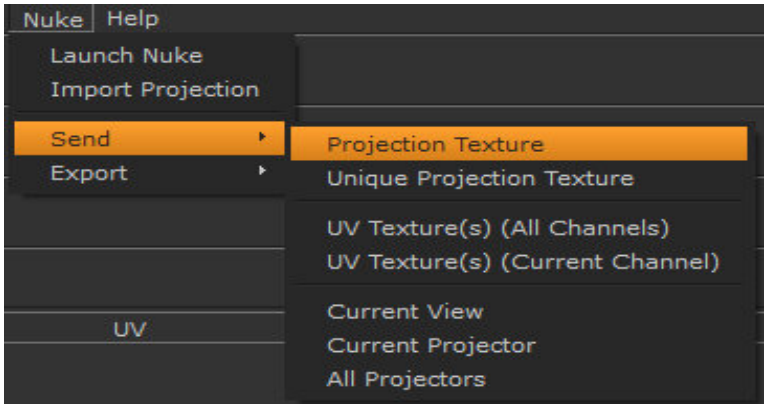
Once you have finished your texture work in Mari, you have several options to send the data back to Nuke. You can use one of the following. In Mari:

1. Select **Nuke** > **Send** > **Projection Texture** to unproject the texture through the current projector. The texture is rendered to the Mari data directory and the filename is based on the name of the projector that it is unprojected from.
2. Once the texture has been written to disk, Nuke creates a Read node. This Read node should be connected to the Project3D node, replacing the original texture with the new texture from Mari.
3. Select **Nuke** > **Send** > **Unique Projection Texture** to unproject the texture through the current projector, like **Projection Texture**. However, it saves the unprojected texture with a unique file name to ensure that it does not overwrite (or is potentially overwritten by) any other version of the texture that is sent to Nuke.

4. Select **Nuke** > **Send** > **UV Texture(s) (All Channels)** to export a set of UV textures from all channels in the **Channels** palette to the Mari data directory. The filename is based on the object (which in this case is the original Geometry node that you sent to Mari) and the channel name. The channels are flattened for export.
5. Once the texture has been written to disk, Nuke creates a Read node. Since this is a UV texture and not a projection texture, you need to connect it directly to an ApplyMaterial node instead of the Project3D node.
6. Select **Nuke** > **Send** > **UV Texture(s) (Current Channel)** to export a set of UV textures from the current channel in the **Channels** palette to the Mari data directory. The filename is based on the object (which in this case is the original Geometry node that you sent to Mari) and the channel name.
7. Once the texture has been written to disk, Nuke creates a Read node. Since this is a UV texture and not a projection texture, you need to connect it directly to an ApplyMaterial node instead of the Project3D node.
8. Select **Nuke** > **Send** > **Current View** to create a projector of the current view in Mari (Perspective View only) and send it to Nuke along with the unprojected texture. In addition to the Read node, an additional Camera node is created that represents Mari's canvas at the time of the export.
9. Select **Nuke** > **Send** > **Current Projector** to send the current projector in Mari's **Projectors** palette to Nuke along with the unprojected texture. In addition to the Read node, an additional Camera node is created that represents Mari's canvas at the time of the export.

Warning: If you create a projector in Mari and intend to send it to Nuke to project onto a ReadGeo node, it is advisable that you create your projector while in **Perspective** view. If you choose to create a projector while in **Ortho** view, the projection may not look the way you expected.

10. Select **Nuke** > **Send** > **All Projectors** to send all projectors in the **Projectors** palette to Nuke along with their unprojected textures. In addition to the Read node, a new Camera node is created that represents Mari's canvas at the time of the export.



Note: You can also opt to export the projection instead of sending it to Nuke using the Nuke<>Mari bridge.

To export:
1. In Mari, select any of the seven options from the **Mari** > **Export** menu.
2. Set a directory location in the **Export Nuke Projectors** dialog.
3. In Nuke, select **Mari** > **Import Projection** and select the file in the **Load Mari Projection** dialog to import the projection into Nuke.
This lets you send files between the two applications without a socket connection.

Note: When exporting files to disk in Nuke, an **.nmb** extension is assigned (for Nuke<>Mari Bridge). When exporting from Mari, an **.mnb** extension is used (for Mari<>Nuke Bridge).

Sending a Sequence Projection Setup

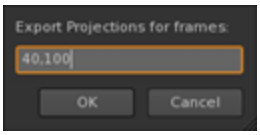
In addition to sending a single projection setup, Nuke can also export multiple snapshots of an animated projection to Mari. This can be used for painting clean plates by combining multiple frames of an image sequence and projecting them through the match-move camera at their respective frames.

This workflow is the same as **Sending a Projection Setup**, with the exception of a **Frame Range** dialog that asks for the frames that the snapshots should be taken from.

Note: When planning to return a UV texture from Mari for a Geometry node, you should uncheck the Geometry node's **Image Aspect** checkbox at this stage. If you don't, the image size changes when plugging in the UV map later on, causing a line-up error.

To send a Nuke Sequence Projection Setup to Mari, follow these steps:

1. Select the Project3D node and the Geometry node that the projection is applied to.
2. Click **Mari** > **Send** > **Sequence Projections**.
3. Enter the frame numbers in the **Frame Range** dialog.



4. Like before, if there is no project open in Mari, the script creates a new project and prompts you to specify a channel for your projections. You are prompted to enter a **Channel name**, set the **Mapping scheme**, provide a **Channel resolution** and finally, select the **Projector resolution** for the new channel.
5. Once your projections have been imported, Mari creates separate projectors for each frame in the sequence.
6. Proceed with the necessary texture work and, following the steps outlined in Sending or Exporting Projections to Nuke, send the data back to Nuke.

Sending Image Nodes

You can also send any number of Read nodes directly from Nuke to Mari. To do this, select the Image nodes you want to send in Nuke, click **Mari** > **Send** > **Images** and select from one of the following options:

- **All Read Nodes** - To send all Read nodes in the Node Graph.
- **Selected Read Nodes** - To send only the selected Read nodes in the Node Graph.
- **Selected Nodes** - To send all Read nodes in the Node Graph. These appear in the **Image Manager** in Mari.

Sending a Lookup Table (LUT)

To ensure that images in Mari look the same as they do in Nuke, Nuke's current Viewer LUT is approximated in a 10-bit 1D LUT and sent to Mari as a viewing LUT. If you change the **viewerProcess** menu in Nuke's Viewer, you can update Mari's viewer LUT accordingly by clicking **Mari** > **Send** > **LUT**.

Using the Nuke<>Mari Bridge over a Network

To use the Nuke<>Mari workflow between different machines on the same network, ensure that both machines have access to the folder where the shared data is stored. This means that the Mari data directory, under **Project Settings > Mari > advanced > Mari data dir** in Nuke, should be set to a shared drive that both machines have access to.

Both computers must be able to access the folder with the same path - for example, two Linux machines might access a shared folder called **/server/netshare/Temp**. In this case, no further changes need to be made and the workflow should function correctly.

If the two machines access the shared folder through different paths - for example, **/server/netfolder/Temp** on a Linux machine, **\\server\netfolder\Temp** on a Windows machine, or **/server/netfolder/Temp** - then some adjustments must be made. The simplest way to account for these differences is to set up symbolic links between directories so the paths match. Your system administrator can help you with this.

File paths sent from Nuke on Windows to Mari on Linux may still need some modification, as the Windows paths generally start with a drive letter such as **C:**, which Mari on a different OS does not understand. An additional Python function, **mari_bridge.filenameFilter**, is available to help translate paths when sending files from Nuke to Mari. You can override this function in Nuke to modify paths as required.

Software API Overview

This section is aimed at developers and provides you with the basic information you need to get started using the Python, Shader, and C APIs in Mari. To access the HTML documentation, in Mari, navigate to **Help** > **SDK**. Alternatively, follow this link for online access.

Using Python in Mari

Learn how to use Python in Mari using the Python Console Palette.

Using Mari's C API

Find out about Mari's C API, an SDK that you can use to develop plug-ins for functionality where Python scripting is not the best solution.

Custom Shaders

Learn about Mari's custom shader API for writing and registering your own shaders.

Using Python in Mari

Mari uses Python 3.11.11 and Unicode Character Set (UCS) 4. Most of Mari's functions are implemented through the use of manager objects, such as **mari.projects**, **mari.menus**, and so on.

Accessing Mari's Python Documentation

1. To view HTML documentation on Mari's Python API, select **Python** > **Documentation** in Mari. Alternatively, follow this link for online access.
2. To view example Python scripts, go to the **Media/Scripts/examples** sub-directory of the Mari application directory and open any of the **.py** files there in a text editor. You can also see the results of these example scripts in the **Python** menu in Mari.

> **Note:** Executing the Python **help()** command in the **Python Console** launches the HTML documentation in a web browser.

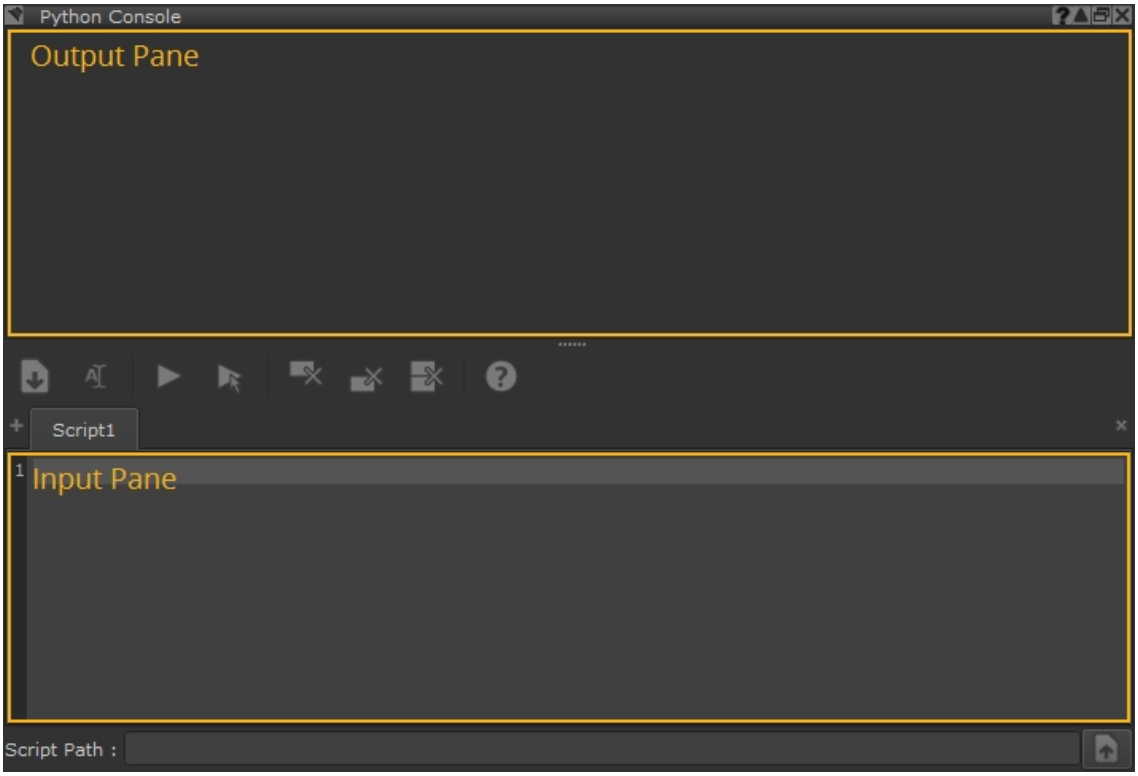
> **Tip:** To read more about Python, review its documentation, or interact with other Python users, you can visit the Python programming language official website at http://www.python.org/.

Mari's Python Console Palette

Mari's **Python Console** palette behaves in a very similar way to a standard Python console. To open the **Python Console** in Mari you can:

- click **Python Console** in the right-hand palette menu,
- navigate to **View** > **Palettes** > **Python Console**,
- navigate to **Python** > **Show Console** in the menu bar or press **Ctrl**/**Cmd**+**Alt**+**P**, or
- right-click in the toolbar area on top of the Mari workspace and select **Python Console**.

The **Python Console** is divided into three parts, the output pane, the toolbar, and the input pane. Use the input pane to enter and evaluate Python statements and the output appears in the output pane.



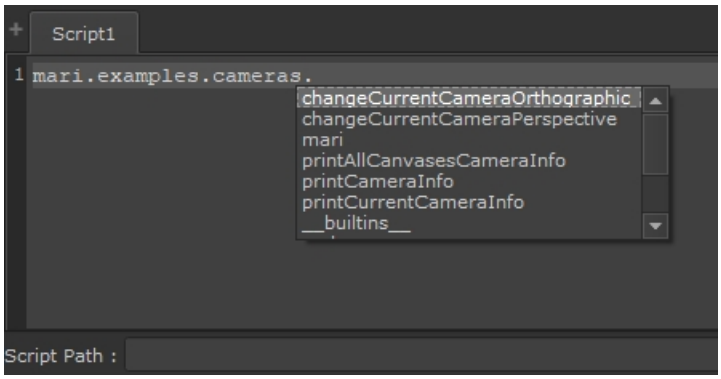
The toolbar manages loading, evaluation, and housekeeping.

	Save As	Click to save the contents of the current script tab to disk. A browser is displayed to locate the required file path.
	Rename Tab	Click to rename the current script tab to something more descriptive. The tab name is the default name given to the save file when you click Save As.
	Evaluate Input Pane	Click to evaluate the contents of the current script tab. You can also use the **Ctrl**/**Cmd**+**Enter** keyboard shortcut. The result is displayed in the output pane.
	Evaluate Selected String	Click to evaluate only the highlighted strings in the current script tab. You can also use the **Ctrl**/**Cmd**+**Shift**+**Enter** keyboard shortcut. The result is displayed in the output pane.
	Clear Output Pane	Click to clear the contents of the output pane at the top of the palette.
	Clear Input Pane	Click to clear the contents of the input pane at the bottom of the palette.
	Clear All	Click to clear the contents of both the output and input panes at once.
	Show Python Help	Click to display a browser containing the Mari Python API Help.

Evaluating Statements

1. Type statements into the input pane. As you type, Mari provides auto-complete suggestions, if any are available. For example, entering:
   `mari.example.cameras.`
   Displays available functions that follow the statement:

> **Tip:** You can temporarily turn off Mari's auto-complete function in the Python Console by pressing **Esc** when the auto-complete list appears.



2. Use the arrow keys to select the required entry and press **Enter** to append it to your statement. Copy and paste functions work the same as in other text editors, but you can also right-click on the input pane and select the desired function.

> **Note:** Copying and pasting statements from another source, such as an email, into the **Python Console** can cause syntax errors. This may be because of mark-up or encoding in the source material. To fix the problem, re-enter the statement manually.

3. Mari includes a syntax highlighting scheme in the input pane, so that code can be more easily identified at a glance. Text in red represents strings, green represents comments, and various Python keywords are represented by yellow, magenta, and blue.

4. To execute the statement, click the Evaluate button or press **Ctrl/Cmd** + **Return**.
Your statement appears in the output pane, preceded by **>>>** (or **...** if your statement spans multiple lines).

If you enter an invalid statement, Mari produces an error in the output pane.

> **Tip:** To execute only part of the code in the input pane, highlight the code you want to execute and click the Evaluate Selected button or press **Ctrl/Cmd** +**Shift**+**Enter**.

5. If you want to repeat a statement, you can step backwards or forwards through the history of your script by pressing **Alt**+**Up** or **Alt**+**Down** on Windows and Linux or **Cmd**+**Ctrl**+**Up** or **Cmd**+**Ctrl**+**Down** on macOS.
6. If you want to clear the input and output panes, click the Clear Input, Clear Output, or Clear All button.

> **Tip:** You can increase or decrease the font size in the **Python Console** by holding down **Ctrl/Cmd** and pressing **+** or **-**.

7. If you want to open the **Mari Python API Help** without going to the **Python** > **Documentation** menu, click the **Help** button.
   You can move this window around and keep typing in the input pane while it's open. This window cannot be docked like a palette, but you can change focus from the window while it's still active, unlike most of Mari's dialogs.

> **Note:** If Mari detects an auto-complete suggestion while the **Mari Python API Help** window is open, the suggestion appears in the help automatically.

Saving Scripts

To save a Python script to disk:

1. Select the script tab you want to save to disk.
   You can work on as many separate script tabs as required and the contents of each tab is loaded every time you launch Mari.
2. Click the Save button to display the **Save Python Script** dialog.
3. Navigate to the save directory and enter a **File name**.

> **Tip:** You can use the Rename button to give script tabs a name before saving to disk, which completes the **File name** automatically.

4. Click **Save** to write the file to disk.

Loading Scripts

To load a script through the **Script Path** text field:

1. Manually enter the script's location, or
2. Click on to launch the **Python Script Path** dialog box and browse to the location of the script.
3. To execute the statement, click the Evaluate button.

Custom scripts load from the **import** directory prior to the scripts being run from the main folder. Mari uses particular order to load modules to ensure the proper, full functionality of the program when using Python scripts.

If existing start-up scripts do not work correctly due to the load module order, you can revert back to the old behavior by setting the environment variable **MARI_OLD_PYTHON_INIT** to any non-empty string other than 0.

> **Note:** Mari looks for metadata that is over 250 MB in size and discards anything over this size. This check is performed on project load, and is intended to strip corrupt and problematic data. This affects metadata added using the Python API.

Running Scripts on Start-up

If you would like your script to be automatically executed at start-up, do the following:

1. Enter your script in any text editor.
2. Save the text file with the extension **.py** (for example, **my_module.py**) to:

- the **~/Mari/Scripts** sub-directory of the Mari application directory, or
- your own directory by setting the environment variable **MARI_SCRIPT_PATH** to a custom folder.

If a script called **init.py** exists, it is run first. If you need scripts to run in a specified order, you should put them in a separate folder and use a Python **import** statement in **init.py**.

You can also specify multiple directories to run scripts on start-up, using a path list separated by ':' or ';' as per the standard for your operating system.

- **On Linux:** /home/username/dir1:/home/username/dir2
- **On Windows:** C:\Users\username\dir1;C:\Users\username\dir2
- **On Mac:** /home/users/username/dir1;/home/users/username/dir2

3. The **import** directory, contained in the following folders of your custom scripts path:
   - **On Linux:** /Mari/Scripts
   - **On Windows:** Documents\Mari\Scripts
   - **On Mac:** /Documents/Mari/Scripts

   Scripts saved to this directory are imported rather than run, which prevents unnecessary pollution of the global namespace. If you save your custom script to the **import** directory, it is automatically imported before the scripts in the main folder are run.

> **Note:** Python files and sub-folders that you wish to import without running in the **import** directory must be **__init__.py** files.

Terminal Mode

Mari provides a terminal mode, which allows users to access features through a Python shell on the command line. This is similar to the Python console palette in the Mari GUI, or to a standard Python shell. Terminal mode can be used to perform administrative actions such as archiving projects, exporting textures, and other tasks that do not require graphical interaction from the user.

Terminal mode is not a true "headless mode" because it still requires graphics functionality to run. Mari still initializes its GUI when starting up terminal mode - it just won't display it. If you try to run a Mari interactive and terminal session at the same time, an error displays on the command line:

```
MriMainApp.cpp:798 ] : [ !! ] Mari already running somewhere else
```

You can supply the file name of a Python script to run on the command line, if desired. If supplied, Mari runs the specified script before providing the Python shell input prompt. It is also possible to start in "execute mode", which is the same as terminal mode but exists after running the provided script.

To start terminal mode from a shell, use the following commands:

On Linux

```
cd /path/to/mari
# Normal terminal mode
./mari -t
# Terminal mode - run the script, then show a Python input prompt
./mari -t /path/to/script_name.py
# Execute mode - run the script, then exit
./mari -x /path/to/script_name.py
```

On Windows

```
cd \path\to\mari
# Normal terminal mode
Mari8.0v1.exe -t
# Terminal mode - run the script, then show a Python input prompt
Mari8.0v1.exe -t \path\to\script_name.py
# Execute mode - run the script, then exit
Mari8.0v1.exe -x \path\to\script_name.py
```

On Mac

```
cd /path/to/Mari
# Normal terminal mode
./Mari8.0v1 -t
# Terminal mode - run the script, then show a Python input prompt
./Mari8.0v1 -t /path/to/script_name.py
# Execute mode - run the script, then exit
./Mari8.0v1 -x /path/to/script_name.py
```

Tip: You can also add the **-v** argument to set the command line output to verbose mode so you can troubleshoot more easily.

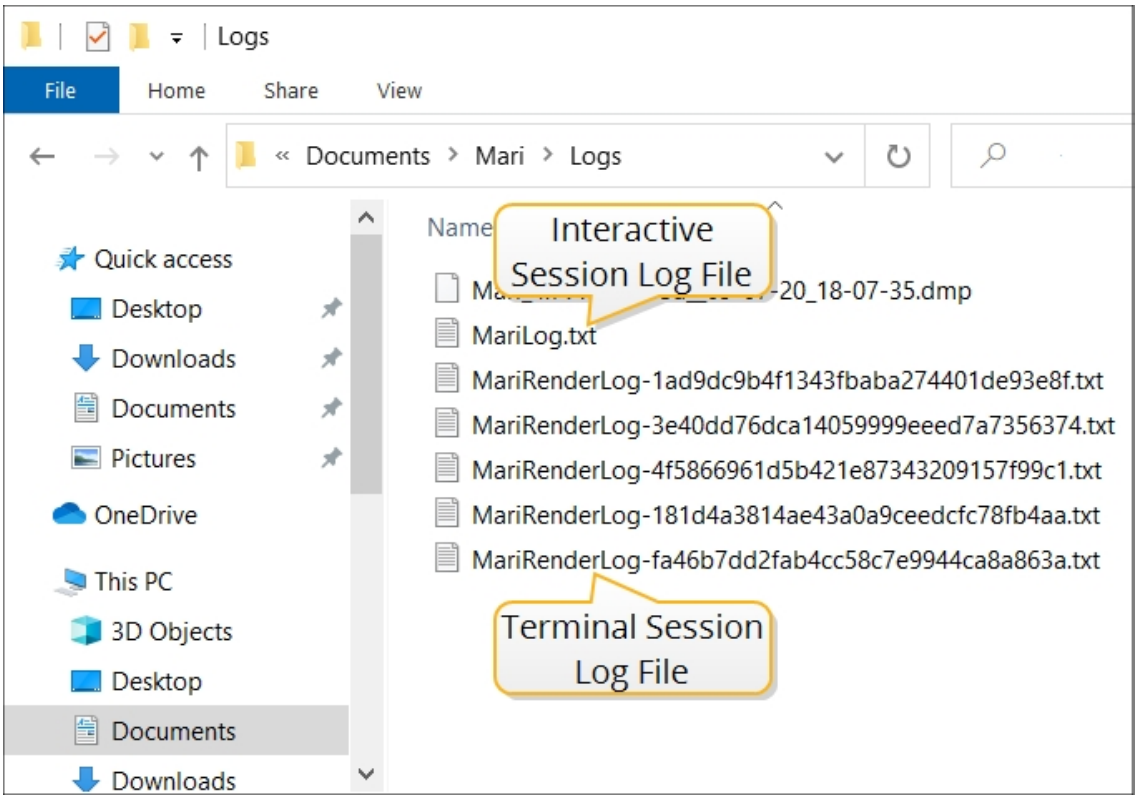
Operational logs for terminal and execute sessions are written to the same location as standard Mari logs, but a unique hash is added to the log file for identification. To print the hash for a terminal or execute session, use the following command:

```
mari.app.logFileName()
```

Log files are written to the following directories by operating system:

- Linux and Mac: `/home/<username>/Mari/Logs`
- Windows: `C:\Users\<username>\Documents\Mari\Logs`

An example of a Windows log directory containing log files is shown in the image.



Using PySide2 in Mari

Mari ships PySide2 together with its Python scripting engine. PySide2 is a library of Python bindings for Qt, similar to PyQt. For more information on PySide2, visit http://qt-project.org/wiki/PySide.

Using Mari's C API

Mari provides an SDK for developing binary plug-ins for functionality where Python scripting is not the best solution.

Introduction

The C API allows users to develop plug-ins to read and write custom image formats, or to read custom geometry formats. In these cases, passing data between the application and plug-ins can be done more efficiently by passing buffers of binary data rather than using intermediate Python types.

The choice to use C instead of C++ for the API provides better compatibility and simpler compilation for users, and as always, can still be used in plug-ins written in C++.

Reviewing Mari's C API Documentation

For full details of the C API, including example plug-ins, please see the C API documentation under **Help** > **SDK** > **C API** > **Documentation**, or through the installation directory in the SDK folder.

Custom Shaders

Modular Shaders

Versions of Mari prior to the 2.0v1 release provided Python API hooks to register custom "modular shaders." However, the modular shader system has since been deprecated in favor of a more flexible node-graph-based system.

Custom Shaders and Layers

A custom shader API for Mari's node-graph-based system exists to give you the ability to write and register your own shaders. Mari renders a scene in the canvas using its extensible shader system, based on a node graph. Internally, within Mari, shaders and layers are in the form of nodes. The custom shaders and layers can then be added by supplying XML files to Mari, defining custom shaders and layers in the form of nodes. Once the node definition has been written in an **.xml** file, it can be registered to Mari using the Python API.

The Custom Shader API that is built into Mari can be accessed by navigating to **Help** > **SDK** > **Custom Shader API**. Additionally, you can read more on the Python and C API documentation by navigating to **Python** > **Documentation** and **Help** > **SDK** > **C API** > **Documentation** on the Mari menu bar. Alternatively, follow this link for online access.

If you have any questions or difficulties with writing or registering custom shaders, please contact Customer Support directly through the Support Portal at the following address: https://support.foundry.com.

Extending Mari

You can add your own custom attributes to objects in Mari, as well as edit configuration files. In particular, you can specify sets of:

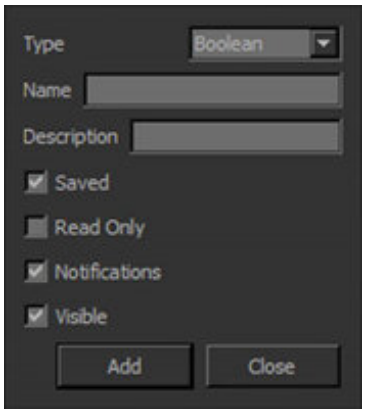
- keyboard shortcuts - Using a shortcut configuration file has the same effect as changing each shortcut individually in the **Manage Keyboard Shortcuts** dialog box, but lets you change all of them, as a single, consistent set. You can then also share a set of shortcuts with other users.
- channel presets - Using a channel preset configuration file, you can specify categories of preset channels (for example, for different types of assets), to display in the **Channel Presets** dialog box when you create channels for a project in bulk.
- channel templates - Using a channel template configuration file, you can specify the filename formats to use in the **Channel Presets** dialog box. These control the filenames that Mari looks for when importing textures into new channels.

You can also configure Mari using environment variables.

Adding Custom Attributes to Objects

1. Start Mari and open your project.
2. Open a palette where you want to add custom attributes (for example, the **Channels**, **Objects**, or **Projectors** palette).
3. Select the object you want to add custom attributes to and click the button. You may need to scroll right in the palette to see this button.

   The **Add User Attribute** dialog displays.



4. Set the following:
   - **Type** - whether the attribute's value is an on/off setting (**Boolean**), a date (**Date**), a floating point number (**Float**), an integer number (**Integer**), or a text string (**String**).

   - **Name** - the name of the attribute. Unless you uncheck **Visible** below, this name appears under **User Attributes** in the palette (in the below example, the name is **MyAttribute**). Note that the name should only include alphanumeric characters and cannot include spaces.
   - **Description** - an optional description of the attribute. This appears in a tooltip if you hover the mouse over the attribute in the palette.
   - **Saved** - if this is on, the attribute persists when you quit and relaunch Mari. If this is off, the attribute is only added to the current session of Mari.
   - **Read Only** - if this is on, users cannot modify the attribute's value. If this is off, the value is shown in the palette where users can adjust it.
   - **Visible** - if this is on, the attribute appears in the palette. If this is off, you cannot see the attribute.

> **Note:** The above settings cannot be edited later. If you want to change them, you need to remove the attribute and create a new one.

5. Click **Add**.
6. If necessary, edit the attribute's value in the palette.

Removing Custom Attributes from Objects

1. Start Mari and open your project.
2. Open a palette where you want to remove a custom attribute from (for example, the **Channels**, **Objects**, or
   Projectors palette).
3. To remove the attribute, click the button next to it in the palette.
   Mari prompts you to confirm that you want to remove the attribute.
4. Click **Remove**.

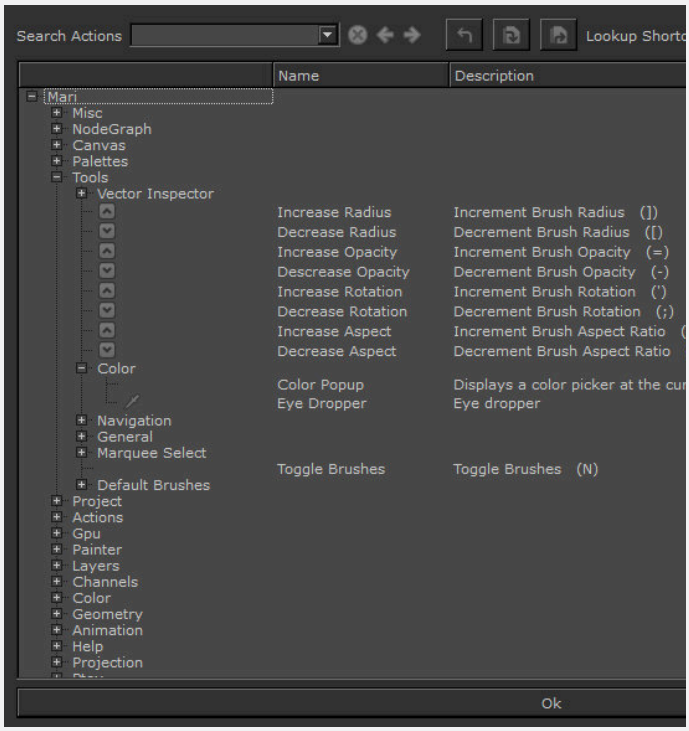
Reviewing the Custom Shortcut File Format

1. Read through the following information about the required format of a Mari custom shortcuts file.
2. Study the example default shortcuts file.

Text File: <action>=<key(s)>

A default (**DefaultShortcuts.conf**) or custom (**Shortcuts.conf**) shortcuts configuration file is a text file that starts with **[Shortcuts]** and then lists Mari actions and their shortcut keys, one per line. The actions are the same as what you find in the **Manage Keyboard Shortcuts** dialog box in Mari, with the hierarchies spelled out, separated by a backslash (\), and with "%20" for spaces. After each action is an equal sign (=) and then the key or key combination. For key combinations, you use one or more "modifier" keys, like **Ctrl**/**Cmd** or **Alt** with a plus sign (**+**) between each.

For example, below are some shortcuts in the **Manage Keyboard Shortcuts** dialog box alongside their listings in the default shortcuts configuration file.

GUI dialog box entries	Configuration file listings
	[Shortcuts] ... Mari\Tools\Increase%20Radius=']" Mari\Tools\Decrease%20Radius="[" Mari\Tools\Increase%20Opacity="=" Mari\Tools\Descrease%20Opacity="-" Mari\Tools\Increase%20Rotation=' Mari\Tools\Decrease%20Rotation=";" Mari\Tools\Increase%20Aspect=0 \Mari\Tools\Decrease%20Aspect=9 ... Mari\Tools\Color\Eye%20Dropper=C Mari\Tools\Color\Color%20Popup=J ... Mari\Tools\Toggle%20Brushes=N

Tip: Case is not important. The lines **Mari\Tools\General\Transform%20Paint=M** and **Mari\Tools\General\Transform%20Paint=m** have the same meaning. If you want to set a shortcut key to only the uppercase version of a character, include **Shift** in the key sequence. For example:
Mari\Tools\General\Transform%20Paint=Shift+M only invokes the **TransformPaint** tool when you hold down both **Shift** and **M** at the same time.

Using Modifier Keys

Some modifier keys can only be used as part of key combinations - you cannot assign a shortcut to the **Ctrl/Cmd** key by itself, for example. This table lists the special keys available, the text strings to use for them in **Shortcuts.conf**, and whether the key can be used on its own.

Key	Text string in file	Can be used by itself?
Ctrl/Cmd	Ctrl	No
Shift	Shift	No
Alt (not **Alt Gr**)	Alt	No
Escape	Esc	Yes
Caps Lock	CapsLock	Yes
Space	Space	Yes
F1 – F12	F1 – F12	Yes
Scroll Lock	ScrollLock	Yes
Pause	Pause	Yes
Insert	Ins	Yes
Left Arrow	Left	Yes
Right Arrow	Right	Yes
Up Arrow	Up	Yes
Down Arrow	Down	Yes
Home	Home	Yes
End	End	Yes
Page Up	PgUp	Yes
Page Down	PgDown	Yes

Key	Text string in file	Can be used by itself?
Number Lock	NumLock	Yes
Backslash	\\	Yes

Tip: You can use **** (backslash) as a shortcut, but as it is used as a directory separator in the action part of the line, you must use a double backslash to make it clear that you are actually assigning the key rather than indicating a directory.

Example: DefaultShortcuts.conf

The default shortcuts file, **DefaultShortcuts.conf**, is included in the following sub-directory of the Mari application directory:

- **Media/Settings** (on Linux),
- **Bundle\Media\Settings** (on Windows), or
- **Media/Settings** (on Mac).

We recommend you create your custom shortcuts by copying this file and editing it, rather than starting from scratch.

Note: If you are using a Mac, you need to replace all references of **Ctrl** in the table below to **Cmd** for it to work correctly in Mari.

```
[Shortcuts]
Mari\Tools\General\Select=S
Mari\Tools\General\Scrub=Shift+?
Mari\Tools\General\Mouse%20Gestures=\\
Mari\Tools\General\Hide%20Widgets=/
Mari\Tools\General\Transform%20Paint=M
Mari\Tools\General\Zoom%20Paint%20Buffer=Z
Mari\Tools\General\Vector%20Inspector=V
Mari\Tools\Vector Inspector\Enable%20Vector%20Inspector=Shift+I
Mari\Tools\Default%20Brushes\Paint=P
Mari\Tools\Default%20Brushes\Paint%20Through=U
```

```
Mari\Tools\Default%20Brushes\Paint%20Buffer%20Eraser=E
Mari\Canvas\Bake=B
Mari\Canvas\Bake%20and%20Clear=Ctrl+Shift+B
Mari\Canvas\Display\Toggle%20Wireframe=Shift+W
Mari\Canvas\Camera\Focus=A
Mari\Canvas\Camera\Jump%20To=F
Mari\Tools\Color\Eye%20Dropper=C
Mari\Tools\Color\Color%20Popup=J
Mari\Tools\General\Shelf%20Popup=K
Mari\Tools\General\Channel%20Popup=I
Mari\Tools\General\Image%20Popup=L
Mari\Tools\General\Pie%20Selector=F9
Mari\Animation\Step%20Forwards=Ctrl+Right
Mari\Animation\Step%20Backwards=Ctrl+Left
Mari\Animation\Add%20Keyframe=Ctrl+K
Mari\Animation\Rewind%20Animation=Ctrl+Down
Mari\Animation\Fastforward%20Animation=Ctrl+Up
Mari\Canvas\Camera\Camera%20Left=1
Mari\Canvas\Camera\Camera%20Right=2
Mari\Canvas\Camera\Camera%20Top=3
Mari\Canvas\Camera\Camera%20Bottom=4
Mari\Canvas\Camera\Camera%20Front=5
Mari\Canvas\Camera\Camera%20Rear=6
Mari\Onscreen%20Controls\Adjust%20Radius=R
Mari\Onscreen%20Controls\Adjust%20Rotation=W
Mari\Onscreen%20Controls\Adjust%20Squish=Q
Mari\Onscreen%20Controls\Adjust%20Opacity=O
Mari\Actions\Quit=Ctrl+Q
Mari\Actions\Next%20Tab=Ctrl+Tab
Mari\Actions\Previous%20Tab=Ctrl+Shift+Backtab
Mari\Actions\Undo=Ctrl+Z
Mari\Actions\Redo=Ctrl+Y
Mari\Actions\Previous%20View="Ctrl+,"
Mari\Actions\Next%20View=Ctrl+.
Mari\Actions\Toggle%20Palettes%20Visibility=Home
Mari\Project\Save=Ctrl+S
Mari\Project\New=Ctrl+N
Mari\Project\Close=Ctrl+W
Mari\Project\Lighting\Flat=F1
Mari\Project\Lighting\Basic=F2
Mari\Project\Lighting\Full=F3
Mari\Painter\Toggle Cursor Type=F5
Mari\Palettes\Tool Properties=F7
Mari\Canvas\Camera\View%20All=Shift+A
Mari\Painter\Swap%20foreground%20and%20background=X
Mari\Project\Project%20Explorer\Quick%20Paste=Ctrl+V
Mari\Project\Project%20Explorer\Quick%20Copy=Ctrl+C
```

```
Mari\Geometry\Selection\Select Invert=Ctrl+I
Mari\Geometry\Selection\Select All=Ctrl+A
Mari\Geometry\Selection\Select None=Ctrl+E
Mari\Geometry\Selection\Hide%20Selected=H
Mari\Geometry\Selection\Hide%20Unselected=Shift+H
Mari\Geometry\Selection\Show%20All=Ctrl+Shift+H
Mari\Painter\Toggle%20Edge%20Mask=G
Mari\Painter\Toggle%20Show%20Painting=.
Mari\Projection\Toggle%20Mask%20Preview%20Enabled=","
Mari\Projection\Toggle%20Masking=Ctrl+M
Mari\Channels\Toggle%20Channels=T
Mari\Channels\Next%20Channel=
Mari\Channels\Previous%20Channel=
Mari\Tools\Toggle%20Brushes=N
Mari\Tools\Decrease%20Aspect=9
Mari\Tools\Decrease%20Radius=-
Mari\Tools\Decrease%20Rotation=";"
Mari\Tools\Descrease%20Opacity=[
Mari\Tools\Increase%20Aspect=0
Mari\Tools\Increase%20Opacity=]
Mari\Tools\Increase%20Radius="="
Mari\Tools\Increase%20Rotation='
Mari\Painter\Set%20Colors%20to%20Black%20And%20White=D
Mari\Canvas\Camera\Ortho%20Camera=F10
Mari\Canvas\Camera\Perspective%20Camera=F11
Mari\Canvas\Camera\UV%20Camera=F12
Mari\Tools\Marquee Select\Clear%20Marquee%20Selection=Ctrl+D
Mari\Canvas\Take%20Screenshot=Ctrl+Shift+Print
Mari\Painter\Clear%20Painting=Ctrl+Shift+C
Mari\Tools\Navigation\Roll=Ctrl+R+LeftBtn
Mari\Tools\Navigation\Pan=Alt+Shift+LeftBtn, Alt+MidBtn
Mari\Tools\Navigation\Orbit=Alt+LeftBtn
Mari\Tools\Navigation\Zoom=Alt+Ctrl+LeftBtn, Alt+RightBtn
Mari\Painter\Reset%20Blend%20Mode=Ctrl+0
Mari\Painter\Next%20Blend%20Mode=Shift+)
Mari\Painter\Previous%20Blend%20Mode=Shift+(
Mari\Painter\Last%20Blend%20Mode=Ctrl+9
Mari\Painter\Clear%20Mode=Shift+Backspace
Mari\Layers\Quick%20Find=Tab
[Node Graph/Shortcuts]
Mari\NodeGraph\Autoplace=L
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer=1
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer2=2
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer3=3
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer4=4
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer5=5
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer6=6
```

```
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer7=7
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer8=8
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer9=9
Mari\NodeGraph\Connect%20Selected%20Node%20to%20Viewer0=0
Mari\NodeGraph\Move%20Viewer%20to%20Cursor=Shift+!
Mari\NodeGraph\Delete=Delete
Mari\NodeGraph\Frame%20Nodes=A
Mari\NodeGraph\Focus%20Selection=F
Mari\NodeGraph\Disable=D
Mari\NodeGraph\Bypass=Ctrl+D
Mari\NodeGraph\Change%20Current%20Channel=W
Mari\NodeGraph\Quick%20Create=Tab
Mari\NodeGraph\Select%20All=Ctrl+A
Mari\NodeGraph\Cut=Ctrl+X
Mari\NodeGraph\Copy=Ctrl+C
Mari\NodeGraph\Paste=Ctrl+V
Mari\NodeGraph\Search%20Node=J
Mari\NodeGraph\Create%20Paint%20Node=P
Mari\NodeGraph\Create\Misc\Group=Ctrl+G
Mari\NodeGraph\Create\Misc\Dot=.
Mari\NodeGraph\Info%20Viewer=I
Mari\NodeGraph\Create\Layer\Merge=M
Mari\NodeGraph\Swap%20Inputs=Shift+X
Mari\NodeGraph\Extract=Ctrl+Shift+X
Mari\NodeGraph\Rename%20Node=N
Mari\NodeGraph\Toggle%20Port%20List=Shift+S
```

Defining Custom Shortcuts

To create a Mari shortcut customization file:

1. Copy the Mari default shortcuts file, **DefaultShortcuts.conf**, from the following sub-directory of the Mari application directory:
   - **Media/Settings** (on Linux),
   - **Bundle\Media\Settings** (on Windows), or
   - **Media/Settings** (on Mac).
2. Rename the file as a custom shortcuts file: **Shortcuts.conf**.
3. Edit the file, replacing the defaults with your custom shortcuts.
4. Save the file in the following sub-directory of your home directory:
   - **.config/TheFoundry** (on Linux),
   - **.mari\TheFoundry** (on Windows), or

- **.config/TheFoundry** (on Mac).

Tip: If you set an illegal key combination in the **Shortcuts.conf** file, Mari ignores the line. For example, the following lines would be ignored:

Mari\Tools\General\Select=Tab (you cannot use tab as part of a shortcut)
Mari\Tools\General\Transform%20Paint=Altw (needs a **+** between the two characters).

If the same function key is assigned to more than one function, Mari ignores both settings and uses the defaults. If a function is assigned to more than one key combination, Mari uses the defaults.

Reviewing the Custom Channel Preset File Format

1. Read through the following information about the required format of a Mari custom presets file.
2. Study the example default presets file.

Note: To know more about the channel presets, see About Custom Channel Presets.

XML File: Header, Details, Closing Blocks

Channel preset files are **XML** files with an **.mtd** extension. Each file can hold as many channels as you want, divided into categories. The files can have any name you want, as long as they have the extension **.mtd**. Mari picks up the settings from all **.mtd** files present, so you can use multiple files. For example, you could use one local **.mtd** file for a single artist's specific channel presets and one on a network drive for multiple artists' presets.

Each file must have a **header** block, one **details** block per channel, and a **closing** block. Below are examples of each type of block, with values you need to change **bolded** and explained in a table below.

Header Block

This must be the first thing in the file.

```
<!DOCTYPE MariML>

<Mari Version="2" >

<MriSerializableObject Type="MriSerializableObject" >

<objectName Type="QString" ></objectName>

<selected Type="bool" >false</selected>

<m_Channels Size="1" Type="MriChannelInfo"
ContainerType="QVector">
```

You must set the following values:

Setting	Meaning	Example value
m_Channels Size	Number of channels in this preset file.	1

Details Blocks

You need one details block for each channel. Separate the blocks with a blank line.

```
<MriChannelInfo Type="MriChannelInfo" >

<objectName Type="QString" ></objectName>

<selected Type="bool" >false</selected>

<channelName Type="QString" >My Custom Channel</channelName>

<fileFormat Type="QString" ></fileFormat>

<path Type="QString" ></path>
```

```
<scalar Type="bool" >false</scalar>

<width Type="int" >4096</width>

<height Type="int" >4096</height>

<channels Type="QString" >RGB</channels>

<depth Type="QString" >Byte</depth>

<color Type="MriColor" >(0.5,0.5,0.5,1)</color>

<category Type="QString" >Custom</category>

</MriChannelInfo>
```

You must set the following values:

Setting	Meaning	Example value
selected	When set to **true**, indicates that the **Create** checkbox, in the **New Project** dialog, should be enabled by default.	true
channelName	Name for the channel to create.	My Custom Channel
scalar	Whether the channel is color or scalar. This is a Boolean; true means a scalar channel, false means color.	false
width	Width of patches in the channel. This must match the heightType. **Tip:** If the width and height are greater than a project's maximum resolution, Mari creates the channel at the project's maximum resolution.	4096
height	Height of patches in the channel. This must match the widthType.	4096

Setting	Meaning	Example value
	Tip: If the width and height are greater than a project's maximum resolution, Mari creates the channel at the project's maximum resolution.	
channels	The color channels for the new channel. Must be one of: • RGBA • RGB • 1 • 2	RGB
depth	Color depth for the channel. Must be one of: • Byte • Half • Float	Byte
color	Default color for the new channel. This is the color that Mari fills the channel with when created. This must be entered as a floating value, with values from 0 to 1 for the Red, Green, Blue and Alpha.	0.5,0.5,0.5,1
category	The category for the channel. When the artist selects this category in the **Channel Presets** dialog, this channel appears in the list.	Custom
fileSpace	Allows you to set whether the channel or image should be exported as a Normal file (in the way Mari stores them), as a Vector file (-1.0 to +1.0 range), or as a Vector (flipped Y) file, which is the same as the	Vector

Setting	Meaning	Example value
	Vector file in terms of range, but inverts the vectors in the Y direction.	
colorSpaceConfig	Specifies the default advanced colorspace options, including the Automatic values for each parameter.	See colorSpaceConfig Parameter

Closing Block

This must be the last thing in the file.

```
</m_Channels>

</MriSerializableObject>

</Mari>
```

Example: DefaultChannels.mtd

The default channel presets file is included in Mari in the **../Bundle/Media/Settings** directory (on Windows) (**DefaultChannels.mtd**). We recommend you create your custom channels file by copying this and editing it, rather than starting from scratch.

Note: The **fileSpace** and **colorSpaceConfig** parameters are not present in the **DefaultChannels.mtd** file, but you can add them to it, see Reviewing the Custom Channel Preset File Format

Note: The example below shows the parameters of the color channel. These parameters apply to all channels.

```
<!DOCTYPE MariML>
<Mari Version="2" >
 <MriSerializableObject Type="MriSerializableObject" >
  <objectName Type="QString" ></objectName>
  <selected Type="bool" >false</selected>
  <m_Channels Type="MriChannelInfo" ContainerType="QVector" >

   <MriChannelInfo Type="MriChannelInfo" >
    <objectName Type="QString" ></objectName>
    <selected Type="bool" >false</selected>
    <channelName Type="QString" >color</channelName>
    <fileFormat Type="QString" ></fileFormat>
    <path Type="QString" ></path>
    <scalar Type="bool" >false</scalar>
    <width Type="int" >4096</width>
    <height Type="int" >4096</height>
    <channels Type="QString" >RGBA</channels>
    <depth Type="QString" >Byte</depth>
    <color Type="MriColor" >(0.5,0.5,0.5,1)</color>
    <category Type="QString" >General</category>
    <fileSpace Type="QString" >Vector</fileSpace>
    <colorSpaceConfig Type="MriColorSpaceConfig"
>0|1|2|3|4|5|6|7|8|9|10</colorSpaceConfig>
   </MriChannelInfo>
```

colorSpaceConfig Parameter

As the **colorSpaceConfig** parameter is a bit more complex than the other parameters, it is explained in detail below:

Note: **Automatic** is the default value.

Key	Description	Data type with an example value
0	colorspace filename	string (for example, **nuke-default** or **Automatic**)
1	data colorspace	string (for example, **rec709** or **Automatic**)
2	output colorspace	string (for example, **rec709** or **Automatic**)

3	working colorspace	string (for example, **rec709** or **Automatic**)
4	automatic colorspace 0	int (for example, **0**, **1**, **2**) Sets the automatic value for the data colorspace, assigning certain colorspace values to set integers.
5	automatic colorspace 1	int (for example, **0**, **1**, **2**) Sets the automatic value for the output colorspace, assigning certain colorspace values to set integers.
6	raw mode enabled	bool (int) Enables (**1**) or disables (**0**) raw mode
7	scalar mode enabled	bool (int) Enables (**1**) or disables (**0**) scalar mode
8	automatic data colorspace	string (for example, **rec709**) Specifies which colorspace is chosen when using **Automatic** value. **Note:** Setting this value overwrites **4 automatic colorspace 0**.
9	automatic output colorspace	string (for example, **rec709**) Specifies which colorspace is chosen when using **Automatic** value. **Note:** Setting this value overwrites **5 automatic colorspace 1**.
10	automatic working colorspace	string (for example, **rec709**) Specifies which colorspace is chosen when using **Automatic** value.

Below is an example of the **colorSpaceConfig** parameter:

```
<colorSpaceConfig Type="MriColorSpaceConfig">nuke-default|rec709|Automatic|Automatic|0|0|0|0|Cineon|Cineon|Cineon</colorSpaceConfig>
```

This creates a channel with **rec709** data colorspace, but **Cineon** working and output colorspaces. **Raw** and **Scalar** are off.

Defining Custom Channel Presets

To create a Mari custom channel presets file:

1. Copy the Mari default channel presets file, **DefaultChannels.mtd**, from the following sub-directory of the Mari application directory:
   - **Media/Settings** (on Linux),
   - **Bundle\Media\Settings** (on Windows), or
   - **Media/Settings** (on Mac).
2. Rename the file to something descriptive, but keep the **.mtd** extension.
3. Edit the file, replacing the key values.
4. Save the file in the following sub-directory of your home directory:
   - **.config/TheFoundry** (on Linux),
   - **.mari\TheFoundry** (on Windows), or
   - **.config/TheFoundry** (on Mac).

   This directory path should then be added to the **Channel Presets** window of the **Mari Preferences > Paths** dialog box.

Reviewing the Custom Channel Template File Format

1. Read through the following information about the required format of a Mari custom presets file.
2. Study the example default channel template file.

Text File: <name>="<format>"

A default (**DefaultTemplates.mtt**) or custom (**<filename>.mtt**) channel templates file is a text file that starts with **[Templates]]** and then lists channel templates and their file formats.

Each line is a channel template. The name is listed on the left, and the format in quotes on the right. The format can include the following variables, and on Linux they are preceded with the dollar sign (**$**) while on Windows they're
preceded with the percent sign (**%**):

Variable	Description
Any of the user's environment variables **except** PATH	These are taken from the shell from which the user opened Mari. Note that PATH here has a specific meaning - it is not the user's standard PATH.
PATH	The project's root path, as set in the **Root Path** field of the **Channel Presets** dialog box. This overrides the normal PATH variable from the user's original shell.
PREFIX	The filename prefix, as set in the **Prefix** field of the **Channel Presets** dialog box.
CHANNEL	Name of the individual channel.
UDIM	Udim number for each individual patch.

Example File: DefaultTemplates.mtt

On Linux:

```
[Templates]

Default="$TEXTURE_HERO_DIR/tif/$SHOT/$CHANNEL/$SHOT_
$CHANNEL_$UDIM.tif"
```

On Windows:

```
[Templates]

Default="%TEXTURE_HERO_
DIR/tif/%SHOT/%CHANNEL/%SHOT_%CHANNEL_&UDIM.tif"
```

Defining Custom Channel Templates

To create a Mari custom channel templates file:

1. Copy the Mari default channel templates file, **DefaultTemplates.mtt**, from the following sub-directory of the Mari application directory:
   - **Media/Settings** (on Linux),
   - **Bundle\Media\Settings** (on Windows), or
   - **Media/Settings** (on Mac).
2. Rename the file to something descriptive, but keep the **.mtt** extension.
3. Edit the file, replacing the key values.
4. Save the file in the following sub-directory of your home directory:
   - **.config/TheFoundry** (on Linux),
   - **.mari\TheFoundry** (on Windows), or
   - **.config/TheFoundry** (on Mac).

   This is also the directory specified on the **Channels** tab of the **Mari Preferences** dialog box.

Environment Variables That Mari Recognizes

Article: Check the knowledge base article about How To Set Environment Variables.

This table lists the environment variables that Mari recognizes.

Environment variable	Description	Example value
FN_DISABLE_LICENSE_DIALOG	By default, if you have installed a temporary license, Mari displays a dialog at start-up alerting you to the number of days remaining. If you want to disable this behavior, you can set this environment variable to 1 to suppress the warning message about imminent license expiration. **Warning:** If you don't have an available Mari license and set this environment variable, Mari won't launch. **Article:** Read more about Temporary License Warning.	
FN_LICENSE_DIALOG_DAYS_LEFT_BEFORE_PROMPT	You can set this environment variable to an integer to specify how many days before expiry the license warning should appear. See FN_DISABLE_LICENSE_DIALOG environment variable above.	
FN_SUBSCRIPTION_LICENSE_DIR	Changes the license directory to an alternate location using the **FN_SUBSCRIPTION_LICENSE_DIR** variable. **Note:** On Windows, there is a known issue with user names containing non-ASCII characters causing licensing to fail. If a licensing error displays, try using the **FN_SUBSCRIPTION_LICENSE_DIR** variable to specify an alternate directory.	
MARI_APPLICATION_THREADCOUNT	Overrides the application thread count setting. When the environment variable is unset, Mari uses all available threads.	**MARI_APPLICATION_THREADCOUNT=2**

Environment variable	Description	Example value
MARI_CACHE	The location(s) of the Mari project files. This can be just one directory or several (to make caching faster). You should separate multiple path entries by a **:** (colon) on Linux and Mac, or a **;** (semi-colon) on Windows. Project directories must be permanent directories and not temporary locations as they contain important project file information. If this variable is not set, Mari prompts you where to store the project files when you first launch it. Note that when you set this environment variable, Mari takes its value and stores it in the **CacheLocations.ini** file (by default found in the **.config/TheFoundry** sub-directory of your home directory on Linux, or the **.mari\TheFoundry** sub-directory on Windows and Mac). Be cautious of changing this. The set of project locations must not be changed after project creation; doing so results in project corruption. For more information please read the chapter on Installation and Launch in the *Getting Started Guide*.	• On Linux: **/user/**<MyPath>**/MariCache** • **On Windows:** **C:\<**MyPath>**\MariCache** • On Mac: **/user/**<MyPath>]**/MariCache**
MARI_COLORSPACE_OCIO_UI_ALLOWLIST	Sets the list of colorspaces artists can select from Mari's colorspace menus. Some OCIO config files contain 100+ different named colorspaces, which can lead to confusion. Limiting the number of options reduces the risk of working in an unsuitable colorspace.	Separate colorspaces with ; (semicolon). For example: MARI_COLORSPACE_OCIO_UI_ALLOWLIST="linear;rec709"

Environment variable	Description	Example value
	Tip: You can also limit the number of colorspaces available to artists using the Python API **registerConfigUiAllowlist()** function. See **Help** > **SDK** > **Python** > **Documentation** from Mari's UI menus for more details.	
MARI_COLORSPACE_USER_INTERFACE_MODE	Sets the colorspace mode Mari is using. You can set the following modes: • **0** - Basic mode This mode displays a limited amount of the colorspace options available in Mari. • **1** - Default mode This mode displays the standard colorspace options available in Mari. • **2** - Advanced mode This mode displays all the colorspace options available in Mari. **Article:** Take a look at the How to enable the advanced color management in Mari article.	**MARI_COLORSPACE_USER_INTERFACE_MODE=0**
MARI_DEFAULT_ARCHIVE_PATH	The default path to load and save project archives.	
MARI_DEFAULT_CAMERA_PATH	The default path to load and save cameras and projectors.	
MARI_DEFAULT_EXPORT_PATH	The default path to export textures to.	
MARI_DEFAULT_	The default path to load geometry from.	

Environment variable	Description	Example value
GEOMETRY_PATH		
MARI_DEFAULT_ IMAGE PATH	The default path to load and save reference images.	
MARI_DEFAULT_ IMPORT_PATH	The default path to import textures from.	
MARI_DEFAULT_ RENDER_PATH	The default path to save renders such as turntables.	
MARI_DEFAULT_ SHELF_PATH	The default path to load and save shelf files.	
MARI_DONT_AUTO_ TRIGGER_BACKUP	When set to **1**, this disables the automatic back up system, which triggers when the current project is saved. This should be set to **1** if you want to manually trigger the backup system using a script.	
MARI__DONT_TREAT_ EXR_PREMULTIPLIED	When set to **1**, this disables the pre-multiplication of OpenEXR images when exporting channels. Mari 8.0v1 follows the official Open EXR convention when handling Alpha Channels, which states: "OpenEXR images are "premultiplied" - the color channel values are already matted against black using the alpha channel." http://www.openexr.com/photoshop_plugin.html This means that by default when exporting a channel with an alpha channel, the R, G, and B components are multiplied against the alpha channel.	
MARI_FORCE_COLOR_ MANAGEMENT	When set to **0**, color management is forced off. When set to **1**, color management is on.	

Environment variable	Description	Example value
MARI__ENABLE_LEGACY_COLOR_MANAGER_PALETTE	When set to **1**, displays the **Color Manager (View Transform)** palette in Mari.	**Note:** The **Color Manager (View Transform)** palette is being deprecated and will be removed in a future release.
MARI_GIZMO_PATH	Nodes are exported from Mari as Gizmos in **.mng** files. Use **MARI_GIZMO_PATH** to specify a different directory from which to load Gizmos, in addition to the default Gizmos directory. **MARI_GIZMO_PATH** accepts multiple values separated by a **:** (colon) on Mac OS X and Linux operating systems, or a **;** (semi-colon) on Windows.	**Warning:** Mari loads any **.xml** files found at the directory, or any subdirectory, of the location you specified using **MARI_GIZMO_PATH**, even very large files, which may slow down projects loading.
MARI_INSTALL_PATH	When you intend to launch Mari from Nuke, from the **Mari** > **Launch Mari** menu, you can set the environment variable **MARI_INSTALL_PATH** so that you don't need to point to the Mari executable every time. For more information, refer to the Nuke<>Mari Bridge section.	For example: **MARI_INSTALL_PATH=/usr/local/Mari**<version>**/mari**
MARI_LOG_FILE	The name and location of the Mari log file. If this variable is not set, the log is saved to: • **MariLog.txt** in the **/home/**<Username>**/Mari/Logs** directory on Linux and Mac, or • **MariLog.txt** in the	• On Linux: **/user/Mari/MyLog.txt** • On Windows: **C:\TEMP\MyLog.txt** • On Mac:

Environment variable	Description	Example value
	C:\Users<Username>**\Documents\Mari\Logs** directory on Windows.	**/user/Mari/MyLog.txt**
MARI_MATERIAL_INGEST_PRESETS_SEARCHPATHS	This loads a custom list of presets in the Material Ingest Tool instead of the default Mari ones stored in **C:\Program Files\Mari 8.0v1 \Bundle\Media\Settings\MaterialIngestPresets** Usage: MARI_MATERIAL_INGEST_PRESETS_SEARCHPATHS=C:\material_presets	
MARI_MATERIAL_INGEST_TEMPLATES_SEARCHPATHS	The path where Mari looks for material ingest templates. Usage: MARI_MATERIAL_INGEST_TEMPLATES_SEARCHPATHS=C:\material_templates	
MARI_MODO_BAKE_PRESETS	The path where Mari looks for baking preset files. For more information, see Modo Render Settings	
MARI_MODO_RENDER_PRESETS	The path where Mari looks for rendering preset files. For more information, see Modo Render Settings	
MARI_NAMEDFILEIMPORT_SEQUENCELIMIT	The maximum number of patches objects are allowed to have for the **Named Files** tab to appear on the **Import Channel** and **Export Channel** dialogs. By default, the **Named Files** tab does not appear if you have more than 20 patches in your object, but you can change this using the **Patch Cutoff** preference.	**MARI_NAMEDFILEIMPORT_SEQUENCELIMIT=30**

Environment variable	Description	Example value
	You can use the **Named Files** tab to import or export individual files rather than a sequence of patches with the udim number (1001, 1002, etc.) in the file names.	
MARI_NO_CRASH_PROMPT	When set, the Crash Report Manager will not appear when Mari crashes. For more information on Mari's crash reporting, go to Crash Reporting in Mari.	
MARI_OLD_PYTHON_INIT	When loading a script, if existing start-up scripts do not work correctly due to the load module order, you can revert back to the old behavior by setting the environment variable **MARI_OLD_PYTHON_INIT** to any non-empty string other than **0**. For more information, refer to the Using Python in Mari section.	
MARI_PLUGINS_PATH	The location Mari searches for plug-ins to load at start-up.	For example: **MARI_PLUGINS_PATH=/home/holtc/mari**
MARI_PROJECT_BACKUP_THREADS	Overrides the default thread count (which is set at 2 threads) for if you want to improve the backup system performance. Note that setting this too high/ where hardware doesn't allow it can cause a negative impact on other processes.	
MARI_SCRIPT_PATH	A list of paths to run scripts from.	**~/Mari/Scripts** on Linux and Mac, or **C:\Users**<username>**\Documents\Mari\Scripts** on Windows
MARI_SVT_ATLAS_	Forces the **Layer Count** to a specific value that	**MARI_SVT_ATLAS_LAYER_**

Environment variable	Description	Example value
LAYER_COUNT	may not have been available through the Mari Preferences Dialog, see the Mari Preferences Dialog for more information. **Note:** This environment variable is targeted at MAC OS X users as the **Layer Count** preference is always disabled on this platform.	**COUNT=3**
MARI_SVT_SUPPRESS_WARNINGS	This stops the SVT from displaying the warning dialog when the SVT properties may cause issues on the current hardware.	**MARI_SVT_SUPPRESS_WARNINGS=true**
MARI_USER_PATH	Root of the default user path.	**~/Mari** on Linux and Mac, or **C:\Users**<username>**\Documents\Mari** on Windows.
MARI_VSYNC	Sets the VSync option to use to prevent screen tearing. There are three modes: • MARI_VSYNC=0 - Vsyn is off. • MARI_VSYNC=1 - Vsync is on. • MARI_VSYNC not set, the default behavior - Vsyn is controlled by the setting of the GPU driver installed. **Note:** You must restart Mari for this to take effect. If the VSync option is On, then this forces the main canvas to only update at the monitor's refresh rate. If it is Off, then the main canvas can update at any	

Environment variable	Description	Example value
	time. This could result in tearing, if the canvas is running faster than the monitor's refresh rate. By default, this option is not set and Mari uses the system and driver settings for VSync. **Article:** See Knowledge Base Article Q100642 for more information.	
MARI_WORKING_DIR	The location of temporary Mari files, such as crash logs. If this variable is not set, the files are saved to the same path as the log file.	• **MariLog.txt** in the **/home/**<username>**/Mari/Logs** directory on Linux and Mac, or • **MariLog.txt** in the **C:\Users**<username>**\Documents\Mari\Logs** directory on Windows.
SHOW_BASE_DIR	The path where Mari looks when importing images. You can call these variables anything, and as long as you precede the variable name in the image import path with **$** on Linux and Mac, or **%** on Windows. Mari replaces that part of the path with the value of the environment variable. For more information, see Image Importing	For example: • On Linux and Mac: If the import path is **$SHOW_BASE_DIR/Images/$SHOT**, **$SHOW_BASE_DIR** and **$SHOT** is replaced by the values of the environment variables **SHOW_BASE_DIR** and **SHOT**. • On Windows: If the import path is **%SHOW_BASE_DIR%/Images/%SHOT%**, **%SHOW_BASE_DIR** and **%SHOT** is replaced by the

Environment variable	Description	Example value
		values of the environment variables **SHOW_BASE_DIR** and **SHOT**.
XDG_CONFIG_HOME	The location of the directory where the Mari configuration files are stored. If you're on Windows, you can't change this directory, and the **Mari.ini** file can only be stored in the default location: <drive letter>**:\Users**<user>**\.mari\TheFoundry**.	On Linux and Mac: **/SharedDisk/Mari**
MARI_USD_LEGACY_EXPORT_TAB	Enables the legacy USD Export Tab.	

Image Importing

You can set environment variables that Mari looks when importing images. You can call these variables anything, and as long as you precede the variable name in the image import path with **$** on Linux and Mac, or **%** on Windows. Mari replaces that part of the path with the value of the environment variable. For example:

- On Linux and Mac: If the import path is **$SHOW_BASE_DIR/Images/$SHOT**, **$SHOW_BASE_DIR** and **$SHOT** is replaced by the values of the environment variables **SHOW_BASE_DIR** and **SHOT**.
- On Window: If the import path is **%SHOW_BASE_DIR%/Images/%SHOT%**, **%SHOW_BASE_DIR** and **%SHOT** is replaced by the values of the environment variables **SHOW_BASE_DIR** and **SHOT**.

Overriding Mari Configuration File

You can also use environment variables to override any setting in the Mari configuration file named **Mari8.0v1.conf** (on Linux) or **Mari8.0v1.ini** (on Windows and Mac). This file contains all user settings that are not project specific. By default, you can find it in the following sub-directory of your home directory:

- **.config/TheFoundry** (on Linux),

- **.mari\TheFoundry** (on Windows), or
- **.config/TheFoundry** (on Mac).

Here's a short example of what you may find the Mari configuration file:

```
[HUD]

Canvas_fps_hudVisibility=2

Canvas_triangleCount_hudVisibility=2

Project_currentObjectName_hudVisibility=2

Project_currentChannelName_hudVisibility=2



[Application]

ThreadCount=5
```

To override a setting in the Mari configuration file, you need to set an environment variable of the format **MARI_<Group>_<Name>**. So, for example, if you wanted to override the application thread count setting in the above example file and set it to two rather than five, you could set **MARI_APPLICATION_THREADCOUNT** to 2.

Article: Take a look at the Mari - extended environment variables article.

Setting Environment Variables

On Linux

1. The procedure for setting an environment variable depends on what your default shell is. To get the name of the shell you are using, launch a shell and enter **echo $SHELL**.

   This returns the type of shell you are using.
2. Depending on the output of the previous step, do one of the following:

- If your shell is a **csh** or **tcsh** shell, add the following command to the **.cshrc** or **.tcshrc** file in your home directory: **setenv VARIABLE value**. Replace **VARIABLE** with the name of the environment variable and **value** with the value you want to give it, for example:

  setenv MARI_CACHE /User/Temp/Mari
- If your shell is a **bash** or **ksh** shell, add the following command to the **.bashrc** or **.kshrc** file in your home directory: **export VARIABLE=value**. Replace **VARIABLE** with the name of the environment variable and **value** with the value you want to give it, for example:

  export MARI_CACHE=/User/Temp/Mari

For a list of the environment variables that Mari understands, see Environment Variables That Mari Recognizes.

On Windows

1. Right-click on **My Computer** and select **Properties**.
2. In the **System** window click on **Advanced system settings** in the left pane.
3. Go to the **Advanced** tab.
4. Click the **Environment Variables** button.

   The **Environment Variables** dialog opens.
5. Click the **New** button under either **User variables** or **System variables**, depending on whether you want to set the variable for the current user or all users. To set environment variables for all users, you need to have administrator privileges.
6. In the **Variable name** field, enter the name of the environment variable you want to set. For a list of the environment variables Mari understands, see Environment Variables That Mari Recognizes.
7. In the **Variable value** field, enter the value for the variable. The value can be a directory path, for example.
8. Click **OK**.

For a list of the environment variables that Mari understands, see Environment Variables That Mari Recognizes.

Article: How to set environment variables
How to delete or unset environment variables
How to list environment variables

Checking Environment Variables Are Set

On Linux

1. Launch a shell, if you don't have one open already.
2. Enter **echo $VARIABLE**. Replace **VARIABLE** with the name of the environment variable you set earlier. For example, to check if MARI_CACHE is set, enter **echo $MARI_CACHE**.

   If the variable is set, its value is displayed in the shell window.

On Windows

1. Select **Start** > **All Programs** > **Accessories** > **Command Prompt**.
2. In the command window that opens, enter **echo %VARIABLE%**. Replace **VARIABLE** with the name of the environment variable you set earlier. For example, to check if MARI_CACHE is set, enter **echo %MARI_CACHE%**.

   If the variable is set, its value is displayed in the command window.

On Mac

1. Launch a terminal, if you don't have one open already.
2. Enter **echo $VARIABLE**. Replace **VARIABLE** with the name of the environment variable you set earlier. For example, to check if MARI_CACHE is set, enter **echo $MARI_CACHE**.

If the variable is set, its value is displayed in the terminal window.

Article: How to get environment variable values

Importing Plug-ins

This page explains a generalized process for importing plug-ins into Mari.

1. Once plug-ins have been downloaded, navigate to this folder in your home directory:
   On Windows: C:\Users\<user>\Documents\Mari\Scripts
   On Linux/Mac: UserDirectory\Mari\Scripts
   If you do not already have the **\Scripts** folder, you can create it.
2. Select your downloaded plug-in files from your downloads folder, and **Extract** these files from zip into your **\Scripts** folder.
3. Restart Mari. The new plug-in may show a splash screen to confirm that it has been downloaded. If not, go to **Tools** > **Plug-ins**. Here you can see a list of the installed plugins.

Tip: You can also go to **Python** > **Show Console** to see if any plugins have been installed.

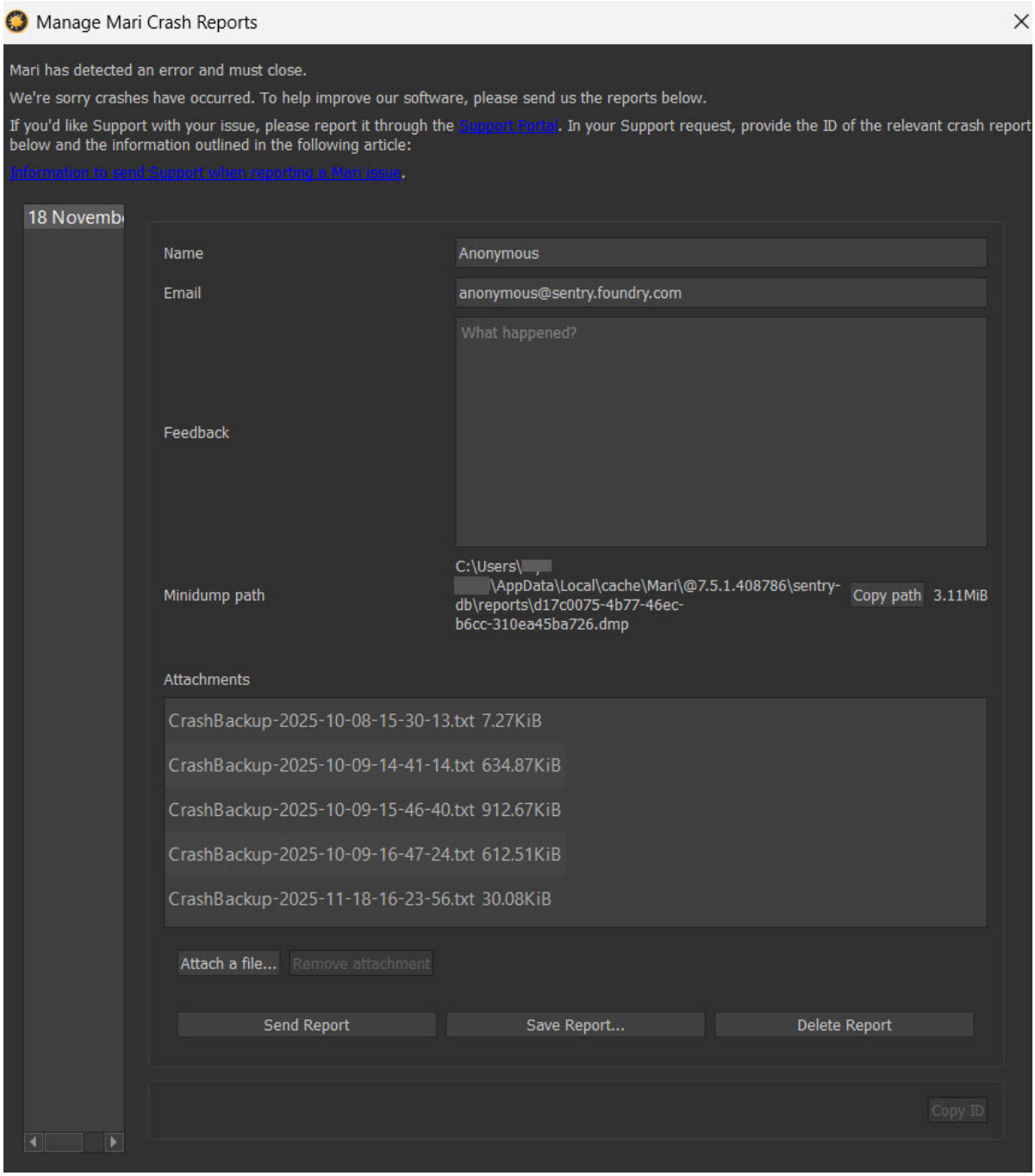
Note: Some plug-ins may require additional or alternative steps: refer to provider's instructions for further information.

Crash Reporting in Mari

If you experience a crash in Mari, the Crash Report Manager will appear. This manager enables you to send the crash report to Foundry, or save it for your own use. Sending the crash reports to Foundry can help the development team identify issues that users encounter, and hopefully address them.

Mari uses Sentry for handling crash reports. Sentry enables you to easily submit crash reports without sharing any sensitive project information.

You can also find this window at any point by going to **Help** > **Manage Crash Reports**.

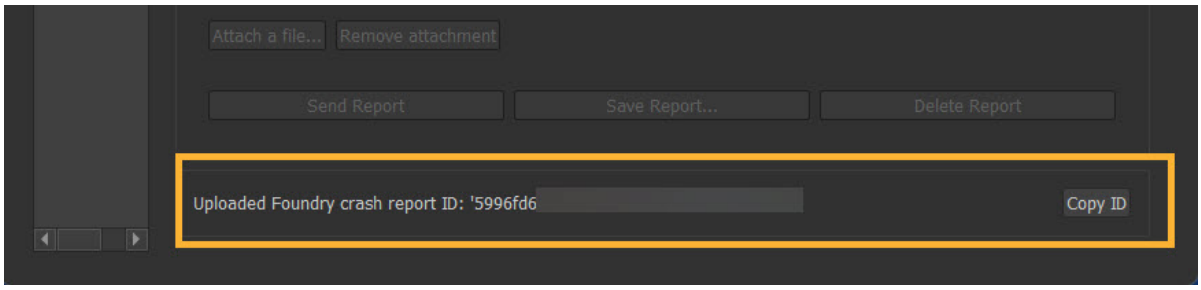


- **Name** and **Email** - You can enter your name and email here, or leave the anonymous options. Entering your name and email is helpful if you wish to reach out to Foundry to discuss the crash, as it identifies crash reports that you sent.
- **Feedback** - Here you can give information about what happened when Mari crashed.
- **Minidump path** - You can copy the minidump path to access the crash report file. If you want to contact Foundry directly, you should send this file with your message. This is also helpful if you are working on a machine without internet, as you can send the file to Foundry when you next go online.

- **Attachments** - Use **Attach a file...** to attach any supporting documents that give additional information about the crash. By default Sentry includes a censored report as an attachment.

> **Tip:** Additional information, such as screenshots, system reports, or a Mari Archive can all be helpful information depending on the crash. To find out more about information you can add to your crash report, go to Q100090: Information to send Support when reporting a Mari issue.

- **Send report** - This button sends the crash report to Foundry. Once you have sent the report, the crash report ID will appear at the bottom of the window. You can copy this ID for future reference, for example if you wish to reach out to Support.



- **Save Report...** - this option opens a file browser and saves the report to your machine. Once you have sent and closed the Crash Report Manager, the report is removed from Mari. Ensure to save your report before you sent it if you want it for future reference.

> **Note:** Crash reports should be sent before any further crashes occur, as the folder for DMP files has an 8MB size limit, which if reached will automatically start deleting older reports.

- **Delete Report** - deletes the crash report.

> **Note:** If you don't want the Crash Report Manager to appear when Mari crashes, you can completely disable the display of the Issue Reporter when a crash occurs in Mari. This can be helpful if your machine isn't connected to the internet so you cannot send the report to Foundry. To do this, set the environment variable **MARI_NO_CRASH_PROMPT=1** .
> Crash reports will still be saved however, and can be accessed in Mari under **Help > Manage Crash Reports**. From here you can save the report and send it to Support if you require assistance.

> **Note:** For more information on using crash reports when contacting Foundry Support, go to Q100729: Managing and sending Mari Crash Reports to Foundry.

Appendices

This section contains supplemental reference information that you may need when using Mari.

Organization of the Section

The section consists of the following appendices:

- Appendix A: Shortcuts lists the keyboard shortcuts you can use for quicker and easier access to Mari's features. You can also open a list of keyboard shortcuts from the application by selecting **Edit** > **Shortcuts**.
- Appendix B: Frequently Asked Questions and Troubleshooting lists of some of the most common questions about Mari.
- Appendix C: Third-Party Licenses lists third party libraries used in Mari.

Appendix A: Shortcuts

Keyboard Shortcuts

Note: In many Linux windows managers, the **Alt** key is used by default as a mouse modifier key. This can cause problems in 3D applications where **Alt** is used for camera navigation in 3D environments.

You can use key mapping to assign the mouse modifier to another key, such as the (**Super** or **Meta**) key, but the method changes depending on which flavor of Linux you're using. Please refer to the documentation on key mapping for your particular Linux distribution for more information.

Painting

Bake	**B**
Bake and Clear	**Ctrl+Shift+B**
Undo	**Ctrl+Z**
Redo	**Ctrl+Y**
Toggle Painting	**.**
Paint	**P**
Paint Through	**U**
Paint Buffer Eraser	**E**
Toggle Mask Preview	**,**
Toggle Edge Mask	**G**
Toggle Masking	**Ctrl+M**
Paint Target quick palette	**I**
Colors quick palette	**J**
Shelf quick palette	**K**
Image Manager quick palette	**L**
Pie Menu	**F9**
Zoom Paint Buffer	**Z**
Clear Paint Buffer	**Ctrl+Shift+C**
Next Blend Mode	**Shift+)**
Previous Blend Mode	**Shift+ (**
Last Blend Mode	**Ctrl+9**
Reset Blend Mode	**Ctrl+0**
Clear Blend Mode	**Shift+Backspace**

Selection

Quick Copy	**Ctrl**+**C**
Quick Paste	**Ctrl**+**V**
Select	**S**
Select All	**Ctrl**+**A**
Select None	**Ctrl**+**E**
Hide Selected	**H**
Hide Unselected	**Shift**+**H**
Show All	**Ctrl**+**Shift**+**H**
Clear Marquee Selection	**Ctrl**+**D**
Invert	**Ctrl**+**I**
Toggle selection	**Ctrl**+**Shift**+click

Brush

Adjust Radius	**R**
Rotation	**W**
Squish	**Q**
Toggle Brushes	**N**
Adjust Opacity	**O**
Increase/Decrease Aspect Ratio	**0** / **9**
Increase/Decrease Opacity	**=** / **-**

Increase/Decrease Radius	**[/]**
Increase/Decrease Rotation	**' / ;**

Tools

Transform	**M**
Mouse Gestures	****
Hide/Show Widgets	**/**
Color Picker and Pixel Analyzer	**C**
Toggle Vector Inspector	**Shift+I**
Modify Vector Inspector	**V**

Camera

Focus on Cursor Position	**F**
Camera View Presets	**1**,**2**,**3**,**4**,**5**,**6**
Focus on Selection	**A**
View All	**Shift**+**A**
Ortho Camera	**F10**
Perspective Camera	**F11**
UV Camera	**F12**
Orbit	**Alt**+drag
Zoom	**Alt**+**Ctrl**+drag left or right

Pan	**Alt**+**Shift**+**Left-click**+drag or **Alt**+**Middle-click**+drag
Roll	**Ctrl**+**R**+drag

View

Go to Next Tab	**Ctrl**+**Tab**
Go to Previous Tab	**Ctrl**+**Shift**+**Back Tab**
Full Screen	**Ctrl**+**Shift**+**F**
Previous View	**Ctrl**+ **,**
Next View	**Ctrl**+ **.**
Show Python Console	**Alt**+**Ctrl**+**P**
Toggle Cursor Type	**F5**

Animation

Move Forward One Frame	**Ctrl**+**Right**
Move Backward One frame	**Ctrl**+**Left**
Fast Forward	**Ctrl**+**Up**
Rewind	**Ctrl**+**Down**
Add keyframe	**Ctrl**+**K**
Scrub Animation	**Shift**+**?**

Project

New	**Ctrl**+**N**
Save	**Ctrl**+**S**
Close	**Ctrl**+**W**

General

Quit	**Ctrl+Q**
Hide/show Palettes	**Home** (**Fn** + Left arrow)
Toggle Tool Properties Palette	**F7**
Toggle Wireframe View	**Shift**+**W**
Take Screenshot	**Ctrl**+**Shift**+**Print**
Space bar	Press the **Space** bar while hovering over the canvas to toggle between showing and hiding the currently open palettes. You can also press the **Space** bar while hovering over a palette to expand it. **Note:** You can't expand a palette if you selected an item in it.

Channels

Channel Menu	**I**

Bake on New Layer	**Ctrl**+**B**
Duplicate	**Alt**+drag and drop
Select first channel visible	**Page Up**
Select last channel visible	**Page Down**
Toggle between the last two selected channels	**T**

Lighting

Flat	**F1**
Basic	**F2**
Full	**F3**

Color

Swap Foreground and Background	**X**
Set to Black & White	**D**

Navigation

Orbit	**Alt**+**left-click**+drag
Pan	**Alt**+**Shift**+**Left-click**+drag or **Alt**+**Middle-click**+drag
Roll	**Ctrl**+**R**+**Left-click**+drag
Zoom	**Alt**+**Ctrl**+**Left-**

	click+drag or **Alt**+**Right-click**+drag

UV View Presets

Reset Model to Default	**1**
Rotate 90 degrees counterclockwise	**2**
Rotate 180 degrees counterclockwise	**3**
Rotate 270 degrees counterclockwise	**4**

Node Graph

Fit All Nodes in Node Graph	**A**
Focus Selection in Node Graph	**F** (If no selection is made, behaves the same as **Fit All Nodes in Node Graph**.)
Search Node	**J**
Create Paint Node	**P**
Create Group	**Ctrl**+**G**
Copy Selection	**Ctrl**+**C**
Paste Selection	**Ctrl**+**V**
Bypass Selected Node	**Ctrl**+**D**
Disable Selection	**D**
Delete Selection	**Delete**

Extract Node	**Ctrl**+**Shift**+**X**
Autoplace	**L**
Switch Node Inputs	**Shift**+**X**
Pan in Node Graph	**Middle-click**+drag, **Alt**+**middle-click**+drag, or **Alt**+**right-click**+drag
Zoom in Node Graph	Mouse wheel scroll or **Alt**+**middle-click**+drag
Pan up	**Alt** + Up arrow
Pan down	**Alt** + Down arrow
Pan to the right	**Alt** + Right arrow
Pan to the left	**Alt** + Left arrow
Zoom out	**Shift** + **Alt** + Up arrow
Zoom in	**Shift** + **Alt** + Down arrow
Select the previous sibling node of the selected node. Sibling nodes are nodes on the same hierarchical level under the same parent node.	Up arrow
Select the next sibling node of the selected node. Sibling nodes are nodes on the same hierarchical level under the same parent node.	Down arrow
Add the previous sibling node of the selected node to the current selection. Sibling nodes are nodes on the same hierarchical level under the same parent node.	**Shift** + Up arrow
Add the next sibling node of the selected node to the current selection. Sibling nodes are nodes on the same hierarchical level under the same parent node.	**Shift** + Down arrow
Select the next upstream node(s) of the selected node	Left arrow
Add the next upstream node(s) of the selected node	**Shift** + Left arrow

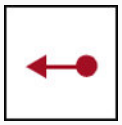
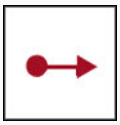
to the current selection	
Select all nodes upstream of the selected node	**Ctrl** + Left arrow
Select nodes upstream of the selected node that directly affect the selected node	**Shift** + **Ctrl** + Left arrow
Select the next downstream node of the selected node	Right arrow
Add the next downstream node of the selected node to the current selection	**Shift** + Right arrow
Select all nodes downstream of the selected node	**Ctrl** + Right arrow
Select nodes downstream of the selected node that the selected node directly affects	**Shift** + **Ctrl** + Right arrow
Select all nodes between the currently selected nodes	**Ctrl** + **Shift** + Space bar
Open the **Node Properties** panel for the selected nodes	**Ctrl** + **P**
Fully Collapse Selected Node	**Alt** + **1**
Collapse Selected Node to show Connected Only	**Alt** + **2**
Fully Expand Selected Node	**Alt** + **3**
Stream Collapse Selected Node (Multi Channel Merge Nodes only)	**Alt** + **4**
Cycle Through Port List Mode	**Shift** + **~**
Merge two multi-channel material nodes together by creating a **Multi-Channel Merge** node.	**M**
Swap the nodes by pressing and holding S while dragging one node on top of another.	Hold **S**
To move a whole Backdrop node, press and hold W and drag it to a new location. Shift and Ctrl modifiers can be used to add or remove from the current selection.	Hold **W**

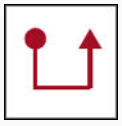
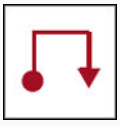
Select a Backdrop and hold B to select the Backdrop but not any of its contained nodes.	Hold **B**

Python Console

Indent selected string or add four spaces at the cursor position when no string is selected	**Tab**
Unindent selected string	**Shift**+**Tab**
Toggle commenting on selected lines, adding or removing # notation	**Ctrl**+**/**
Zoom in on text in the Python Console	**Ctrl**+ **+**
Zoom out from the text in the Python Console	**Ctrl**+ **-**
Evaluate a script in the Python Console	**Ctrl**+**Enter**
Evaluate selected statements in the Python Console	**Ctrl**+**Shift**+**Enter**
Step backwards through previously evaluated statements	**Alt**+**Up Arrow**
Step forwards through previously evaluated statements	**Alt**+**Down Arrow**

Mouse Gestures

Mouse gestures			
Hold down \ and draw...			
Undo		Redo	

Mouse gestures			
Clear Painting		Bake	
Previous Channel		Next Channel	
Toggle Last Two Channels		Paint Tool	
Paint Through Tool		Selection Tool	
Paint Buffer Eraser Tool		Show Palettes	
Increase Brush Radius		Decrease Brush Radius	
Hide Unselected		Hide Selected	
Show All		View All	
Toggle Brush Tip			

Appendix B: Frequently Asked Questions and Troubleshooting

Below are some of the most common questions about Mari. If your question isn't answered here, review the documentation and the release notes or contact Foundry's customer support for help.

I can’t see or bake what I've painted.

There can be several reasons for this:

- **Paint Opacity** has been set to 0 in the **Painting** palette.
- A mask is preventing paint from appearing. Check if any of the masks are enabled in the **Painting** palette.
- **Project On** has been set to **Selected Only** in the **Painting** palette, but you haven't selected any objects, patches, or faces. Either select the areas where you want paint to appear or set **Project On** to **All**.
- The paint is on objects, patches or faces that have been locked. To unlock them, click **Selection** > **Unlock Selected**, **Unlock All**, or **Unlock Entire Object**.
- The paint blending mode (Painting **Mode** in the **Painting** palette) has been set to a mode that does nothing to your textures. For example, if your texture is white, the paint is red and **Painting Mode** has been set to **Hue**, the paint is not visible.
- **Brush Opacity** has been set to **0** in the **Tool Properties** palette.
- **Brush Flow** has been set to **0** in the **Tool Properties** palette.
- The paint buffer has been moved off the screen. To reset the buffer to its defaults, click **Reset** under **Paint Buffer** in the **Painting** palette.

My object has disappeared from the canvas.

The object may have been moved off the screen. Try pressing **A** or **F** to bring it back.

Mari’s bake behavior seems strange.

In the **Painting** palette, you can set whether Mari bakes automatically when you change the view, and whether the buffer clears automatically once you bake. Set **Bake Behavior** to:

- **AutoBakeAndClear** – Mari automatically bakes and clears the buffer when you move the model.

- **Manual** – You have to bake manually; and when you do, the paint stays in the paint buffer until you manually clear it.
- **ClearOnly** – You have to bake manually; and when you do, the paint buffer clears automatically.

Mari seems slow when using the Clone Stamp tool.

When cloning from a particular channel, the **Clone Stamp** tool takes a "snapshot" of that channel as it appears on the screen and uses that as the source. As a result, the resolution of the cloned texture can depend on the current display resolution within Mari. Higher display resolutions produce better cloning results, but can dramatically slow Mari's performance.

Increasing the Paint Buffer Size or Resolution and Layer Count can affect Mari's performance.

Increasing the **Buffer Size** or the **Resolution** and **Layer Count**, even when your graphics card supports large textures, can slow Mari's performance. Adjust the **Buffer Size** under **Paint Buffer** in the **Painting** palette, and the **Resolution** in the **Mari Preferences** dialog under **GPU** > **Virtual Texture**.

Displacement preview performance seems slow.

If you have allocated a channel to be used in a **Displacement** shader component, the **Type** under **Edit** > **Preferences** > **GPU** > **Virtual Texture**, should be set to at least **Half** for more precise displacement. In addition, the channel containing the displacement map should be set to at least **16-bit**. Please note that using values higher than these may reduce preview performance.

Mari keeps highlighting anything I've selected with a selection color. How can I turn this off?

You can configure the way Mari highlights selected areas. Right-click on the canvas and select **Display Properties**. Under **Selection**, adjust **Fill Render** and **Outline Render**.

I can't find a Save As option.

Mari currently has no **Save As** functionality. You can, however, duplicate your project by right-clicking on it in the **Projects** view and selecting **Copy**.

If you want to pass your project on to another user, you can also select the project in the **Projects** view and click **Archive**. This creates an archive (a **.mra** file) that contains your project and all the textures and models associated with it.

If you use **.** (full stop) to separate parts of the file name (for example, "test.file"), Mari does not add the file extension **.mra** automatically. This allows you to use a different extension if desired. However, in most cases you most likely want to add the **.mra** extension to ensure the file is clear.

I want to bake what I've painted, but I can't find the bake button.

The bake button sits at the bottom right of the workspace. It only appears when you have unbaked paint in the paint buffer.

You can also press **B** to bake.

Clicking on another application caused the floating palettes in Mari to disappear.

This is not a problem in Mari but caused by your Linux window behavior settings. To fix this on KDE 4, for example, you can:

1. Go to **Settings** > **System Settings** > **Window Behavior** > **Advanced**.
2. Uncheck **Hide utility windows for inactive applications**.

I am having problems with my Wacom tablet.

We have found that these links have been useful in solving the problems for some of our users:

https://help.ubuntu.com/community/Wacom

http://linuxwacom.sourceforge.net/

On Linux, pressing the Ctrl/Cmd, Alt, or ■ (Windows) shortcut key doesn't do what it should. Instead, it does something else, like moves the entire Mari window.

This is usually caused, not by a problem in Mari, but rather by your Linux window manager, which is using one of these keys as a global shortcut. Depending on your window manager, you may be able to block the global shortcut, or select which of the three keys is used as the global shortcut. For example, if the Mari shortcuts you often use require you to press the **Alt** key, you can set the global shortcut to be the **Windows** key instead. This way, the Mari shortcuts that don't work are limited to those that require the **Windows** key.

To block the global shortcut on a per-application basis (that is, for Mari only) on KDE 4, you can:

1. Right-click on the Mari title bar and select **Advanced** > **Special Window Settings**. This may open an information dialog where you need to click **OK**.
2. Go to the **Window** tab.
3. Click the **Detect Window Properties** button and then click on the Mari window.
4. Make sure that **Use whole window class** and **Match also window title** are checked and click **OK**.
5. Go to the **Workarounds** tab.
6. Check **Block global shortcuts**, select **Force** from the dropdown menu, and enable the checkbox on the right. Click **OK**.

To select which key is used as the global shortcut on KDE 4, you can:

1. Go to **Settings** > **System Settings** > **Window Behavior** > **Window Actions**.
2. Under **Inner Window**, set **Modifier** to the shortcut key you need the least in Mari or other applications you use (for example, the **Windows** key).

   This allows you to use the other two keys (in this case, **Ctrl**/**Cmd** and **Alt**) in Mari.

On Windows, pressing the Windows shortcut key should open the Start menu. This doesn't happen when Mari has focus. Why is that?

This is because the **Windows** key is a shortcut key in Mari. Whenever Mari has focus, the **Windows** key is reserved for Mari, so that you can use all the Mari shortcuts.

If you need to access the **Start** menu, you can either click it manually or move focus to another application and then press the **Windows** key.

Pressing the Alt Gr key does not trigger the same shortcuts as the Alt key. Why is that?

The **Alt Gr** key is defined differently on the Windows and Linux operating systems. Due to this, Mari cannot determine what the **Alt Gr** key is intended to be linked to. If a shortcut lists the **Alt** key as part of the controls, this does not include **Alt Gr** as an alternative. For this reason, it is advisable to avoid using the **Alt Gr** key in your custom shortcuts.

On Windows, I've removed a drive from Mari's file browser bookmarks. How do I get it back? Is it possible to delete a drive from the bookmarks permanently?

All logical drives that Windows can see are automatically added to Mari's file browser bookmarks when you start Mari. You can remove these while Mari is running, but the only way to get them to reappear is to relaunch Mari. At present, you can't delete a drive from the bookmarks permanently - it always reappears when you relaunch Mari.

Mari also automatically adds mapped network drives to its bookmarks. You should not disconnect a network drive while Mari is running and then attempt to reference files from that drive in Mari, as that may result in undefined behavior.

Generally, I need the channel to have precision when I paint in a linear colorspace, and I use 16-bit channels if I apply the gamma after the paint has been applied. Now that colors are linearized before baking, do I need to use HDR mode for my paint to be applied without artifacts or can I paint in a linear colorspace but with 8-bit channels?

When working with 8-bit (byte) channels in a non-linear colorspace, it is recommended that you set the virtual texture's type and paint buffer's color depth to 16-bit (half) or 32-bit (float). This keeps the project's data footprint down but the data is still converted to floating point and linearized on upload to the GPU for painting, this way it keeps the colorspace transform error down to a minimum. When Mari bakes paint and

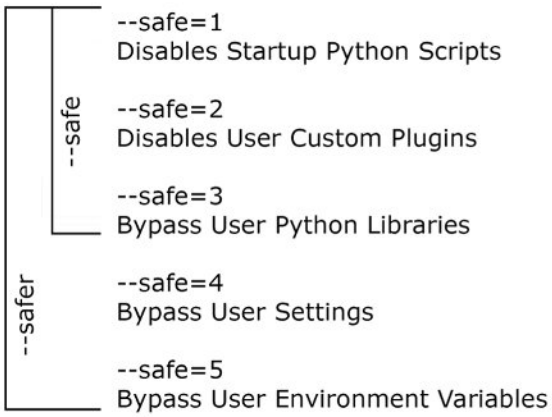
texture data back into the channel, the inverse is applied, so the floating point data is color corrected and converted back to 8-bit (byte).

If you want to eliminate errors as much as possible, it is recommended to use floating point channels and then export to the colorspace and bit-depth of your choosing. However, this means the project is a lot larger and performance might be an issue on a less capable machine. It is not possible to eliminate all the errors even if using full 32-bit float everywhere as you still get floating point and interpolation errors being introduced.

Troubleshooting

Advanced Debugging Methods

If you are launching Mari from the command line or terminal, you can run the application in one of two safe modes: `--safe` and `--safer`. These safe modes run a number of options, described in the *Installation and Licensing* chapter of the *Mari Getting Started Guide*.



If you prefer, you can choose instead to run individual safe mode options rather than those grouped under `--safe` or `--safer`. To do this, type the given number for the option after the `--safe` command. Please be aware that these work on an "up to and including" fashion, so selecting --safe=3 also runs --safe=1 and --safe=2 (effectively, operating the same as running the --safe command).

Safe Mode	Corresponding Safe Feature
--safe=1	Disables startup Python scripts.
--safe=2	Disables user custom plug-ins.

Safe Mode	Corresponding Safe Feature
--safe=3	Bypasses user Python libraries.
--safe=4	Bypasses user settings.
--safe=5	Bypasses user environment variables.
--safe=6	Disables any Python scripts in the Mari installation that have been tampered with.
--safe=7	Disables any Node Graph nodes in the Mari installation that have been tampered with.
--safe=8	Forces the **nuke-default** OCIO colorspace configuration. **Warning:** If your project's OCIO config file(s) differ from the one specified in --safe=8 mode, the colorspace data changes permanently. It is highly recommended that you archive your project before opening it in --safe=8 mode.

Appendix C: Third-Party Licenses

Note: **Note:** If, for any reason, you think Foundry is not entitled to use these libraries and fonts, please visit support.foundry.com.

Third-Party Library Versions

The following table lists third-party libraries included in the application and their current version. This version of Mari is compliant with VFX Reference Platform 2025.

Library	Version	
	Linux	Windows
ACES	1.0.3, 1.1, and 1.3	1.0.3, 1.1, and 1.3

Library	Version	
	Linux	Windows
Note: Note: ACES is supported via the OpenColorIO library.		
Alembic	1.8.8	1.8.8
Autodesk FBX	2020.3.7	2020.3.7
Blosc	1.14.3	1.14.3
Boost	1.85.0	1.85.0
bzip2	1.0.8	1.0.8
ca-bundle	2026.05.14	2026.05.14
catch2	3.14.0	3.14.0
Clang	N/A	N/A
cpp-httplib	0.47.0	0.47.0
Curl	8.5.0	8.5.0
Doxygen	1.9.6	1.9.6
Expat	2.7.4	2.7.4
fakeit	2.0.9	2.0.9
FBO (as part of openMAF)	1.1.2.1	1.1.2.1
fmt	8.1.1	8.1.1
Fontconfig	2.13.93	N/A
Freetype	2.13.2	N/A
GCC	11	N/A
GLEW	1.13.0	1.13.0

Library	Version	
	Linux	Windows
GLib	2.89.0	2.89.0
GLSL Cellular Noise	2011-08-22	2011-08-22
GLSL Classical (Perlin) and Simplex Noise	2011-08-22	2011-08-22
GLSL Perlin Noise	2011-08-22	2011-08-22
gperf	3.1	3.1
GTest	1.12.1	1.12.1
GTS	0.7.6	0.7.6
imath	3.1.12	3.1.12
Intel TBB	2021.13.0	2021.13.0
JPEG	9e	9e
jwt-cpp	17-11-2024	17-11-2024
libdeflate	1.23	1.23
libffi	3.4.2	3.4.2
libnoise	1.0	1.0
libtiff	4.6.0	4.6.0
libxcb-cursor	0.1.3	N/A
libxml2	2.13.8	N/A
MaterialX	1.39.2	1.39.2
Microsoft Visual C++ Redistributable	N/A	14.44.35211
minizip-ng	4.0.4	4.0.4

Library	Version	
	Linux	Windows
moltenvk	N/A	N/A
nlohmann-json	3.11.2	3.11.2
OpenColorIO	2.4.2	2.4.2
OpenCV	4.12.0	4.12.0
OpenEXR	3.3.14	3.3.14
OpenImageIO	2.5.7.0	2.5.7.0
OpenSSL	3.5.8	3.5.8
OpenSubdiv	3.6.0	3.6.0
OpenVDB	12.0.0	12.0.0
Partio	1.14.6	1.14.6
patchelf	0.11	N/A
pcre2	10.42	10.42
PNG	1.6.40	1.6.40
PTex	2.4.2	2.4.2
pybind11	2.9.2	2.9.2
pyside-shiboken	N/A	6.5.3
pystring	1.1.4	1.1.4
Python	3.11.11	3.11.11
Qt	6.5.3	6.5.3
Qt for Python (PySide)	6.5.3	6.5.3

Library	Version	
	Linux	Windows
Note: Note: Shiboken is part of PySide and uses the same version.		
QtSingleApplication	2.6.0	2.6.0
QuaZip	1.1	1.1
Revulytics	7.5.0	7.5.0
RLM	16.0	16.0
robin-map	0.6.3	0.6.3
sentry	0.7.6	0.7.6
Skein-Hash	1.1	1.1
SQLite	3.50.1	3.50.1
sse2neon	N/A	N/A
symbolicator	26.3.1	26.3.1
Tcl	8.6.10	8.6.10
USD	25.08	25.08
Visual Studio	N/A	2022
Vulkan	1.4.309.0	1.4.309.0
Wild Magic	5.0	5.0
WIX	N/A	3.11
yaml-cpp	0.8.0	0.8.0
zlib	1.3.1	1.3.1

Third-Party Library Licenses

The following table lists third-party libraries and their licenses.

Library	License
Alembic	BSD-3-Clause TM & Â© 2009-2015 Lucasfilm Entertainment Company Ltd. or Lucasfilm Ltd. All rights reserved. Industrial Light & Magic, ILM and the Bulb and Gear design logo are all registered trademarks or service marks of Lucasfilm Ltd. Â© 2009-2015 Sony Pictures Imageworks Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Industrial Light & Magic nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY

Library	License
Autodesk FBX	This software contains AutodeskÂ® FBXÂ® code developed by Autodesk, Inc. Copyright 2017 Autodesk, Inc. All rights, reserved. Such code is provided â€œas isâ€ and Autodesk, Inc. disclaims any and all warranties, whether express or implied, including without limitation the implied warranties of merchantability, fitness for a particular purpose or non-infringement of third party rights. In no event shall Autodesk, Inc. be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of such code.

Library	License
Blosc	BSD-3-Clause BSD License For Blosc - A blocking, shuffling and lossless compression library Copyright (C) 2009-2017 Francesc Alted <francesc@blosc.org> Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name Francesc Alted nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
Boost	BSL-1.0 Boost Software License - Version 1.0 - August 17th, 2003 Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following: The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
bzip2	bzip2-1.0.6 ---------- This program, "bzip2", the associated library "libbzip2", and all documentation, are copyright (C) 1996-2019 Julian R Seward. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 3. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT

Library	License
catch2	BSL-1.0 Boost Software License - Version 1.0 - August 17th, 2003 Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following: The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
cpp-httplib	MIT The MIT License (MIT) Copyright (c) 2017 yhirose Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
Curl	curl COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1996 - 2023, Daniel Stenberg, <daniel@haxx.se>, and many contributors, see the THANKS file. All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

Library	License
Doxygen	GPL-2.0 GNU GENERAL PUBLIC LICENSE Version 2, June 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it. For example, if you distribute copies of such a program,

Library	License
Expat	MIT Copyright (c) 1998-2000 Thai Open Source Software Center Ltd and Clark Cooper Copyright (c) 2001-2025 Expat maintainers Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
fakeit	MIT The MIT License (MIT) Copyright (c) 2013 Eran Pe'er Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
fmt	BSD-2-Clause Copyright (c) 2012 - present, Victor Zverovich Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. --- Optional exception to the license --- As an exception, if, as a result of your compiling your source code, portions of this Software are embedded into a machine-executable object form of such source code, you may redistribute such embedded portions in such object form without including the above copyright and permission notices.

Library	License
Fontconfig	HPND fontconfig/COPYING Copyright Â© 2000,2001,2002,2003,2004,2006,2007 Keith Packard Copyright Â© 2005 Patrick Lam Copyright Â© 2007 Dwayne Bailey and Translate.org.za Copyright Â© 2009 Roozbeh Pournader Copyright Â© 2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020 Red Hat, Inc. Copyright Â© 2008 Danilo Å egan Copyright Â© 2012 Google, Inc. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the author(s) not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty. THE AUTHOR(S) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE AUTHOR(S) BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. fontconfig/fc-case/CaseFolding.txt Â© 2010 Unicode, Inc.

Library	License
Freetype	FTL The FreeType Project LICENSE ---------- 2006-Jan-27 Copyright 1996-2002, 2006 by David Turner, Robert Wilhelm, and Werner Lemberg Introduction ========== The FreeType Project is distributed in several archive packages; some of them may contain, in addition to the FreeType font engine, various tools and contributions which rely on, or relate to, the FreeType Project. This license applies to all files found in such packages, and which do not fall under their own explicit license. The license affects thus the FreeType font engine, the test programs, documentation and makefiles, at the very least. This license was inspired by the BSD, Artistic, and IJG (Independent JPEG Group) licenses, which all encourage inclusion and use of free software in commercial and freeware products alike. As a consequence, its main points are that: o We don't promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution) o You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage) o You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (`credits') We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project. Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus

Library	License
GLEW	BSD-3-Clause The OpenGL Extension Wrangler Library Copyright (C) 2002-2007, Milan Ikits <milan ikits[]ieee org> Copyright (C) 2002-2007, Marcelo E. Magallon <mmagallo []debian org> Copyright (C) 2002, Lev Povalahev All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * The name of the author may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
GLib	LGPL-2.1-or-later GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999 Copyright (C) 1991, 1999 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. [This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.] Preamble The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below. When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things. To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it. For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

Library	License
gperf	GPL-3.0-only GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble The GNU General Public License is a free, copyleft license for software and other kinds of works. The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things. To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

Library	License
GTest	BSD-3-Clause Copyright 2008, Google Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
GTS	LGPL-2.0-only GNU LIBRARY GENERAL PUBLIC LICENSE Version 2, June 1991 Copyright (C) 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. [This is the first released version of the library GPL. It is numbered 2 because it goes with version 2 of the ordinary GPL.] Preamble The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This license, the Library General Public License, applies to some specially designated Free Software Foundation software, and to any other libraries whose authors decide to use it. You can use it for your libraries, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things. To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library, or if you modify it.

Library	License
imath	BSD-3-Clause Copyright Contributors to the OpenEXR Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
Intel TBB	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
JPEG	IJG The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy. This software is copyright (C) 1991-2020, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below. Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions: (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation. (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group". (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind. These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us. Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's

Library	License
jwt-cpp	MIT MIT License Copyright (c) 2018 Dominik Thalhammer Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
libdeflate	MIT Copyright 2016 Eric Biggers Copyright 2024 Google LLC Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
libffi	MIT libffi - Copyright (c) 1996-2021 Anthony Green, Red Hat, Inc and others. See source files for details. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the ``Software''), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED ``AS IS'', WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
libtiff	libtiff # LibTIFF license Copyright Â© 1988-1997 Sam Leffler\ Copyright Â© 1991-1997 Silicon Graphics, Inc. Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Library	License
libxcb-cursor	X11-distribute-modifications-variant Copyright Â© 2013 Michael Stapelberg Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the names of the authors or their institutions shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from the authors.

Library	License
libxml2	MIT Except where otherwise noted in the source code (e.g. the files dict.c and list.c, which are covered by a similar licence but with different Copyright notices) all the files are: Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is fur- nished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT- NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
MaterialX	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
Microsoft Visual C++ Redistributable	MSVC-RUNTIME-LICENSE MICROSOFT SOFTWARE LICENSE TERMS MICROSOFT VISUAL C++ 2015 - 2022 RUNTIME These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. They apply to the software named above. The terms also apply to any Microsoft services or updates for the software, except to the extent those have different terms. IF YOU COMPLY WITH THESE LICENSE TERMS, YOU HAVE THE RIGHTS BELOW. 1. INSTALLATION AND USE RIGHTS. You may install and use any number of copies of the software. 2. TERMS FOR SPECIFIC COMPONENTS. a. Microsoft Platforms. The software may include components from Microsoft Windows; Microsoft Windows Server; Microsoft SQL Server; Microsoft Exchange; Microsoft Office; and Microsoft SharePoint. These components are governed by separate agreements and their own product support policies, as described in the Microsoft â€œLicensesâ€ folder accompanying the software, except that, if license terms for those components are also included in the associated installation directory, those license terms control. b. Third Party Components. The software may include third party components with separate legal notices or governed by other agreements, as may be described in the notices file(s) accompanying the software. 3. SCOPE OF LICENSE. The software is licensed, not sold. This agreement only gives you some rights to use the software. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it in certain ways. You may not Â· work around any technical limitations in the software; Â· reverse engineer, decompile or disassemble the software, or otherwise attempt to derive the source code for the software except, and only to the extent required by third party licensing terms governing the use of certain open source components that may be included in the software; Â· remove, minimize, block or modify any notices of Microsoft or its suppliers in the software; Â· use the software in any way that is against the law; Â· share, publish, rent or lease the software; or

Library	License
minizip-ng	Zlib Condition of use and distribution are the same as zlib: This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgement in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.

Library	License
moltenvk	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
nlohmann-json	MIT MIT License Copyright (c) 2013-2022 Niels Lohmann Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
OpenColorIO	BSD-3-Clause Copyright Contributors to the OpenColorIO Project. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
OpenCV	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
OpenEXR	BSD-3-Clause Copyright (c) Contributors to the OpenEXR Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
OpenImageIO	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
OpenSSL	Apache-2.0 Apache License Version 2.0, January 2004 https://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
OpenSubdiv	Apache-2.0 Modified Apache 2.0 License TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for

Library	License
OpenVDB	Apache-2.0 Apache License Version 2.0, January 2004 http://www.apache.org/licenses/ TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types. "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below). "Derivative Works" shall mean any work, whether in Source or

Library	License
Partio	BSD-3-Clause PARTIO SOFTWARE Copyright 2018 Disney Enterprises, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The names "Disney", "Walt Disney Pictures", "Walt Disney Animation Studios" or the names of its contributors may NOT be used to endorse or promote products derived from this software without specific prior written permission from Walt Disney Pictures. Disclaimer: THIS SOFTWARE IS PROVIDED BY WALT DISNEY PICTURES AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND TITLE ARE DISCLAIMED. IN NO EVENT SHALL WALT DISNEY PICTURES, THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND BASED ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Library	License
patchelf	GPL-3.0-only GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble The GNU General Public License is a free, copyleft license for software and other kinds of works. The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too. When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things. To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

Library	License
pcre2	BSD-3-Clause PCRE2 LICENCE ---------- PCRE2 is a library of functions to support regular expressions whose syntax and semantics are as close as possible to those of the Perl 5 language. Releases 10.00 and above of PCRE2 are distributed under the terms of the "BSD" licence, as specified below, with one exemption for certain binary redistributions. The documentation for PCRE2, supplied in the "doc" directory, is distributed under the same terms as the software itself. The data in the testdata directory is not copyrighted and is in the public domain. The basic library functions are written in C and are freestanding. Also included in the distribution is a just-in-time compiler that can be used to optimize pattern matching. This is an optional feature that can be omitted when the library is built. THE BASIC LIBRARY FUNCTIONS ---------- Written by: Philip Hazel Email local part: Philip.Hazel Email domain: gmail.com Retired from University of Cambridge Computing Service, Cambridge, England. Copyright (c) 1997-2022 University of Cambridge All rights reserved. PCRE2 JUST-IN-TIME COMPILATION SUPPORT ---------- Written by: Zoltan Herczeg Email local part: hzmester Email domain: freemail.hu Copyright(c) 2010-2022 Zoltan Herczeg All rights reserved. STACK-LESS JUST-IN-TIME COMPILER

Library	License
PNG	Libpng COPYRIGHT NOTICE, DISCLAIMER, and LICENSE ========== PNG Reference Library License version 2 --- * Copyright (c) 1995-2023 The PNG Reference Library Authors. * Copyright (c) 2018-2023 Cosmin Truta. * Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson. * Copyright (c) 1996-1997 Andreas Dilger. * Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc. The software is supplied "as is", without warranty of any kind, express or implied, including, without limitation, the warranties of merchantability, fitness for a particular purpose, title, and non-infringement. In no event shall the Copyright owners, or anyone distributing the software, be liable for any damages or other liability, whether in contract, tort or otherwise, arising from, out of, or in connection with the software, or the use or other dealings in the software, even if advised of the possibility of such damage. Permission is hereby granted to use, copy, modify, and distribute this software, or portions hereof, for any purpose, without fee, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated, but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This Copyright notice may not be removed or altered from any source or altered source distribution. PNG Reference Library License version 1 (for libpng 0.5 through 1.6.35)

Library	License
PTex	BSD-3-Clause PTEX SOFTWARE Copyright 2014 Disney Enterprises, Inc. All rights reserved Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * The names "Disney", "Walt Disney Pictures", "Walt Disney Animation Studios" or the names of its contributors may NOT be used to endorse or promote products derived from this software without specific prior written permission from Walt Disney Pictures. Disclaimer: THIS SOFTWARE IS PROVIDED BY WALT DISNEY PICTURES AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT AND TITLE ARE DISCLAIMED. IN NO EVENT SHALL WALT DISNEY PICTURES, THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND BASED ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE

Library	License
pybind11	BSD-3-Clause Copyright (c) 2016 Wenzel Jakob <wenzel.jakob@epfl.ch>, All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Please also refer to the file .github/CONTRIBUTING.md, which clarifies licensing of external contributions to this project including patches, pull requests, etc.

Library	License
pyside-shiboken	LGPL-3.0 GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below. An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version". The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are

Library	License
pystring	BSD-3-Clause Copyright (c) 2008-present Contributors to the Pystring project. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
Python	PSF-2.0 A. HISTORY OF THE SOFTWARE ========== Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see https://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others. In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see https://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software. In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations, which became Zope Corporation. In 2001, the Python Software Foundation (PSF, see https://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation was a sponsoring member of the PSF. All Python releases are Open Source (see https://opensource.org for the Open Source Definition). Historically, most, but not all, Python releases have also been GPL-compatible; the table below summarizes the various releases. Release Derived Year Owner GPL- from compatible? (1) 0.9.0 thru 1.2 1991-1995 CWI yes 1.3 thru 1.5.2 1.2 1995-1999 CNRI yes 1.6 1.5.2 2000 CNRI no 2.0 1.6 2000 BeOpen.com no 1.6.1 1.6 2001 CNRI yes (2)

Library	License
Qt	LGPL-2.1-or-later GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below. An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version". The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are

Library	License
Qt for Python (PySide)	LGPL-3.0 GNU LESSER GENERAL PUBLIC LICENSE Version 3, 29 June 2007 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions. As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License. "The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below. An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library. A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version". The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are

Library	License
QtSingleApplication	BSD-3-Clause Copyright (C) 2013 Digia Plc and/or its subsidiary(-ies). All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Library	License
robin-map	MIT MIT License Copyright (c) 2017 Thibaut Goetghebuer-Planchon <tessil@gmx.com> Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
sentry	MIT Copyright (c) 2019 Sentry (https://sentry.io) and individual contributors. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
SQLite	blessing License Information ========== SQLite Is Public Domain ---------- The SQLite source code, including all of the files in the directories listed in the bullets below are [Public Domain](https://sqlite.org/copyright.html). The authors have submitted written affidavits releasing their work to the public for any use. Every byte of the public-domain code can be traced back to the original authors. The files of this repository that are public domain include the following: * All of the primary SQLite source code files found in the [src/ directory] (https://sqlite.org/src/tree/src?type=tree&expand) * All of the test cases and testing code in the [test/ directory] (https://sqlite.org/src/tree/test?type=tree&expand) * All of the SQLite extension source code and test cases in the [ext/ directory] (https://sqlite.org/src/tree/ext?type=tree&expand) * All code that ends up in the "sqlite3.c" and "sqlite3.h" build products that actually implement the SQLite RDBMS. * All of the code used to compile the [command-line interface](https://sqlite.org/cli.html) * All of the code used to build various utility programs such as "sqldiff", "sqlite3_rsync", and "sqlite3_analyzer". The public domain source files usually contain a header comment similar to the following to make it clear that the software is public domain. > ~~~ The author disclaims copyright to this source code. In place of a legal notice, here is a blessing: * May you do good and not evil. * May you find forgiveness for yourself and forgive others. * May you share freely, never taking more than you give. ~~~

Library	License
sse2neon	MIT MIT License Copyright (c) 2015-2024 SSE2NEON Contributors Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
symbolicator	FSL-1.1-MIT # Functional Source License, Version 1.1, MIT Future License ## Abbreviation FSL-1.1-MIT ## Notice Copyright 2018-2025 Functional Software, Inc. dba Sentry ## Terms and Conditions ### Licensor ("We") The party offering the Software under these Terms and Conditions. ### The Software The "Software" is each version of the software that we make available under these Terms and Conditions, as indicated by our inclusion of these Terms and Conditions with the Software. ### License Grant Subject to your compliance with this License Grant and the Patents, Redistribution and Trademark clauses below, we hereby grant you the right to use, copy, modify, create derivative works, publicly perform, publicly display and redistribute the Software for any Permitted Purpose identified below. ### Permitted Purpose A Permitted Purpose is any purpose other than a Competing Use. A Competing Use means making the Software available to others in a commercial product or service that: 1. substitutes for the Software; 2. substitutes for any other product or service we offer using the Software that exists as of the date we make the Software available; or 3. offers the same or substantially similar functionality as the Software. Permitted Purposes specifically include using the Software: 1. for your internal use and access; 2. for non-commercial education; 3. for non-commercial research; and 4. in connection with professional services that you provide to a licensee using the Software in accordance with these Terms and

Library	License
Tcl	TCL This software is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, ActiveState Corporation and other parties. The following terms apply to all files associated with the software unless explicitly disclaimed in individual files. The authors hereby grant permission to use, copy, modify, distribute, and license this software and its documentation for any purpose, provided that existing copyright notices are retained in all copies and that this notice is included verbatim in any distributions. No written agreement, license, or royalty fee is required for any of the authorized uses. Modifications to this software may be copyrighted by their authors and need not follow the licensing terms described here, provided that the new terms are clearly indicated on the first page of each file where they apply. IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT,

Library	License
USD	Apache-2.0 ========== ========== OpenUSD ========== ========== Note: The Tomorrow Open Source Technology License 1.0 differs from the original Apache License 2.0 in the following manner. Section 6 ("Trademarks") is different. TOMORROW OPEN SOURCE TECHNOLOGY LICENSE 1.0 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION 1. Definitions. "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document. "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License. "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License. "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files. "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation,

Library	License
Vulkan	https://vulkan.lunarg.com/license/

Library	License
WIX	Microsoft Reciprocal License Copyright (c) .NET Foundation and contributors. This software is released under the Microsoft Reciprocal License (MS-RL) (the "License"); you may not use the software except in compliance with the License. The text of the Microsoft Reciprocal License (MS-RL) can be found online at: http://opensource.org/licenses/ms-rl Microsoft Reciprocal License (MS-RL) This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software. 1. Definitions The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law. A "contribution" is the original software, or any additions or changes to the software. A "contributor" is any person that distributes its contribution under this license. "Licensed patents" are a contributor's patent claims that read directly on its contribution. 2. Grant of Rights (A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create. (B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software. 3. Conditions and Limitations (A) Reciprocal Grants- For any file you distribute that contains code from the software (in source code or binary format), you must provide recipients the source code to that file along with a copy of this license, which license will govern that file. You may license other files that are entirely your own work and do

Library	License
yaml-cpp	MIT Copyright (c) 2008-2015 Jesse Beder. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Library	License
zlib	Zlib Copyright notice: (C) 1995-2022 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution. Jean-loup Gailly Mark Adler jloup@gzip.orgmadler@alumni.caltech.edu

Library	License
OmniShotCut	MIT License MIT License Copyright (c) 2026 UVA Computer Vision Lab Permission is hereby granted, free of charge, to any person obtaining a copy of this software, model weights, and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

End User License Agreement (EULA)

PLEASE READ THIS EULA CAREFULLY BEFORE ORDERING OR DOWNLOADING OR USING ANY SOFTWARE PRODUCTS OF FOUNDRY. YOUR ATTENTION IS PARTICULARLY DRAWN TO: (A) CLAUSE 8 IN WHICH

SUBSCRIPTION CUSTOMERS (LICENSEES) AGREE TO THE AUTO-RENEWAL OF THEIR LICENSE(S) ON AN ANNUAL BASIS; (B) CLAUSE 15 WHERE WE LIMIT OUR LIABILITY TO USERS OF OUR SOFTWARE PRODUCTS; (C) CLAUSE 19.2 REGARDING THE DATA WE MAY COLLECT AND HOW WE MAY USE IT; AND (D) CLAUSE 19.3 WHERE YOU AUTHORISE FOUNDRY TO USE THE SOFTWARE TO ACCESS AND COLLECT CERTAIN INFORMATION FROM YOUR COMPUTER NETWORKS AND TO TRANSMIT THIS INFORMATION TO FOUNDRY.

IMPORTANT NOTICE TO ALL USERS: BY DOWNLOADING AND/OR INSTALLING AND/OR USING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS EULA, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA DO NOT DOWNLOAD, INSTALL, COPY OR USE THE SOFTWARE.

IMPORTANT NOTICE TO CONSUMERS WHO PURCHASE SOFTWARE PRODUCTS DIRECT FROM FOUNDRY: YOU HAVE THE RIGHT TO CANCEL YOUR CONTRACT AND OBTAIN A FULL REFUND IN ACCORDANCE WITH CLAUSE 9. HOWEVER YOU WILL LOSE THIS RIGHT ONCE YOU INSTALL THE SOFTWARE OR LOGIN TO AN INDIVIDUAL LOGIN OR TEAM LOGIN LICENSE. THIS DOES NOT AFFECT YOUR CONSUMER RIGHTS IN RELATION TO DEFECTIVE PRODUCTS OR SERVICES.

This END USER LICENSE AGREEMENT ("**EULA**") is, in cases where you purchase our product(s) direct from Foundry, incorporated into the agreement between The Foundry Visionmongers Ltd a company registered in England and Wales with company number 4642027 and whose registered office is at Squire Patton Boggs Secretarial Services Limited, Rutland House, 148 Edmund Street, Birmingham, United Kingdom, B3 2JR and whose address for correspondence is 5 Golden Square, London W1F 9HT, ("**Foundry**"), and you, as either an individual or a single company or other legal entity ("**Licensee**") on the terms of which you will purchase the products and services of Foundry (the "**Agreement**"). In cases where you purchase our product(s) from one of our resellers, the use of the term "Agreement" in this EULA refers to the arrangements between Foundry and Licensee on which Licensee is permitted to use Foundry's product(s), including this EULA.

Foundry reserves the right to refuse to grant a License (as defined in clause 1.1) to any Licensee who has failed to pay any sum due either to Foundry or to a reseller of Foundry, in connection with the Agreement, in connection with any other software license, in connection with any other agreed terms between Foundry and the Licensee, to use any Software product(s) of Foundry.

1. GRANT OF LICENSE

1.1 Subject to terms and the scope of the applicable licence model as set out in clause 2, the limitations of clause 3 and all the other terms of the Agreement, Foundry grants to Licensee a limited, non-transferable (subject to clause 2.1(b) below) and non-exclusive license to download, install and use a machine readable, object code version (subject to clauses 3 and 4 below) of the software program(s) purchased by Licensee (the "**Software**") together with, and in accordance with, any accompanying user guide and other documentation (the "**Documentation**"), solely for Licensee's, where the Licensee is a business, own internal purposes or, where the Licensee is a consumer, domestic and private purposes (the "**License**"); provided, however, that

Licensee's right to download, install and use the Software and the Documentation is limited to those rights expressly set out in this EULA.

1.2 Some types of license models set out in clause 2.1 limit the installation and use of the Software to the country in which Licensee is based at the date of purchase (the "**Home Country**"), unless otherwise agreed by Foundry in writing. For the avoidance of doubt, 'use' in a Home Country requires the server, network and user to be based in the Home Country. Notwithstanding such limits, Licensee may still use the Software outside the Home Country if traveling or working outside the Home Country on a temporary basis provided that: a) such use does not exceed 70 days in aggregate in any rolling twelve month period or, in the case of any license which lasts for less than twelve months, does not exceed the number of days representing 20% of the term of the license. Notwithstanding any other terms, our Software may not be accessed or used in countries which are subject to import/export controls or sanctions, as further detailed in clause 23.

1.3 Only to the extent that is proportionate to, and reasonably necessary to support, Licensee's licensed use of the Software in accordance with the Agreement, Licensee may (provided valid license keys or license entitlements have been obtained) install the Software on more than one computer, provided always that Licensee's concurrent use of different installations of the Software does not exceed the number of valid Licenses that Licensee has paid for or licensed (as applicable). Where Licensee has purchased an Enterprise License, Licensee may install the Software on more than one computer, pursuant to the terms of the specific Enterprise License Agreement for that Licensee.

2. LICENSE MODELS

2.1 For each Software product that you purchase from Foundry, the product will be licensed (and not sold) to you on a subscription basis (or a rental basis, if available) in accordance with one or more of the license models set out in this clause 2.1 and clause 2.2, as specified in Foundry's invoice or order confirmation (as applicable), and subject to the other terms and conditions of this EULA. Please note that some licensing models set out below do not apply to certain Software products of Foundry, and certain Software products may be sold on an Enterprise License basis (as set out in the definitions below). Whichever licensing model applies, Licensee shall not, at any one time, use more copies of the Software than the total number of valid licenses purchased by Licensee (other than in relation to Enterprise Licenses, in which case the total number of users shall be in accordance with the terms of the specific Enterprise License Agreement for that Licensee). In addition to the terms expressly stated in this EULA, by using any tool, plug-in, SDK or other functionality in conjunction with the Software which is owned or licensed by any third party (each, a "**Third Party Licensor**"), Licensee agrees to review, comply with, and be legally bound by the licence terms and conditions of that Third Party Licensor.

(a) "**Educational License**" means a License that has been purchased by an educational institution in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's educational program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Educational License that: (i) it is an educational institution that will use the Software only for the purpose of training and instruction, and for no other purpose, (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Educational License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Educational License is a Floating License, Licensee shall use the Software on only one computer at a time.

(b) "**Enterprise License**" means a License that is available for certain software only, relating to large-scale commercial deployments to enable a specific number of users to access the Software within an enterprise environment, with specific additional terms and support arrangements, as agreed separately in an Enterprise License Agreement in writing between Foundry and the Licensee. Use of the Software shall be limited to any site in the Home Country, unless agreed otherwise in the Enterprise License Agreement. Any Enterprise License may also be subject to an additional Enterprise License agreement, as agreed between the Licensee and Foundry (the "**Enterprise License Agreement**").

(c) "**Graduate License**" means a License that has been purchased by a graduate in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's Graduate Licence, Licensee warrants and represents to Foundry as a condition of the Graduate Licence that the Licensee has graduated from a university or tertiary education institute no more than six (6) months prior to the grant of the Graduate Licence. The Licensee may use the Software in a personal capacity only (namely, as a sole trader or freelancer). For the avoidance of doubt, Licensee may not use the Graduate License in connection with any work or employment, whether paid or unpaid, or other such collaborations with studios or similar organisations. Foundry reserves the right to require evidence of graduation by the Licensee. Use of the Software shall be limited to any site in the Home Country.

(d) "**Individual Login License**"

If Licensee purchases an Individual Login License, Licensee warrants and represents that Licensee is a natural person and that only Licensee shall use the Software. Licensee will be issued with log in details and may use the Software on any number of computers (but not simultaneously). Use of the Software shall be limited to any site in the Home Country.

(e) "**Mari indie**"

Variants of Mari with limited functionality as described in the Documentation are available to purchase. If Licensee has purchased such a variant, Licensee warrants and represents to Foundry as a condition of the Agreement that: (i) Licensee is a natural person; or (ii) Licensee is an entity in the direct ownership of a single natural person; (iii) Licensee will only access and/or use one copy of the Software; and (iv) only Licensee will use the Software.

(f) "**Mari Individual Subscription License**"

If Licensee purchases a Mari Individual Subscription License then: (a) Licensee warrants and represents that Licensee is a natural person and that only Licensee will use the Software; (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee may use the Software on different computers which may be located anywhere and use is not restricted to the Home Country; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; (e) Licensee shall not purchase or use more than one Mari Individual Subscription License; and (f) the provisions of clause 8 shall apply.

(g) "**Non-Commercial License**"

If the License is a Non-Commercial License, Licensee warrants and represents that Licensee is a natural person, that they will only access and/or use one copy of a Non-Commercial License for personal, recreational and non-commercial purposes and that only Licensee will use the Software. Under a Non-Commercial License, Licensee will not use the Software: (a) in conjunction with any other copies or versions of the Software, under any type of License model; (b) for any commercial, professional, for-profit and/or on-sale purpose or otherwise to provide any commercial service(s) to a third party (whether or not for financial or other reward and including for education, instruction of or demonstration to any third party for commercial purposes); (c) in the course of any employment or business undertaking of Licensee; (d) on any commercial premises during business hours (except where use of the Software is solely for a personal, recreational, educational or other non-commercial purpose); and/or (e) to create any commercial tools or plug ins.

(h) "**Nuke Indie License**"

If Licensee purchases a License for Nuke Indie, then: (a) Licensee warrants and represents that Licensee (i) is a natural person and that only Licensee will use the Software; (ii) is working independently and shall not use the Nuke Indie License in a pipeline with other Nuke commercial or Nuke Indie licenses, whether those licenses are held by the Licensee, other individuals or other businesses or organisations; and (iii) earns less than $100,000 USD (or local equivalent) a year; and (iv) that Licensee satisfies all criteria set out in Foundry's Nuke Indie Eligibility Requirements as published on its website and which may be amended from time to time (the "**Nuke Indie Eligibility Requirements**"); (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee shall not purchase or use more than one Nuke Indie License; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; and (e) Licensee shall use the Software in accordance with terms of the Nuke Indie Eligibility Requirements, including abiding by any functional restrictions; and (f) the provisions of clause 8 shall apply.

(i) "**Nuke Stage License**"

If Licensee purchases a Nuke Stage License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent Nuke Stage Output Device(s) shall never exceed the total number of valid Nuke Stage Floating Licenses purchased by Licensee; and (b) use of the Software shall

be limited to any site in the Home Country. "Nuke Stage Output Device(s)" shall mean one region designated for rendered output, which may be mapped to one endpoint such as display windows, SDI outputs, files, or other render targets. If Licensee chooses to use the OptiTrack functionality as part of a Nuke Stage Licence, Licensee acknowledges and agrees that its use of this functionality is subject to the terms of OptiTrack's Plugins EULA available at https://optitrack.com/about/legal/eula.html ("**OptiTrack Plugins EULA**"). Licensee agrees to abide by the terms of OptiTrack Plugins EULA. The OptiTrack Plugin EULA applies in addition to any other provisions in this EULA relating to Third Party Licensors. If there is any conflict or inconsistency between this EULA and the OptiTrack Plugin EULA, the OptiTrack Plugin EULA shall take precedence if the conflict or inconsistency relates to the Optitrack Plugin itself.

(j) "**Offline Floating License**"

If Licensee purchases an Offline Floating License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Offline Floating Licenses purchased by Licensee; and (b) use of the Software shall be limited to any site in the Home Country.

(k) "**Offline Node Locked License**"

If Licensee purchases an Offline Node Locked License, Licensee will install and use only a single copy of the Software on only one computer at a time in the Home Country.

(l) "**Rental License**" means a License that has been purchased by a Licensee on a rental basis.

If Licensee has purchased the Software on a rental basis, the License shall be limited to the term of the rental as agreed in writing with Foundry after which it shall automatically expire.

(m) "**Student License**" means a License (whether purchased or provided free of charge) issued to a student in accordance with Foundry's eligibility criteria as published on its website and amended from time to time (at https://www.foundry.com/education/students).

If Licensee has purchased the Software on the discounted terms of Foundry's student program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Student License that: (i) the Licensee is a part-time or full-time student at the time of purchase and will not use the Software for any commercial, professional or for-profit purposes; (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Student License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Student License is a Floating License, Licensee shall use the Software on only one computer at a time.

(n) "**Subscription License**" means a License that has been purchased on a subscription basis, as set out below.

If Licensee has purchased the Software on a subscription basis then: (a) the License shall be limited to the Subscription Period (as defined in clause 8.1) after which it shall automatically expire and; (b) the provisions of clause 8, including but not limited to auto-renewal, shall apply.

(o) "**Team Login License**"

If Licensee purchases a Team Login License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Team Login Licenses purchased by Licensee; (b) use of the Software shall be limited to any site in the Home Country; and (c) use of the Software, and changes to the user authorizations within Licensee's team organization(s), shall be in accordance with any Team Login License guidance and instructions, as published on Foundry's website and which may be amended from time to time.

(p) "**Trial License**"

Licensee may register for a "Trial License" of the Software (not available for all products or in all regions or markets). A Trial License is for product evaluation and learning purposes only and may only be used by a natural person. A Trial License lasts a limited specified period, on the expiry of which the Software will automatically cease to function. Foundry may terminate any Trial License for convenience immediately on notice to Licensee. Licensee will use the Software for product evaluation and learning purposes only.

2.2 If Licensee has purchased a License that permits "non-interactive" use of the Software ("**Headless Rendering**"), Licensee is authorized to use a non-interactive version of the Software for rendering purposes only (i.e. without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Headless Rendering is not available on all products. In all cases, Headless Rendering licenses may be used on any number of computers, provided that the number of concurrent computers shall never exceed the total number of valid Headless Rendering licenses purchased by Licensee.

3. RESTRICTIONS ON USE

Please note that in order to guard against unlicensed use of the Software, a license key is required to access and enable the Software. Licensee is authorised to use the Software in machine readable, object code form only (subject to clause 4), and Licensee shall not: (a) assign, sublicense, sell, distribute, transfer, pledge, lease, rent, lend, share or export the Software (including any tool, plug-in, SDK or other functionality), the Documentation or Licensee's rights under this EULA; (b) alter or circumvent the license keys or other copy protection mechanisms in the Software (including any tool, plug-in, SDK or other functionality) or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software (including any tool, plug-in, SDK or other functionality) in each case except as and to the extent that applicable law requires such reverse engineering, decompilation, disassembly or discovery to be permitted and it is not lawful to contract out of such requirement; (c) implement or use any method or mechanism designed to enable product functionality not available in the Software but available in (i) other

Foundry products; or (ii) other Foundry releases of the same product; (d) (subject to clause 4) modify, adapt, translate or create derivative works based on the Software (including any tool, plug-in, SDK or other functionality) or Documentation; (e) use, or allow the use of, the Software (including any tool, plug-in, SDK or other functionality) or Documentation on any project other than a project produced by Licensee (an "**Authorized Project**") or to provide a service (whether or not any charge is made) to any third party; (f) allow or permit anyone (other than Licensee and Licensee's authorized employees to the extent they are working on an Authorized Project) to use or have access to the Software (including any tool, plug-in, SDK or other functionality) or Documentation; (g) copy or install the Software (including any tool, plug-in, SDK or other functionality) or Documentation other than as expressly provided for in this EULA; or (h) take any action, or fail to take action, that could adversely affect the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of Foundry or any a Third Party Licensor; (i) use the Software, or permit any third party to use the Software, for any illegal purpose. The Licensee shall maintain appropriate internal security, safeguards and controls which shall be at least as great as the measures the Licensee uses to keep Licensee's own information secure (but in any case, using no less than a reasonable degree of care) to ensure that the Software is not used in any way which is in violation of this clause 3 or of this Agreement. For the avoidance of doubt, Licensee remains fully liable for any unauthorized use of the Software by its staff. For purposes of this clause 3, the term "Software" shall include any derivatives of the Software.

Notwithstanding clause 3(b) above, where the reduction of the Software to human readable form is necessary for the purposes of integrating the operation of the Software with the operation of other software or systems used by the Licensee in accordance with section 50B of the Copyright Designs and Patents Act 1988 (or any analogous legislation in other jurisdictions), prior to reducing the Software to human readable form (whether by reverse engineering, decompilation or disassembly), the Licensee shall notify Foundry in advance and allow Foundry a reasonable period to either carry out such action as is required to fulfil the integration or provide the information necessary to achieve such integration to the Licensee, and in either case the Licensee shall meet Foundry's reasonable costs in doing so.

Unless Licensee has purchased an Individual License, a Team Login License or an Individual Login License, for other license types, if the Software is moved from one computer to another (or, in the case of an Offline Floating License, from one license server to another), the issuing of replacement or substituted license keys or approval of the new server location, is in Foundry's sole discretion and is subject to and strictly in accordance with Foundry's License Transfer Policy, which is available on Foundry's website and which requires a fee to be paid in certain circumstances. Foundry may from time to time and at its sole discretion vary the terms and conditions of the License Transfer Policy.

The Software and/or any components thereof are not intended to be used for and Licensee shall not use the Software for (or modify the Software such that is suitable for): (i) prohibited practices under the European Union Artificial Intelligence Act (the "**EU AI Act**") or similar legislation, (ii) any use that would result in the Software being considered a high-risk AI system or that otherwise qualifies as "high-risk" under the EU AI Act. Licensee may not remove, alter or obscure any metadata fields which are automatically applied to content generated by the Software which indicate that such content has been artificially generated or manipulated.

PROHIBITED AND HIGH-RISK USES. Licensee shall not use the Software for: (i) any application that would classify the Software as 'high-risk' under EU AI Act or a similar concept in other applicable regulations; (ii) life-critical systems, medical devices, or safety-critical infrastructure; (iii) autonomous weapons systems or mass surveillance applications; (iv) any illegal, immoral or improper use (including inputting illegal, immoral, or inappropriate content or prompts into any AI models in connection with the Software), or (v) any use that would subject Foundry to additional regulatory obligations. Licensee indemnifies Foundry against all costs and liabilities arising from such prohibited and high-risk uses.

Licensee indemnifies Foundry against regulatory fines, costs or expenses related to Licensee's non-compliant use of the Software.

4. SOURCE CODE

Notwithstanding that clause 1 defines "Software" as an object code version and that clause 3 provides that Licensee may use the Software in object code form only:

4.1 if Foundry has agreed to license to Licensee (including by way of providing SDKs, upgrades, updates or enhancements/customization) source code or elements of the source code of the Software, the intellectual property rights in which belong either to Foundry or to a Third Party Licensor ("**Source Code**"), Licensee shall be licensed to use the Source Code as Software on the terms of this EULA and: (a) notwithstanding clause 3 (c), Licensee may use the Source Code at its own risk in any reasonable way for the limited purpose of enhancing its use of the Software solely for, where the Licensee is a business, its own internal business purposes or, where the Licensee is a consumer, domestic and private purposes and in all respects in accordance with this EULA; (b) Licensee shall in respect of the Source Code comply strictly with all other restrictions applying to its use of the Software under this EULA as well as any other restriction or instruction that is communicated to it by Foundry at any time during the Agreement (whether imposed or requested by Foundry or by any Third Party Licensor);

4.2 to the extent that the Software links to or itself incorporates any open source software and/or software libraries ("**OSS Components**") that are provided to Licensee with or as part of the Software, then where such OSS Components are licensed on the terms of an open source software licence that requires Foundry to make the OSS Components available to the Licensee on specific terms (the "**OSS Licence Terms**"), those OSS Components are licensed to Licensee on, and subject to, the terms of the relevant OSS Licence Terms;

4.3 where Foundry is required by any OSS Licence Terms to make the source code of the relevant OSS Component available to the Licensee, Foundry will at any time during the three year period starting on the date of the Agreement, at the request of Licensee and subject to Licensee paying to Foundry a charge that does not exceed Foundry's costs of doing so, provide Licensee with the source code of the relevant OSS Component (the "**OSS Source Code**") in order that Licensee may modify the OSS Component in accordance with the relevant OSS Licence Terms, together (where appropriate) with certain object code of the Software necessary to enable Licensee to re-link any modified OSS Components to the Software (the "**Object**"); and

4.4 notwithstanding any other term of the Agreement, Foundry gives no express or implied warranty, undertaking or indemnity whatsoever in respect of the Source Code, the OSS Components, the OSS Source Code or the Object, all of which are licensed on an "as is" basis, or in respect of any modification of the Source Code, the OSS Components or the OSS Source Code made by Licensee ("**Modification**"). Licensee may not use the Object for any purpose other than its use of the Software in accordance with this EULA. Notwithstanding any other term of the Agreement, Foundry shall have no obligation to provide support, maintenance, upgrades or updates of or in respect of any of the Source Code, the OSS Components (save for any obligations Foundry may have in respect of any elements that form part of the Software as a whole), the OSS Source Code, the Object or any Modification. Licensee shall indemnify Foundry against all liabilities and expenses (including reasonable legal costs) incurred by Foundry in relation to any claim asserting that any Modification infringes the intellectual property rights of any third party.

5. BACK-UP COPY

Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location within the Home Country that is owned or leased by Licensee in order to provide a back-up in the event of destruction by fire, flood, acts of war, acts of nature, vandalism or other incident. In no event may Licensee use the back-up copy of the Software or Documentation to circumvent the usage or other limitations set forth in this EULA.

6. OWNERSHIP

Licensee acknowledges that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation and all related intellectual property rights and other proprietary rights are and shall remain the sole property of Foundry and the Third Party Licensors. Licensee shall not remove, or allow the removal of, any copyright or other proprietary rights notice included in and on the Software or Documentation or take any other action that could adversely affect the property rights of Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this EULA, Licensee shall reproduce in and on all such copies any copyright and/or other proprietary rights notices provided in and on the materials supplied by Foundry hereunder. Nothing in the Agreement shall be deemed to give Licensee any rights in the trademarks, service marks, patents, trade secrets, confidential information, copyrights or other intellectual property rights of Foundry or any Third Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of Foundry or any Third Party Licensor in Licensee's promotion or publicity without Foundry's prior express written approval.

Subject to clause 4.3, Foundry undertakes (the "**Undertaking**") to defend Licensee or at Foundry's option settle any claim brought against Licensee alleging that Licensee's possession or use of the Software or Documentation in accordance with the Agreement infringes the intellectual property rights of a third party in the same country as Licensee ("**Claim**") and shall reimburse all reasonable losses, damages, costs (including

reasonable legal fees) and expenses incurred by or awarded against Licensee in connection with any such Claim, provided that the Undertaking shall not apply where the Claim in question is attributable to possession or use of the Software or Documentation other than in accordance with the Agreement, or in combination with any hardware, software or service not supplied or specified by Foundry. The Undertaking is conditional on Licensee giving written notice of the Claim to Foundry as soon as reasonably possible, cooperating in the defence of the Claim and not making any admission of liability or taking any step prejudicial to the defence of the Claim. If any Claim is made, or in Foundry's reasonable opinion is likely to be made, against Licensee, Foundry may at its sole option and expense (a) procure for Licensee the right to continue using the Software, (b) modify the Software so that it ceases to be infringing, (c) replace the Software with non-infringing software, or (d) terminate the Agreement immediately by notice in writing to Licensee and refund the License Fee (less a reasonable sum in respect of Licensee's use of the Software to the date of termination) on return of the Software and all copies by Licensee. The Undertaking constitutes Licensee's exclusive remedy and Foundry's only liability in respect of any Claim.

Where the Software integrates with third-party AI models, tools, libraries, or services provided by Third Party Licensor(s), Foundry has not reviewed, approved, confirmed, pre-vetted any such third-party components (including their fitness for a particular purpose), nor does Foundry host, control, or endorse such third-party components. Licensee acknowledges and agrees that use of such components is at the Licensee's own risk and is subject to the Third Party Licensor's terms and conditions, as applicable.

Where the Software enables generation of AI-powered content or workflows ("**AI Outputs**"), ownership of such AI Outputs vests in the Licensee (in so far as applicable law allows), subject to any third-party rights (including any rights of any Third Party Licensors) in the underlying AI models, training data, or other components. Licensee is solely responsible for ensuring that all inputs are legally compliant and appropriate, and for reviewing, validating, and ensuring the accuracy, quality, legality, and appropriateness of all AI Outputs before use or distribution, and Foundry disclaims all warranties and liabilities in relation to AI Outputs (as further set out in clause 13). Licensee shall not input illegal, immoral or inappropriate content to, or use the Software to generate AI Outputs that infringe third-party rights or violate applicable laws.

7. LICENSE FEE

7.1 Licensee acknowledges that the rights granted to Licensee under this EULA are conditional on Licensee's timely payment of the license fee payable to Foundry in connection with the Agreement or, as the case may be, payable to Foundry's reseller (the "**License Fee**"). Except where the applicable invoice expressly sets out payment in instalments, the License Fee shall be payable in full as one single payment.

7.2 Licensee will be charged and agrees to pay to Foundry or Foundry's authorized reseller (as applicable): (a) the License Fee as notified by Foundry (or its reseller) at the time of the initial purchase of the License; and (b) in respect of any Subscription Auto-renewal Period for a Subscription License, the License Fee as notified by Foundry (or its reseller) on or about the applicable Renewal Date.

7.3 Unless stated otherwise, any Licence Fee notified to the Licensee by Foundry (or its reseller) is exclusive of sales tax, VAT and any other similar taxes, duties or levies, which shall be payable by the Licensee (and the Licensee agrees to pay) in addition to the Licence Fee.

7.4 In the cases of Non-Commercial Licenses or Trial Licenses or specific End of Life Software, the fact that no License Fee may be payable shall not be construed as a waiver by Foundry of any right or remedy available to it in relation to any breach by the Licensee of this EULA or the Agreement, or of any other right or remedy arising under applicable law, all of which are expressly reserved.

8. SUBSCRIPTION LICENSES AND AUTO-RENEWAL

8.1 If Licensee has purchased a Subscription Licence, the License shall be limited to the Initial Subscription Period and any/all Auto-renewal Periods (each as defined below) (together the "**Subscription Period**") after which it shall automatically expire.

8.2 The Subscription Licence shall begin as soon as Foundry accepts Licensee's order by issuing Licensee with a license key (the "**Subscription Start Date**") and shall continue for an initial period of twelve (12) months (the "**Initial Subscription Period**") unless earlier terminated in accordance the terms of this EULA, or unless otherwise communicated by Foundry.

8.3 Unless Licensee opts out of auto-renewal in accordance with clause 8.5 then upon the first anniversary of the Subscription Start Date and each subsequent anniversary (each a "**Subscription Renewal Date**"), Licensee's Subscription Licence shall renew automatically for a further twelve (12) months (each an "**Auto-renewal Period**"). Licensee's Subscription License will continue to auto-renew in this manner until Licensee opts out of auto-renewal or unless earlier terminated in accordance with the terms of this EULA.

8.4 Prior to each Subscription Renewal Date, Foundry shall send one (1) email to advise you that your Subscription License is approaching auto-renewal to the contact email address as provided by Licensee in accordance with clause 24. The reminder email will be sent not less than thirty (30) days prior to the relevant Subscription Renewal Date.

8.5 **Opting Out of Auto-renewal and License Expiry.** If Licensee wishes to opt out of auto-renewal then you must email licenses@foundry.com providing details of the Subscription Licence(s) which you wish to opt out not less than seventy-two (72) hours prior to the relevant Subscription Renewal Date. Provided that Licensee notifies Foundry in accordance with the provisions of this clause 8.5 then your Subscription License (s) will not auto-renew and shall expire at the end of the then-current Subscription Period.

8.6 **Increases to the License Fee for Subscription Licenses.** Foundry reserves the right to increase the License Fee for Subscription Licenses from time to time provided that it shall provide Licensee with not less than thirty (30) days' notice of any increase prior to the relevant Subscription Renewal Date.

9. CANCELLATIONS

9.1 Licensee may cancel a License within 14 days of the original purchase date to obtain a full refund and Licensee will no longer be able to use the Software from the cancellation date. Licensee's right to obtain a refund will be lost once the Software has been installed.

9.2 Refunds are not payable for cancellations made after such date. This includes Subscription licenses which are subject to the fixed twelve (12) month terms and for which the Licensee may opt out of auto-renewal in accordance with clause 8.

9.3 Cancellations and requests for refunds can be made by contacting Foundry's Sales Support team at licenses@foundry.com.

10. END OF LIFE

10.1 From time to time, Foundry will retire Software product(s) (and any/ all versions) by withdrawing them from the market. Such Software product(s) will be designated as end of life ("**EoL**") and a corresponding announcement (the "**Announcement**") will be published on Foundry's website from time to time (the date of such announcement being the "**EoL Notification Date**"). The announcement will include an EoL date ("**EoL Date**") and other relevant information.

10.2 No new License(s) (including renewals, if applicable) for the EoL Software product(s) will be sold after the EoL Notification Date. Unless otherwise provided for in the Announcement, the EoL Date will be the last day that the EoL Software product(s) is sold, improved, maintained, updated, or supported (subject to the below).

10.3 Unless otherwise provided for in the Announcement, a Licensee with an active Subscription License(s) at the time of the EoL Date will receive certain limited support until the relevant Subscription Period expires.

10.4 In the event of EoL, Foundry advises Licensee to install the latest product releases, security updates, fixes and patches available until the EoL Date, to remain as secure as possible. Older products may not meet security or performance requirements, and Foundry cannot confirm compatibility with future operating system updates after the EoL Date. To the maximum extent permitted by applicable law, Foundry shall not be liable for any form of loss, damage or disruption in respect of the EoL Software product(s) beyond the EoL Date.

10.5 Foundry retains all rights, including intellectual property rights, associated with the EoL Software product(s) beyond the EoL Date. Any use by Licensee of the EoL Software product(s) beyond the EoL Date will continue to be governed by the terms of this Agreement.

11. MAINTENANCE AND SUPPORT

Subject to Licensee's timely payment of the applicable License Fee, each Subscription Licence (and, if sold by Foundry and applicable, each Rental License) shall include access to certain maintenance and support services in accordance with the Maintenance and Support Terms which are available on Foundry's website (the "**Maintenance and Support Terms**"). The specific maintenance and support services offered or made available by Foundry shall be those set out in the table at clause 1.2 of the Maintenance and Support Terms (the "**Maintenance and Support**").

12. TAXES AND DUTIES

Licensee agrees to pay, and indemnify Foundry from claims for, any local, state or national tax (exclusive of taxes based on net income), duty, tariff or other impost related to or arising from the transaction contemplated by the Agreement.

13. LIMITED WARRANTY

13.1 Subject to clause 13.3, Foundry warrants that, for a period of ninety (90) days after Licensee first downloads the Software ("**Warranty Period**"): (a) the Software will, when properly used on an operating system for which it was designed, perform substantially in accordance with the functions described in the Documentation; and (b) that the Documentation correctly describes the operation of the Software in all material respects. If, within the Warranty Period, Licensee notifies Foundry in writing of any defect or fault in the Software as a result of which it fails to perform substantially in accordance with the Documentation, Foundry will, at its sole option, either repair or replace the Software, provided that Licensee makes available all the information that may be necessary to identify, recreate and remedy the defect or fault. This warranty will not apply to, and Foundry shall have no liability for, any defect or fault caused by: (a) unauthorised use of or any amendment made to the Software by any person other than Foundry; and/or (b) use of the Software in conjunction with third party technology. If Licensee is a consumer, the warranty given in this clause is in addition to Licensee's legal rights in relation to any Software or Documentation that is faulty or not as described.

13.2 Foundry does not warrant that the Software or Documentation will meet Licensee's requirements or that Licensee's use of the Software will be uninterrupted or error free. Where the Software includes AI-enabled features, Foundry makes no warranties, express or implied, regarding the accuracy, quality, originality, merchantability, non-infringement or fitness for purpose of any AI Outputs, and Licensee uses such AI Outputs entirely at its own risk.

13.3 If Licensee purchases a license of the Software that is of a fixed term duration, the Warranty Period in clause 13.1 shall apply only to Licensee's first purchase of such license and not to any subsequent renewal(s) even if a renewal involves another download.

13.4 Where consumers based in the European Union use the Software, the safety of the Software is dependent on Licensee's use in accordance with this Agreement and the Documentation.

14. INDEMNIFICATION

14.1 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use including but not limited to modification of the Software or Documentation by Licensee or any person connected with Licensee or any person provided with the Software by Licensee, infringes the intellectual property rights or other proprietary rights of any third party, including without limitation any claims arising from Licensee's input into, or outputs from any AI-enabled features (including Licensee's creation, use or distribution of any AI Outputs) in connection with the Software.

14.2 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use (including but not limited to modification) of the Software or Documentation, by Licensee, any person connected with Licensee or any person provided with the Software by Licensee, causes any of them or any third party (including but not limited to those viewing a product of the Software) to suffer harm or loss of any kind.

15. LIMITATION OF LIABILITY

15.1 **To Business Users:** This sub-clause applies where Licensee is a business user (and for these purposes any Licensee that is not a consumer shall be treated as a business user). Licensee acknowledges that the Software has not been developed to meet its individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation is supplied only for Licensee's internal use for its business, and not for any re-sale purposes or for the provision of the Software (whether directly or indirectly) to third parties. Foundry shall not under any circumstances whatever be liable to Licensee, its affiliates, officers, directors, shareholders, employees, agents or other representatives, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with the Agreement for loss of profits, sales, business, or revenue, business interruption, loss of anticipated savings, loss or corruption of data or information, loss of business opportunity, goodwill or reputation (in each case whether the loss is direct or indirect) or any indirect or consequential loss or damage. In respect of any other losses, Foundry's maximum aggregate liability under or in connection with the Agreement whether in

contract, tort (including negligence) or otherwise, shall in all circumstances be limited to the greater of US$5,000 (five thousand USD) and a sum equal to the License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law. This EULA sets out the full extent of our obligations and liabilities in respect of the supply of the Software and Documentation. Except as expressly stated in writing in this EULA, there are no conditions, warranties, representations or other terms, express or implied, that are binding on Foundry. Any condition, warranty, representation or other term concerning the supply of the Software and Documentation which might otherwise be implied into, or incorporated in, the Agreement, whether by statute, common law or otherwise, is excluded to the fullest extent permitted by law.

15.2 **To Consumers:** This sub-clause applies where Licensee is a consumer. Licensee acknowledges that the Software has not been developed to meet Licensee's individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation are only supplied for Licensee's domestic and private use. Licensee agrees not to use the Software and Documentation for any commercial, business or re-sale purposes, and Foundry has no liability to Licensee for any loss of profit, loss of business, business interruption, loss of business opportunity or loss or harm caused to property or data used for professional purposes. Foundry is only responsible for loss or damage suffered by Licensee that is a foreseeable result of Foundry's breach of the Agreement or its negligence but Foundry is not responsible for any loss or damage that is not foreseeable. Our maximum aggregate liability under or in connection with the Agreement, whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to a sum equal to the greater of US$5,000 (five thousand USD) and a sum equal to the latest annual License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law.

15.3 Where applicable, Foundry's liability under the EU Product Liability Directive is subject to the following conditions: (a) Licensee must prove that a defect in the Software directly caused the claimed damage; (b) Licensee must demonstrate that the Software was used strictly in accordance with this EULA and Documentation; (c) Liability is excluded where damage results from Licensee's failure to install available updates within 90 days of notification; (d) Liability is excluded where damage results from use of the Software in combination with third-party software, hardware, or services not approved by Foundry; (e) Foundry's maximum aggregate liability under the Directive shall not exceed the greater of €100,000 or three times the total License Fees paid by Licensee in the 12 months preceding the claim; and (f) Claims must be brought within 3 years of discovery of the defect or damage, whichever is earlier.

16. TERM; TERMINATION

16.1 The Agreement is effective upon Licensee's download of the Software, and the Agreement will remain in effect until termination or expiry. Licensee may terminate the Agreement on written notice to Foundry if

Foundry is in material breach of this Agreement and fails to cure the breach within 10 (ten) working days of receiving notice of such breach. If Licensee breaches the Agreement, Foundry may terminate the License immediately by notice to Licensee.

16.2 If the Agreement expires or is terminated, the License will cease immediately and Licensee will immediately cease use of any Software and Documentation and either return to Foundry all copies of the Software and Documentation in Licensee's possession, custody or power or, if Foundry directs in writing, destroy all such copies. In the latter case, if requested by Foundry, Licensee shall provide Foundry with a certificate confirming that such destruction has been completed within 10 days.

16.3 Foundry reserves the right to terminate and/or suspend the License as it deems reasonable in its sole discretion by notice to Licensee if: (a) it becomes aware that Licensee has failed to pay any sum due either to Foundry or to a reseller of Foundry, either in connection with the Agreement, or in connection with any other Software license to use any product(s) of Foundry, and/or in connection with any other agreed terms between Foundry and the Licensee in connection with the Software, and/or in connection with any Maintenance and Support; or (b) the Licensee is otherwise in breach of or fails to comply with any term of the Agreement and/or with any other agreed terms between Foundry and the Licensee.

16.4 Foundry may also terminate this Agreement if Licensee becomes subject to bankruptcy proceedings, becomes insolvent, or makes an arrangement with Licensee's creditors. This Agreement, and for the avoidance of doubt the Licenses to which it relates, will terminate automatically without further notice or action by Foundry if Licensee goes into liquidation. In the event of termination under this sub-clause, Foundry reserves its rights in full to any amounts due and not paid by the Licensee, including any interest payments, and to any other remedies whether under law, contract, statute or otherwise.

17. CONFIDENTIALITY

17.1 Licensee agrees that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation are proprietary to and the confidential information of Foundry or, as the case may be, the Third Party Licensors, and that all such information and any related communications (collectively, "**Confidential Information**") are confidential and a fundamental and important trade secret of Foundry and/or the Third Party Licensors. If Licensee is a business user, Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" such Confidential Information for the purposes of that Authorized Project, and shall advise any recipients of Confidential Information that it is to be used only as expressly authorized in the Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information available to any other of Licensee's employees or to any third parties without the express written consent of Foundry. Licensee agrees to segregate, to the extent it can be reasonably done, the Confidential Information from the confidential information and materials of others in order to prevent commingling. Licensee shall take reasonable security measures, which measures shall be at least as great as the measures Licensee uses to keep Licensee's own confidential information secure (but in any case using no less than a reasonable degree of care), to hold the Software, Documentation and any other Confidential Information in strict

confidence and safe custody. Foundry may request, in which case Licensee agrees to comply with, certain reasonable security measures as part of the use of the Software and Documentation. This clause shall not apply to any information that is in or comes into the public domain (other than as a result of the Licensee's breach of its obligations under the Agreement), or was in Licensee's lawful possession before receipt or which Licensee develops independently and without breach of this clause. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or other equitable relief as may be deemed proper by a court of competent jurisdiction.

17.2 Licensee acknowledges that Foundry may provide Licensee's identity to any Third Party Licensor for legitimate purposes, including, but not limited to, that Third Party Licensor's monitoring of Licensee's use of its intellectual property rights and enforcement of its license terms.

18. FEEDBACK

Licensee may, but is not required to, provide suggestions, comments, ideas, or know-how, in any form, to Foundry's products, services or technology ("**Feedback**"). Any such Feedback shall be owned in full by Foundry and Foundry may use and incorporate such Feedback for any purpose, without payment of royalties or other consideration to Licensee. There shall also be no obligation to provide attribution for use of Feedback.

19. INSPECTION AND INFORMATION

19.1 Licensee shall advise Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit Foundry and/or its authorized agents to audit all such locations during normal business hours and on reasonable advance notice, and in the case of Enterprise Licenses, Licensee shall provide Foundry with accurate records of the total number of users who have accessed or used the Software, at any time upon Foundry or its authorized agent's request.

19.2 The Software may include mechanisms to access and collect limited information from computer(s) on which it is installed and from any IT systems to which those computer(s) may be connected (including any system registry files) and transmit it to Foundry and/or its resellers, including the ability to locally cache such information on such computers. Such information (the "**Information**") may include: details of relevant license(s) to Foundry products, details of computer and network equipment, details of the operating system (s) in use on such computer equipment, email domain relating to owners of such computer and network equipment, the location of the computer(s) on which the Software is installed and the profile and extent of use of the different elements of the Software and other Foundry software. Foundry may use the Information to (a) model the profiles of usage, hardware and operating systems in use collectively across its customer base in order to focus development and support, (b) to provide targeted support to individual customers, (c) to ensure that the usage of the Software by Licensee is in accordance with the Agreement and does not exceed any user number or other limits on its use, (d) to confirm the identity of Licensee, to identify

unlicensed use of the Software (including use of pirated or other unlicensed copies of the Software) and to assist Foundry (and its resellers and any enforcement bodies) in contacting any unlicensed users of the Software and seeking to terminate unlicensed use of the Software, and (e) to advise Licensee about any maintenance and/or service issues such as available upgrades. To the extent that any Information constitutes personal data for the purposes of the General Data Protection Regulation (EU) 2016/679 ("**EU GDPR**") and the version of the EU GDPR retained in UK domestic law as further defined in the Data Protection Act 2018 ("**UK GDPR**"), in each case, as amended, superseded or replaced from time to time (as applicable, "**GDPR**") it shall be processed in accordance with the GDPR and with Foundry's Privacy Notice (seE https://www.foundry.com/privacy-notice), as may be updated by Foundry from time to time. Licensee undertakes to make all of its users of the Software (including its Licensee Representatives, as defined in the Maintenance and Support Terms, as applicable) aware of, and procure they all agree to, the uses which Foundry will make of the Information and of the terms of Foundry's Privacy Policy.

19.3 By downloading or using the Software, you (i) warrant that you are entitled to control access to the computer(s) on which the Software is downloaded and any IT systems to which they may be connected, and (ii) irrevocably authorise Foundry (through the use of the Software) to access such computer(s) and IT systems (including any system registry files) and collect the Information from them and to transmit that Information to Foundry and its resellers (and any enforcement bodies) and use it for the purposes identified at clause 19.2(a) to (e) above.

20. DEFECT NOTIFICATION AND LICENSEE OBLIGATIONS TO CONSUMERS

20.1 Notification Procedures: Foundry will notify its Licensees and other consumer customers, as applicable, of material security vulnerabilities through reasonable means, which may include email, website posting, or in-software notifications. Where the Software is made available to consumers in the European Union, Foundry will comply with notification obligations under the EU General Product Safety Regulation (GPSR) 2023/988, including notifying relevant authorities where required by law. Notification timing is at Foundry's discretion based on patch availability and risk assessment. Foundry has no obligation to provide individual notifications for non-material issues, to be determined in Foundry's sole discretion.

20.2 Licensee Response Obligations: Licensee must implement provided security updates (as described further in clause 1.6.2 of the Maintenance and Support Terms) within 90 days of notification to maintain warranty and support coverage. Licensee must maintain current contact information with Foundry for security notifications. Failure to respond to critical security or safety notifications may result in license suspension. Licensee shall promptly report to Foundry via our Support Portal any safety concerns, incidents, or accidents involving the Software.

20.3 Liability Limitations: Foundry's liability to the Licensee for notification delays is limited to direct damages not exceeding the Licensee's annual (one-year) License Fee. Foundry has no liability for damages resulting from Licensee's failure to act on security notifications.

21. U.S. GOVERNMENT LICENSE RIGHTS

All Software, including all components thereof, and Documentation qualify as "commercial items," as that term is defined at Federal Acquisition Regulation ("**FAR**") (48 C.F.R.) 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in FAR 12.212. Consistent with FAR 12.212 and DoD FAR Supp. 227.7202-1 through 227.7202-4, and notwithstanding any other FAR or other contractual clause to the contrary in any agreement into which this Agreement may be incorporated, a government end user will acquire the Software and Documentation with only those rights set forth in this Agreement. Use of either the Software or Documentation or both constitutes agreement by the government that all Software and Documentation are "commercial computer software" and "commercial computer software documentation," and constitutes acceptance of the rights and restrictions herein. The Software is the subject of the following notices:

* Copyright © 2001 - 2026 The Foundry Visionmongers Ltd. All Rights Reserved.

* Unpublished- rights reserved under the Copyright Laws of the United Kingdom.

22. SURVIVAL

Clause 6, clause 7 and clauses 12 to 27 inclusive shall survive any termination or expiration of the Agreement.

23. IMPORT/EXPORT CONTROLS AND SANCTIONS

To the extent that any Software made available under the Agreement is subject to restrictions upon export and/or re-export from any applicable jurisdiction (including the United States), Licensee agrees to comply with, and not act or fail to act in any way that would violate, applicable international, national, state, regional or local laws and regulations, including, without limitation, the U.S. Export Administration Act and the Export Administration Regulations, the regulations of the U.S. Department of Treasury Office of Foreign Assets Control, the International Traffic in Arms regulations, the United States Foreign Corrupt Practices Act, the UK Export Control Act 2002 and the UK Export Control Order 2008 (collectively, "**Export Laws**") as those laws may be amended or otherwise modified from time to time, and neither Foundry nor Licensee shall be required under the Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations. Without limiting the foregoing, Licensee agrees that it will not sell, supply, transfer, export or re-export, directly or indirectly, Foundry's Software or related products and services, or any commodity, technology, technical data, software or service that incorporates, contains or is a direct product of Foundry's Software, products and/or services, (i) in violation of the Export Laws; (ii) to any country for which an export license or other governmental approval is required at the time of export, without first obtaining all necessary export licenses or other approvals; (iii) to any country, or national or resident of a country, to which trade is embargoed by the United States and/or sanctioned by the United Kingdom or the European Union; (iv) to any person or firm on any government agency's list of blocked, denied or barred

persons or entities, including but not limited to the U.S. Department of Commerce's Denied Persons List and Entities List, and the U.S Treasury Department's Specially Designated Nationals List; or (v) to the Russian Federation and/or Belarus or for use in the Russian Federation and/or Belarus, or (vi) for use in any nuclear, chemical or biological weapons, or missile technology end-use. Licensee warrants and represents that it is not subject to any Export Law, sanction and/or other form of trading compliance restriction or regulation which may limit or prohibit its right to download, install and/or use the Software in accordance with this EULA.

24. MISCELLANEOUS

Unless Licensee is a consumer, the Agreement is the exclusive agreement between the parties concerning its subject matter and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject matter. Licensee acknowledges that Licensee has not relied upon any representation or collateral warranty not recorded in the Agreement inducing it to enter into the Agreement.

The Agreement may be modified only in writing, by Foundry, at any time. Any such modified terms will become effective upon posting a revised version on Foundry's website. It is your responsibility to check the Foundry website regularly for modifications to this Agreement.

Foundry shall not be in breach of this Agreement nor liable for delay in performing, or failure to perform, any of its obligations under this Agreement if such delay or failure result from events, circumstances or causes beyond its reasonable control.

The failure of either party to enforce any rights granted under the Agreement or to take action against the other party in the event of any such breach shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

The Agreement and any dispute or claim arising out of or in connection with it or its subject matter or formation (including, unless Licensee is a consumer, non-contractual disputes or claims) shall be governed by, and construed in accordance with English Law and the parties irrevocably submit to the non-exclusive jurisdiction of the English Courts, subject to any right that a consumer may have to bring proceedings or to have proceedings brought against them in a different jurisdiction.

If Foundry fails to insist that Licensee performs any obligation under the Agreement, or delays in doing so, that will not mean that Foundry has waived its rights or any other right or remedy, nor shall it prevent or restrict the further exercise of that or any other right or remedy. No single or partial exercise of such right or remedy shall prevent or restrict the further exercise of that or any other right or remedy by Foundry.

If any provision or part-provision of the Agreement is or becomes invalid, illegal or unenforceable, it shall be deemed modified to the minimum extent necessary to make it valid, legal and enforceable. If such modification is not possible, the relevant provision or part-provision shall be deemed deleted. Any

modification to or deletion of a provision or part-provision under this clause shall not affect the validity and enforceability of the rest of the Agreement.

Unless Licensee is a consumer, Licensee agrees that Foundry may refer to Licensee as a client or a user of the Software, may display its logo(s) for this purpose and may publish quotations and testimonials from Licensee, its directors, partners, officers or employees. Foundry agrees to promptly cease any such use on Licensee's written request.

Foundry and Licensee intend that each Third Party Licensor may enforce against Licensee under the Contracts (Rights of Third Parties) Act 1999 (the "Act") any obligation owed by Licensee to Foundry under this EULA (or any associated Third Party Licensor terms) that is capable of application to any proprietary or other right of that Third Party Licensor in or in relation to the Software. Foundry and Licensee reserve the right under section 2(3)(a) of the Act to rescind, terminate or vary this EULA without the consent of any Third Party Licensor.

Email Address for General Correspondence: Licensee shall notify Foundry of an email address for the provision of any correspondence in connection with the Agreement and shall notify Foundry via licenses@foundry.com of any change(s) to that email address. Please note, the email address you provide is important for the provision of information and notices to you, including in relation to the auto-renewal of any Subscription License (if applicable). It is your responsibility to provide and maintain an up-to-date email address. Foundry shall store details of and may use the email address to notify you in accordance with the terms of this Agreement.

25. NOTICES

Any legal notices from the Licensee to Foundry should be sent via postal mail or international delivery to The Foundry Visionmongers Ltd, 5 Golden Square, London, W1F 9HT, addressed to The General Counsel. These notices will be effective upon receipt by Foundry. Additionally, a copy of the notice should be emailed to Foundry at legal@foundry.com. Unless otherwise agreed in writing, any notices from Foundry to the Licensee will be provided either (a) by email to the registered email address associated with the Licensee's account, or (b) in any other reasonable manner that involves specific notification to the Licensee. Notices from Foundry to the Licensee will be effective (i) one day after being sent by email, and (ii) five days after being posted or sent by international delivery service. The Licensee agrees that service of process may be effected on the Licensee by registered mail sent to the last known address on record with Foundry, if permitted by applicable law.

26. COMPLAINTS & ONLINE DISPUTE RESOLUTION PLATFORM

We hope that you are satisfied with any Software purchase made or service received from Foundry, but if you have a complaint, in the first instance, please contact us on licenses@foundry.com or through our Support

Portal: https://support.foundry.com/hc/en-us (for technical support and bug reports), or you can request a call back from the Sales team here: https://www.foundry.com/contact-us.

27. COMPLAINTS & ONLINE DISPUTE RESOLUTION PLATFORM

The Agreement and any dispute or claim (including non-contractual disputes or claims) arising out of or in connection with it or its subject matter or formation shall be governed by and construed in accordance with the law of England and Wales and the courts of England and Wales shall have non-exclusive jurisdiction to settle any dispute or claim (including non-contractual disputes or claims) arising out of or in connection with the Agreement or its subject matter or formation.

Last updated March 2026.

Copyright © March 2026 The Foundry Visionmongers Ltd.

All Rights Reserved. Do not duplicate.