



MARI

USER GUIDE

VERSION 1.6v1

Mari™ User Guide. Copyright © 2012 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this 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. ("The Foundry") and/or The Foundry's licensors.

The EULA can be read in the Mari User Guide Appendices.

The 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 The 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 The Foundry and/or The 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 © 2012 The Foundry Visionmongers Ltd. All Rights Reserved.

Mari™ is a 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 the 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.

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 software engineering: Jack Greasley, Kiyoyuki Nakagaki, Marcus Shoo, Kevin Atkinson, Tim Ebling, Jed Soane, Daniel Lond, Robert Fanner, Duncan Hopkins, Mark Final, Chris Bevan, Carl Rand and Phil Hunter

Product testing: Michael Zannetou, Mark Titchener, Robert Elphick and Antoni Kujawa

Writing and layout design: Jack Elder, Jon Hertzog, Eija Närvänen, Charles Quinn and Erica Cargle

Proof reading: Jack Elder and Eija Närvänen

Mari includes Disney technology licensed from Walt Disney Animation Studios.

The Foundry
6th Floor, Communications Building,
48 Leicester Square,
London
WC2H 7LT

Rev: December 7, 2012

Contents

PREFACE	About this manual.	12
	Contact customer support	12
CUSTOMIZING THE VIEW	Display Options.	13
	Lighting.	13
	Perspective Camera Settings	14
	Configuring Navigation.	14
	Steps for Customizing the View in Mari.	14
	Set the Main View Options.	15
	Configure the Lighting	18
	Set the Perspective Camera Details	21
	Configure Navigation	21
USING SHORTCUTS	Shortcuts for Actions	23
	Canceling Actions	23
	Shortcuts for Settings	23
	Steps for Using Shortcuts and Gestures	24
	View the Default Shortcuts and Gestures	24
	Try the Mouse Gestures	25
	Customize General Keyboard Shortcuts	26
	Set Keyboard Shortcuts for Project Items	27
	Use the Pie Selection Control.	28
	Unassign Shortcuts.	29
CONFIGURING BRUSHES	Steps for Configuring Brushes	33
	Open the Brush Editor	34
	Review the Preset Brushes That Come with Mari.	35
	Review Properties You Can Change	35
	Customize a Brush	40
	Export and Import Brushes.	41
	Change Brush Properties “on the Fly”	42
CLONE STAMPING	Choosing a Source Point.	44
	Setting a Start Point.	45
	Image Options.	46
	Color Grading the Image	46

	Steps for Clone Stamping	48
	Set the Clone Stamp Options	48
	Clone stamp	51
	Apply a Color Grade as You Clone	55
PAINTING THROUGH AN IMAGE OR GRADIENT	About Using a Gradient	58
	Steps for Painting Through.	58
	Set the Paint Through Options.	59
	Paint Through an External Image	61
	Clone from a Floating Image.	63
	Using Tint and Stencil Options.	64
	Paint Through a Gradient	65
CONTROLLING THE WAY MARI APPLIES PAINT TO YOUR MODEL	About the Paint Buffer	67
	Boundaries	67
	Depth and Resolution	68
	About Masking	68
	Edge Masks	68
	Channel masks	71
	Ambient Occlusion Mask.	74
	Depth Mask.	75
	Backface Mask	75
	Fractal Noise Mask	76
	About Blend Modes.	76
	Steps for Controlling How Paint Applies to the Model	77
	Configure the Paint Buffer	77
	Erase Paint	82
	Mask What You Can Paint On.	82
	Edge Masking	84
	Channel Masking.	85
	Ambient Occlusion Masking	86
	Depth Masking	87
	Backface Masking	87
	Fractal Noise Masking.	88
	Set the Paint Blending Mode	88
	Paint Blending Modes.	89
	Paint Buffer Symmetry	99
TRANSFORMING AND WARPING PAINT	Transforming the Paint Buffer	102
	Warping Within the Paint Buffer	103
	Pulling Paint Around with Slerp	104

	Pinning and Pulling with Pinup	104
	Steps for Transforming and Warping Paint	105
	Move or Transform Paint	105
	Warp Paint Using a Grid	107
	Pull, Smudge, Grow, and Shrink Paint	110
	Distort the Paint with Pins	112
TOWING PAINT AROUND	Steps for Towing Paint	116
	Select and Tow Paint	117
COPYING AND PASTING PAINT	Copying a Single Patch	119
	Copy Textures Between Channels and Objects	119
	Using Offsets	119
	Steps for Copying and Pasting Paint	120
	Copying and Pasting Single Patches	120
	Copying and Pasting Multiple Patches	121
EDITING PAINT ON CHANNELS	Blur	123
	Pulling Paint into the Buffer	123
	Steps for Editing Paint on Channels	124
	Blur Paint on a Channel	124
VECTOR BRUSH	Vector Inspector	126
	Vector-related Paint Blending Modes	127
	Vector Shaders	128
	Vector Default Shaders	128
	Vector File Space	128
	Steps for Working with Vector Brushing	129
	Painting Vectors	129
	Modifying Vectors with Paint Blending Modes	130
	Inspecting Your Vector Work	130
	Adding Vector Shader Modules	131
	Adding Default Vector Shaders	131
	Changing the File Space	132
EXPORTING AND IMPORTING TEXTURES	Importing Textures When Creating Channels	133
	Steps for Importing and Exporting Textures	133
	Import Existing Textures	134
	Import Individual Files	135
	Import a Sequence of Files	136

	Set Resize Details and Import.	138
	Export Textures	138
	Export Individual Files.	139
	Export a Sequence of Files	140
	Exporting to a Layered .psd File	141
	Set Small Texture Preferences and Export	142
	Export to Maya.	143
STORING RESOURCES		
USING SHELVES	Steps for Using Shelves	145
	Open Shelves	146
	Add a Brush to a Shelf	147
	Add a Color to a Shelf.	148
	Add an Image to a Shelf	148
	Create a Custom Shelf	150
	Modify Custom Shelves and Their Contents.	151
	Import and Export Shelves and Items.	152
	Create a New Shelf Palette.	154
	Add Any Mari Function to a Shelf.	154
SELECTING ITEMS	Selection Tool	156
	Selection Modes	157
	Smart Selection.	158
	Visibility and Locking	158
	Selection and Painting	158
	Saving Selected Areas.	158
	Marquee Select Tool.	161
	Steps for Making Selections	161
	Selecting Objects, Patches, or Faces	161
	Hiding and Locking	164
	Saving Selections on the Model	165
	Selecting Areas Onscreen.	167
	Moving and Distorting Marquee Selections	169
WORKING WITH		
PATCHES	Patch Resolution.	171
	Changing Patches as a Whole.	171
	Saving Images of UV Masks on a Patch.	173
	Bleeding Patch Edges	174
	Linked Patches	175
	Steps for Working with Patches	176
	View and Resize Patches	176
	Resize Patches	177

	Using UDIM Values	178
	Copy and Paste Patches	178
	Extract Patches to the Image Manager	178
	Fill Patches with Color	179
	Flip and Rotate Painting	179
	Mirror Painting	179
	Save Images of the UV Masks	180
	Bleeding Across Patch Edges	180
	Link or Unlink Patches	180
CONFIGURING THE USER INTERFACE	Steps for Customizing Your Workspace	183
	Choose and Organize Palettes	183
	Choose and Organize Toolbars	184
	Create Custom Toolbars	185
	Configure the HUD	187
	Save Your layout	187
	Understanding the Status Bar	188
	Status Bar Icons	188
MANAGING IMAGES IN MARI	Steps for Managing Images	191
	Open the Image Manager Palette	192
	Load and Select Images	192
	Crop and Filter Images	194
	Run Scripts on Images	196
MANAGING COLORS IN MARI	About the ColorSpace Toolbar	198
	Steps for Managing Colors	199
	Pick Colors from Paint on the Model	199
	Select Colors from a Picker or Image	200
	Set How Colors Are Displayed Onscreen	204
CHANNELS	Different Ways to Create Channels	208
	Presets and Templates	208
	Merging and Blend Modes	209
	Alpha and Masks	209
	HDR Channels	210
	Snapshots	210
	Steps for Managing Channels	211
	Create and Delete Individual Channels	211

Create Multiple Channels from Presets	214
Navigate Between Channels	215
Set and Change Channel Properties	216
Take and Manage Channel Snapshots	216
Snapshot Current Channel	216
Snapshot all Channels	217
Use the Snapshots Palette	217
Autosnapshots	219
Lock and Unlock Channels	220
Merge Channels	221
Convert Masks to and from Alpha	223
Convert a Mask to Alpha	223
Convert Alpha Values to a Mask	224

PTEx

Ptex Support in Mari	225
Limitations of Mari's GPU Previews with Ptex	226
Ptex Modelling Requirements in Mari	226
Minimum Face Size	227
Paint Preservation	227
Steps for Using Ptex	228
Create a New Project	228
Set the Ptex face Texture Options	229
Manage Ptex Faces	231
Export a Ptex Channel	232
Import a Ptex Channel	233
Paint Preservation with Multiple Ptex Geometries	235
Shader Baking to a Ptex Channel	236

SHADERS

Available Modules	238
Shader Limits	239
Shading and Channels	240
Ordering Modules	240
Shaders and Lights	241
Cube Mapping	242
Projection Shaders	242
Baking Shaders onto Channels	242
Creating Layered Shaders	243
Vector Brushing	243
Steps for Using Shaders	244
Open the Shaders Palette	244

	Create a Custom Shader	245
	Set the Shader Details	246
	Bake a Shader Component onto a Channel.	250
	Example: Shader Component in a Channel	251
	Create a Layered Shader.	251
	Copy an Existing Shader to Another Object.	252
	Creating Flow Maps on Multiple Objects	253
	Save, Load, and Delete Shaders	254
	Shader Module Types and Examples.	255
FILTERING IMAGES AND PAINT	Steps for Using Filters	283
	Select and Apply Your Filter	283
	Available Filters	285
PROJECTORS	Steps for Using Projectors	295
	Set the View	296
	Create the Projector	296
	Unproject from Your Projector.	299
	Edit the File(s) Externally	300
	Project the File(s) Back onto the Model	301
	Unproject Multiple Channels.	301
	Use the Quick Project Functions	302
	Use the Layered Painting Project Functions.	304
	Project onto a Model Using .fbx Files.	305
	Generate a Turntable from a Projector.	308
	Generating a Render Turntable.	309
	Generating a Diagnostic Turntable.	310
	Save, Load, and Delete Projectors	312
MULTIPLE OBJECTS	Channels and Shaders.	314
	Versioning.	314
	Ambient Occlusion	314
	Object Shadows	314
	Steps for Working with Multiple Objects:.	315
	Adding and Removing Objects	315
	Moving, Viewing, and Locking Objects	317
	Matching Channels Between Objects	317
	Adding and Removing Object Versions.	319
	Calculating Ambient Occlusion	320

	Shadow Settings	320
LAYERS IN MARI	Base Channels	321
	Flattening Layers	321
	Steps for Working with Layers	321
	Create the Shader and Channels	322
	Paint on Your Layers	323
	Flatten Layers	324
PAINTING ANIMATED OBJECTS	Keyframes	325
	Example	326
	Steps for Working with Animated Objects	329
	Loading Animated Objects	329
	Playing Animation	329
	Painting Animated Textures	331
	Setting Keyframes	331
	Painting the Textures	331
	Exporting Animated Textures	331
EXTENDING MARI	About Custom Attributes	333
	About Custom Shortcuts	333
	About Custom Channel Presets	334
	About Custom Channel Templates	335
	About Environment Variables	335
	Steps for Using Custom Attributes	335
	Steps for Customizing Mari	335
	Steps for Setting Environment Variables	336
	Add Custom Attributes to Objects	336
	Remove Custom Attributes from Objects	337
	Review the Custom Shortcut File Format	337
	Text File: <action>=<key(s)>	337
	Using Modifier Keys	338
	Example: DefaultShortcuts.conf	339
	Define Custom Shortcuts	342
	Review the Custom Channel-Preset File Format	343
	XML File: Header, Details, Closing Blocks	343
	Example: DefaultChannels.mtd	345
	Define Custom Channel Presets	349
	Review the Custom Channel Template File Format	349
	Text File: <name>="<format>"	349

	Example File: DefaultTemplates.mtt	350
	Define Custom Channel Templates	350
	Choose an Environment Variable to Set.	351
	Environment Variables That Mari Recognizes	351
	Image Importing	353
	Overriding Mari Configuration File	353
	Set the Environment Variable.	354
	On Linux	354
	On Windows	355
	Check That the Variable Now Exists.	355
	On Linux	355
	On Windows	355
APPENDIX A:	Keyboard Shortcuts	356
SHORTCUTS	Mouse Gestures	360
APPENDIX B:	Frequently Asked Questions.	361
FREQUENTLY ASKED		
QUESTIONS		
APPENDIX C: THIRD	Third Party Licenses	366
PARTY LICENSES		
APPENDIX D: END USER	End User Licencing Agreement (EULA)	387
LICENSE AGREEMENT		



1 PREFACE

Mari is a creative texture-painting tool that can handle extremely complex or texture heavy projects. It was developed at Weta Digital and has been used on films such as *The Adventures of Tintin: The Secret of the Unicorn*, *District 9*, *The Day the Earth Stood Still*, *The Lovely Bones*, and *Avatar*.

The name Mari comes from the Swahili 'Maridadi', meaning 'beautiful' and carrying connotations of 'usefulness'.

About this manual

This manual provides you with the full details of running and using Mari. You should read the *Getting Started Guide* first for installation and basic usage instructions.

The first part of the manual walks you through the various Mari functions. The tutorials provide comprehensive walk-throughs of all Mari's functions, complete with geometries and associated textures. As you follow the steps in these tutorials, you'll gain a good feel for Mari's user interface and workflow. This, in turn, provides a good working knowledge for creating more complex production projects.

For more detailed information on Mari and its functions, see the accompanying *Mari Reference Guide*.

Contact customer support

Should questions arise that this manual fails to address, you can contact Customer Support directly via e-mail at support@thefoundry.co.uk or via telephone to our London office on +44 (0)20 7968 6828 or to our Los Angeles office on (310) 399 4555 during office hours.



2 CUSTOMIZING THE VIEW

In the *Getting Started Guide*, you learned how to control the view in Mari (spinning, panning, rotating, switching to preset views, and so on). You have a choice of how many of these controls work, and how Mari displays the model on screen.

Display Options

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.

Lighting

Mari comes with three default modes for **lighting** the model, and optional shadow control:

- **Flat** — a flat, uniform, directionless light.
- **Basic** — basic diffuse lighting without specular.
- **Full** — full lighting with diffuse and specular lighting.
- **Shadows** - toggles whether shadows are displayed or not.

In **Basic** and **Full** modes, you can set custom lights. This lets you control the intensity of the lights from four directions (right, left, front, and top, relative to the model).

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

Toggling **Shadows** does not enable or disable shadow processing. It simply controls whether the shadows are displayed on the canvas. To enable shadow processing, enable **Depth Projections** in the **Preferences**. For more information on shadows, please refer to [Object Shadows](#) in the Multiple Objects chapter.

Perspective Camera Settings

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 > Display Properties** and set **Camera Mask Opacity** to anything other than 0.*

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.

Steps for Customizing the View in Mari

To set the view in Mari, follow these steps:

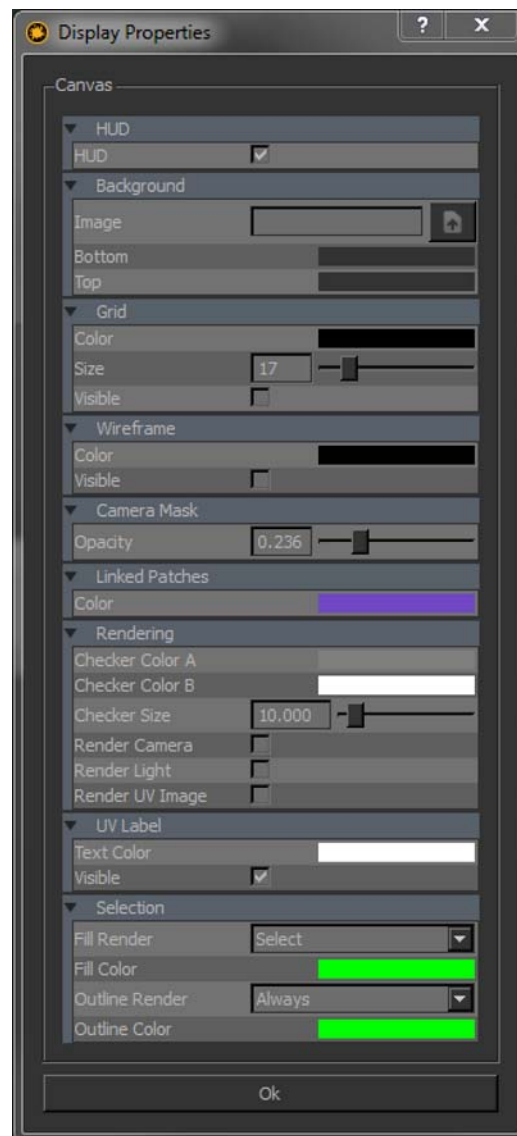
1. [Set the Main View Options](#)
2. [Configure the Lighting](#)
3. [Set the Perspective Camera Details](#)
4. [Configure Navigation](#)

Set the Main View Options

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

1. Switch to the view to set the view options for.
2. Either:
 - right-click on the canvas and select **Display Properties** from the drop-down 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. <i>Use the button next to the field to open a dialog to find your image.</i>
a single color	set the Bottom and Top fields to the same color. <i>Click on the fields to set the color.</i>
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. <i>Click on the fields to set the color.</i>

5. 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.*

6. To show the wireframes on the model, turn the **Visible** option on under **Wireframe**.

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

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

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.

7. If you want to adjust the **Camera Mask Opacity**, use the entry box or the slider to choose a value between 0 and 1.
8. To select the color that you want shown when selecting **Linked Patches**, click on the color swatch to choose from the color picker. You can also choose the **Linked Patch** color from within the **Patches** palette.
9. 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 entry box 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.

10. 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.

11. 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.

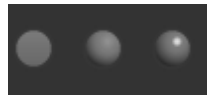
12. Click **OK** to save your settings.

Sandbox Try changing...

- the project background to a different color, then a gradient, then an image
- whether wireframes display, and their color.

Configure the Lighting

To switch between flat, basic, and full lighting, click the toolbar buttons:



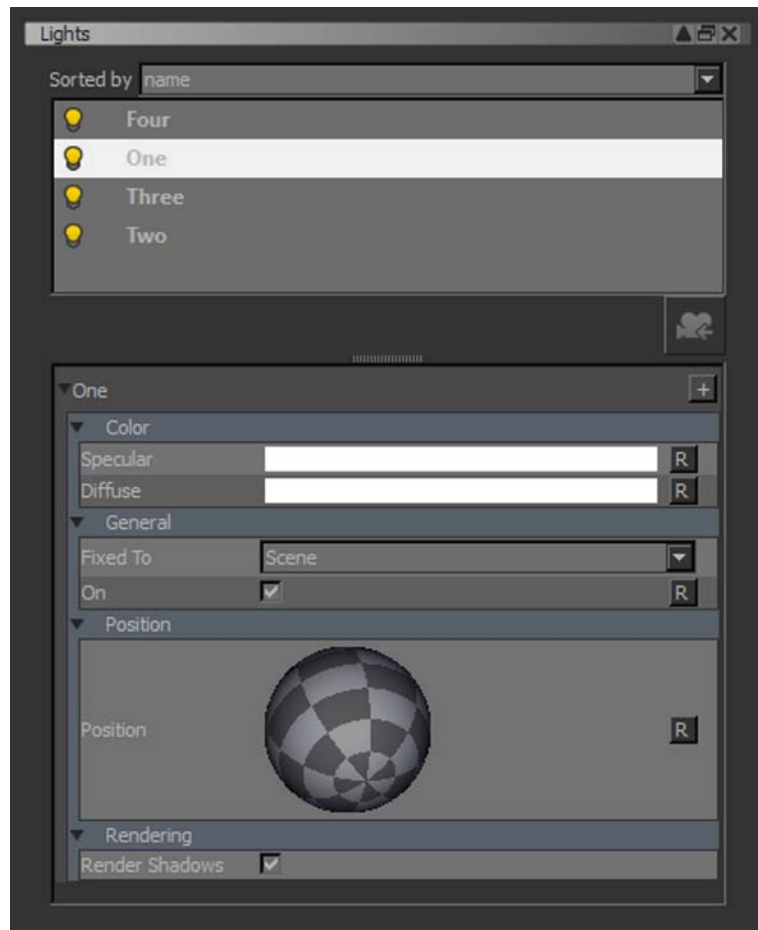
(left to right: flat, basic, full).

1. To choose whether or not to display shadows, toggle the **Shadows** icon in the toolbar:

-  (shadows off)
-  (shadows on)

Note *If the **Depth Projections** option is turned off in the **Preferences**, 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 customizable lights. You cannot add or delete lights. If you don't want a particular light, turn it off.*

3. To turn individual lights on and off, click on the light bulb icon next to the light.
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).
 - **Fixed to** — 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.

5. To move a light around, either:

Select...	then...
The Pan Object tool 	click on the light and drag it around the canvas.
The sphere control in the Lights palette 	click and pull the sphere around to move the light.
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.*

- To choose whether or not you want to cast shadows from the selected light, click on the checkbox to toggle shadows on or off. This does not impact the light itself, but the model casts shadows depending on the light's position.
- More options for shadows and depth projection settings are located in the **Preferences** under **Data > Graphics Card**. The settings in the

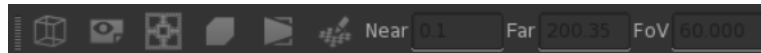
Preferences allows you to trade off the quality of the rendered shadows and the amount of GPU memory and processing required. Increasing the resources available for shadow rendering decreases the resources available for other operations, which may decrease Mari's performance.

Tip *Mari saves custom lights with the project. If you need the same lights in another project, you'll need to set them up separately.*

Sandbox *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?*

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.

Configure Navigation

To configure navigation settings: go to **Edit > Preferences**, and select the **Navigation** tab.

1. To set the controls, select one of the options below, from the **Control Type** dropdown menu:
 - Houdini
 - Lightwave

- Mari
- Maya
- Nuke

These controls allow you to set what type of control options, shortcuts and preferences you want to use.

2. 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 choose how the orbit view is centered: **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 viewport. If disabled, rolling is achieved by simply making a horizontal motion across the viewport.



3 USING SHORTCUTS

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

Shortcuts for Actions

As you'll realize by now, 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 [Extending Mari](#) chapter for details on how to do this.*

Canceling Actions

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
- shader baking
- ambient occlusion
- autosave
- projecting
 - from the paint buffer
 - through projectors
- turntable rendering
- paint projection.

Shortcuts for Settings

Mari includes a configurable onscreen "pie" menu control 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 eraser tools.

Steps for Using Shortcuts and Gestures

To use and customize Mari keyboard shortcuts and mouse gestures, follow these steps:

1. [View the Default Shortcuts and Gestures](#)
2. [Try the Mouse Gestures](#)
3. [Customize General Keyboard Shortcuts](#)
4. [Set Keyboard Shortcuts for Project Items](#)
5. [Use the Pie Selection Control](#)
6. [Unassign Shortcuts](#)

View the Default Shortcuts and Gestures

The list of shortcuts and mouse gestures is in [Appendix A: Shortcuts](#).

- Tip**
- *The appendix is formatted so you can print out the pages and use them as a reference card.*
 - *For the keyboard shortcuts, a plus sign (+) means to hold down two keys at the same time. For example, “**Ctrl+C**” means: press **Ctrl**, 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.*

Sandbox *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.*

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, in the Mari **Preferences** dialog box (from the **Edit** menu, select **Preferences**), and on the **Misc.** tab, turn off **Toggle On Key Held**.*

Try 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, tracing an “E” to select the **Erase** tool).

Sandbox *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.*

Customize General Keyboard Shortcuts

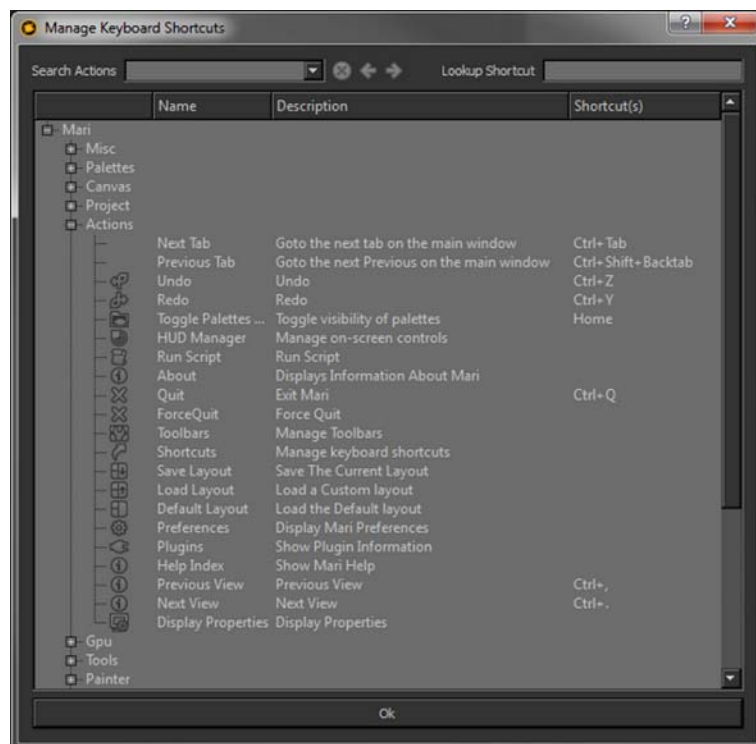
To add and change the general keyboard shortcuts:

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

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

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+R**+left-click can't be set if a shortcut already exists with the definition **Ctrl+R**.*

4. If there are no conflicts, press **OK** (otherwise select a different key).

Sandbox *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.*

Set Keyboard Shortcuts for Project Items

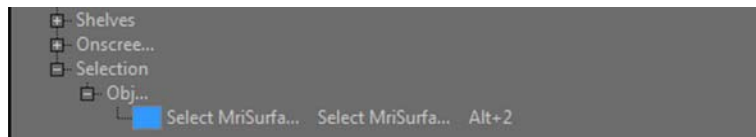
You can set keyboard shortcuts for:

- channels
- patches
- shaders
- projectors
- brushes
- objects
- sets of any of the above.

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.*



This works exactly as described in [Customize General Keyboard Shortcuts](#).

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

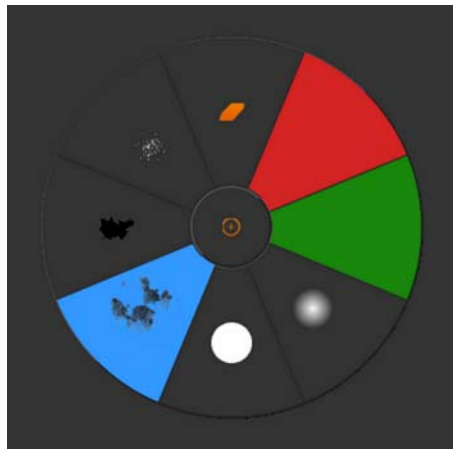


2. To remove the shortcut from the item, right-click on it again and select **Assign Shortcut** from the dropdown menu.

*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.*

Use the Pie Selection Control

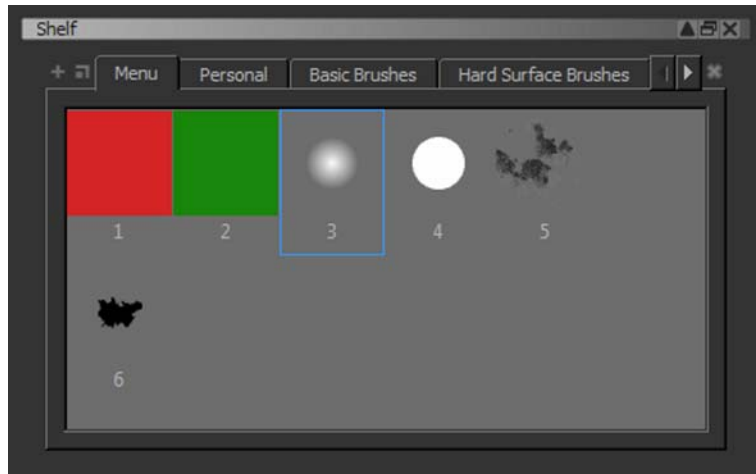
1. With the mouse over the canvas, press **F9**.
The pie selection menu displays:



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](#).

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

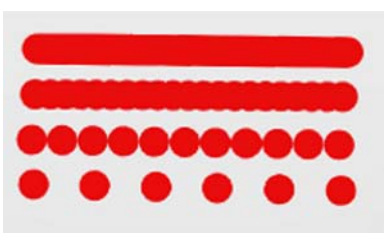
4 CONFIGURING BRUSHES

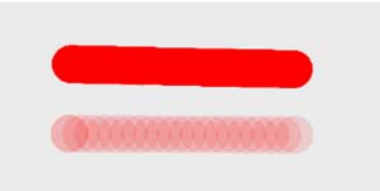
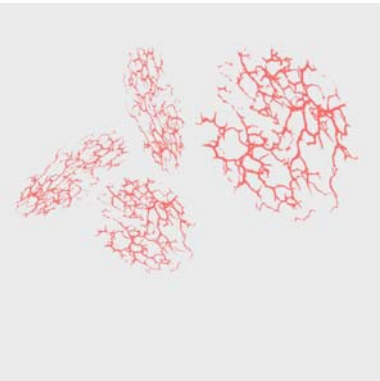
The Mari **Brush Editor** palette lets you select and customize a wide array of preset brushes, with a wide variety of brush properties you can edit. You can then drag modified brushes to a shelf for your ongoing personal use, or for a particular project only. You can also export and import brushes to share with other Mari users, and you can use keyboard-and-mouse shortcuts to modify your brush tips “on the fly”.

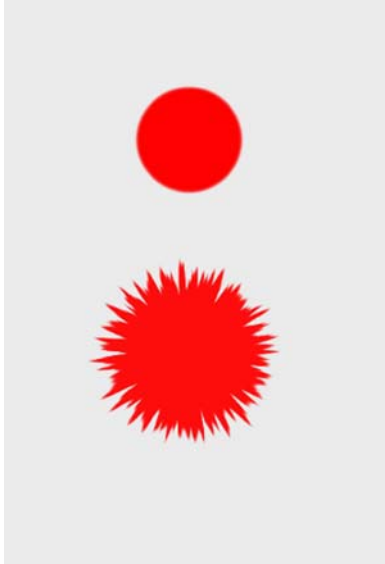
Brush properties generally apply to a **splat** (also sometimes called a “stamp”) — one application of the brush tip onto the canvas. A **stroke** is one continuous application of splats.

For some properties you can specify **jitter** — a random variation, to approximate how an actual (as opposed to digital) paintbrush might work.

Below are brief explanations and examples of some of the properties Mari lets you modify:

Property	Explanation	Example
spacing	the distance between splats in a stroke, as a proportion of splat size	a stroke from the same brush with spacing set to .02, .25, .50, and 1.0: 
opacity	how dark a stroke can be (maximum alpha)	a stroke from the same brush with opacity set to 1.00 and 0.50 with no jitter, and 1.00 with maximum jitter: 

Property	Explanation	Example
flow	how quickly paint flows out of the brush, mimicked digitally by how dark a splat can be (maximum opacity)	a completely hard brush tip, full opacity, with spacing of 0.15 and flow at 1.00 and 0.30: 
radius	how big the brush tip is; you can also specify an inner radius and link pressure to vary between the two, and apply jitter to randomly vary between the two	a stroke with and without radius jitter, where the jitter is equal to the whole radius of the brush: 
distortions to the shape of the brush tip	• rotation — spinning it. • shear — pulling opposite corners vertically or horizontally. • scale — sizing vertically or horizontally. • squish — like vertically scaling down, but affects the radius rather than brush tip.	a splat squished, rotated, sheared, and scaled: 

Property	Explanation	Example
noise	how “spikey” the brush is, as measured in: • octaves — 1 to 8 applications of noise. • frequency — 0 to 100, how many spikes per octave. • scale — 0.00 to 1.00, how big the spikes are, as a proportion of the radius (1.00 = twice as big as the radius). • segments — 4 (a square) to 360 (a circle) lines making up the brush outline (to which noise is applied). Tip: Noise is very hard to explain in words! (Developers understand it in mathematical equations.) Your best bet is to experiment with the settings, to get the right spikiness for your purposes!	the same brush with the default and maximum noise: 

Property	Explanation	Example
profile	opacity from the center to the edge of a brush, determines whether it is hard (more opaque throughout) or soft (fuzzier at the edges)	hard and soft brush tip profiles: 
pressure	whether pressure when painting on your graphics tablet increases the radius, flow or opacity (as calibrated for your tablet)	

Steps for Configuring Brushes

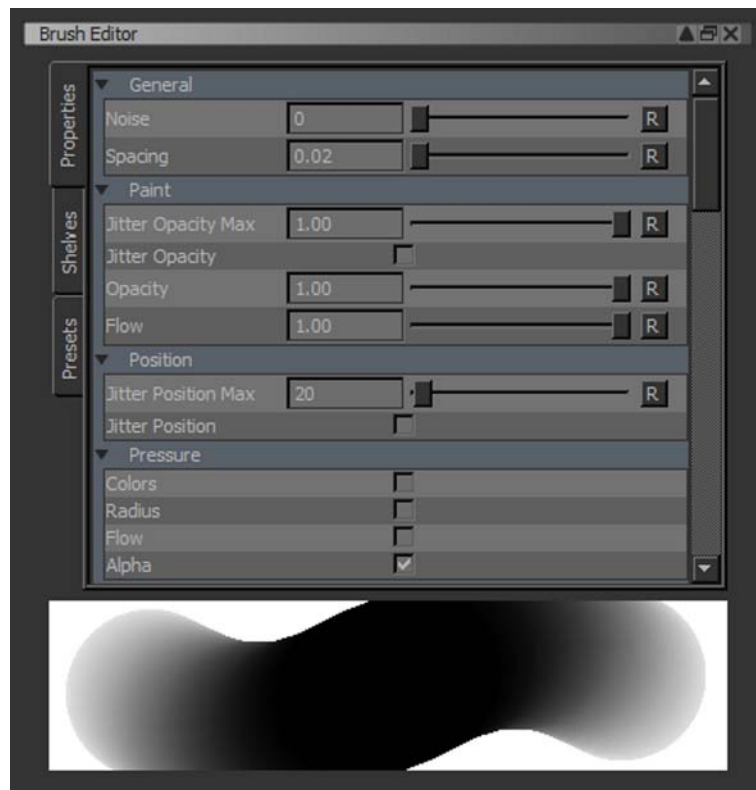
To configure your brushes in Mari, follow these steps:

1. [Open the Brush Editor](#)
2. [Review the Preset Brushes That Come with Mari](#)
3. [Review Properties You Can Change](#)
4. [Customize a Brush](#)
5. [Export and Import Brushes](#)
6. [Change Brush Properties “on the Fly”](#)

Open the Brush Editor

1. Start Mari, and either create a new project or open an existing one.
2. Open the **Brush Editor** palette:
 - from the **View** menu, select **Palettes > Brush Editor**, or
 - right-click in the toolbar area and select **Brush Editor** from the drop-down menu.

The **Brush Editor** palette displays.



The **Brush Editor** has three tabs:

- **Properties** — configurable properties of the selected brush.
- **Shelves** — the same shelves that display in your Shelf palette.
- **Presets** — an array of predefined brushes that come with Mari.

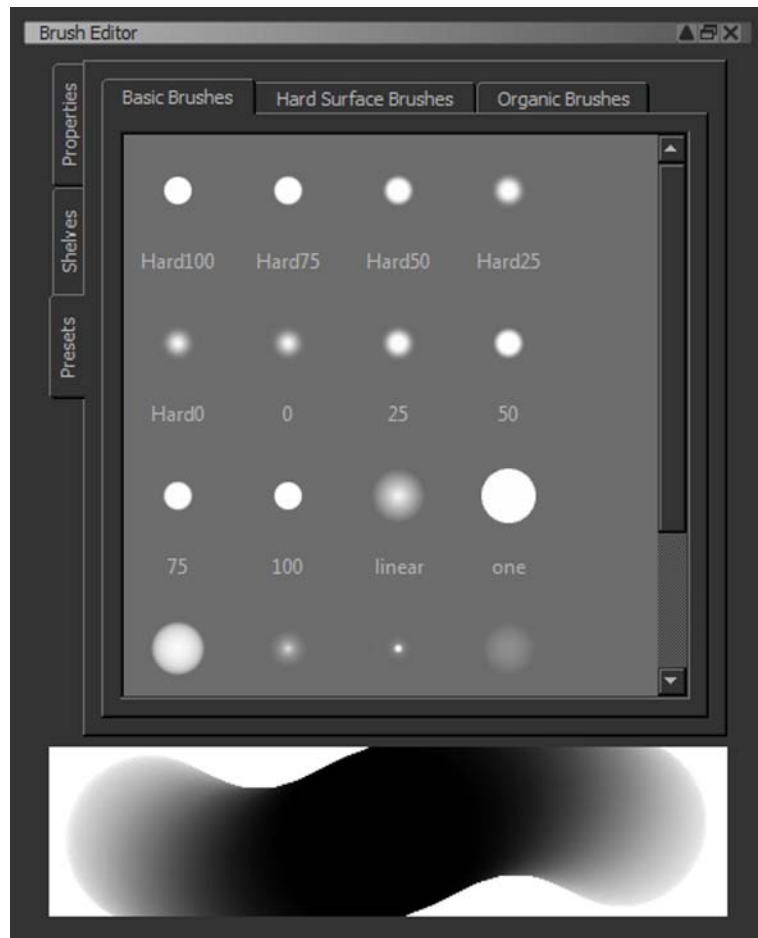
The **Brush Editor** also includes a “scratch pad” at the bottom. The scratch pad shows a squiggle painted with the selected brush. You can also paint directly in the scratch pad (the default squiggle disappears while you enter a stroke).

Review the Preset Brushes That Come with Mari

Mari comes with a set of preset brushes.

1. Click the **Presets** tab.

The preset brushes display and are collected into tabs.



2. Click through each tab and familiarize yourself with the different brushes that are available. Try selecting some brushes and painting in the scratch pad with them.

Review Properties You Can Change

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

1. To view the properties of the selected brush, click the **Properties** tab.

The following properties display. (Below is an extract from the Mari Reference Guide explaining each property.)

Property	Explanation
GENERAL	
Noise	How much noise to add to each splat, to soften the appearance and reduce 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
PAINT	
Jitter Opacity Max	How much to randomly vary the opacity (if Jitter Opacity is enabled). Range: 0.00 to 1.00, Reset: 0.00
Jitter Opacity	Whether to randomly vary the opacity (to the degree specified in Jitter Opacity Max).
Opacity	How dark a stroke can be (maximum alpha). Range: 0.01 to 1.00, Reset: 1.00
Flow	Mimics how quickly paint is applied, by setting the maximum opacity in a splat. Range: 0.01 to 1.00, Reset: 1.00
POSITION	
Jitter Position Max	How many pixels at most to randomly vary displacement from the line of the brush stroke (if Jitter Position is enabled). Range: 0 to 1000, Reset: 20
Jitter Position	Whether to randomly vary displacement from the line of the brush stroke (to the degree specified in Jitter Position Max).
PRESSURE	
Whether pressure on the graphics tablet affects properties of a splat (as calibrated for your tablet)	
Colors	Whether more pressure increases the color strength.
Radius	Whether more pressure increases the radius.
Flow	Whether more pressure increases the flow.
Opacity	Whether more pressure increases the opacity.

Property	Explanation
RADIUS	
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. Range: 0.05 to 1.00, Reset: 1.00
Jitter Radius	Whether to randomly vary the radius (to the degree specified for the Radius and Inner Radius).
Inner Radius	How few pixels the radius of a splat can be (if Jitter Radius and/or Radius Pressure are enabled). Range: 1 to 1200, Reset: 1
Radius	How many pixels at most the radius of splat can be (if Jitter Radius and/or Radius Pressure are enabled). Range: 1 to 1200, Reset: 100
ROTATION	
Align to Stroke	Whether the brush tip rotates along with the direction you're painting. (With calligraphy, for example, the tip does <i>not</i> align.)
Jitter Rotation Max	How many degrees at most to randomly vary the rotation (if Jitter Rotation is enabled). Range: 1 to 360, Reset: 360
Jitter Rotation	Whether to randomly vary the 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
GENERAL	
Type	Whether to use a Bitmap image for the brush tip, or Render based on the qualities specified in these properties
preview	An image of the brush tip, based on the selected properties.
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 that .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. Note that to use an image as the brush tip, you need to set Type to Bitmap .

Property	Explanation
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. 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	
Noise is very hard to explain in words! (Developers understand it in mathematical equations.) Your best bet is to experiment with the settings, to get the right spikiness for your purposes!	
Octaves	number of applications of noise. Range: 1 to 8, Reset: 4
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 Type above), alpha profile of the brush tip from center to edge — determines how “hard” or “soft” it is.
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 .

2. Study the descriptions (above) of the different brush properties.
3. Note the following types of controls used to specify brush properties:
 - **Entry boxes** — to type an entry (usually a number).

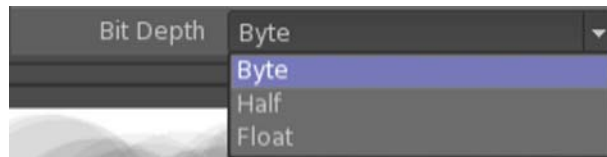
- **Sliders** — to slide between the minimum and maximum allowable entries.
- **Reset button** — to return to the default setting.
(Often entry boxes, sliders and reset buttons are combined.)



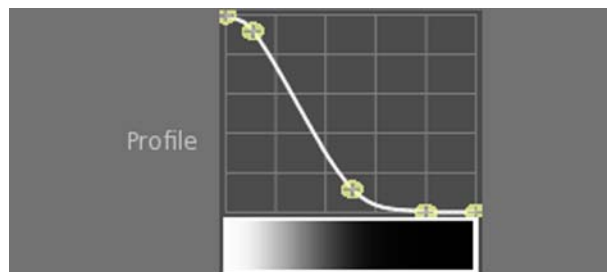
- **Checkboxes** — to select or deselect a property.



- **Dropdown menus** — to select from available options.

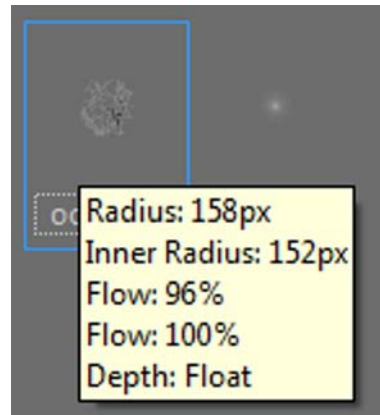


- **Spline Editors** — to move values along a curve; you can drag any of the points on the curve in any direction, and see the effects in the preview bar below the curve. Click to add a new point, and right-click to invert, select presets, and export and import settings. To remove a point, click on it while holding **Ctrl**.



4. Try changing some brush properties, and note the change in the squiggle in the scratch pad (it updates dynamically). You can also try painting in the scratch pad, or painting on the canvas (and then undo-ing) to see the effect of your changes.

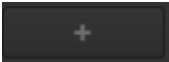
Tip When you hover the mouse over a brush, tooltips display key properties:



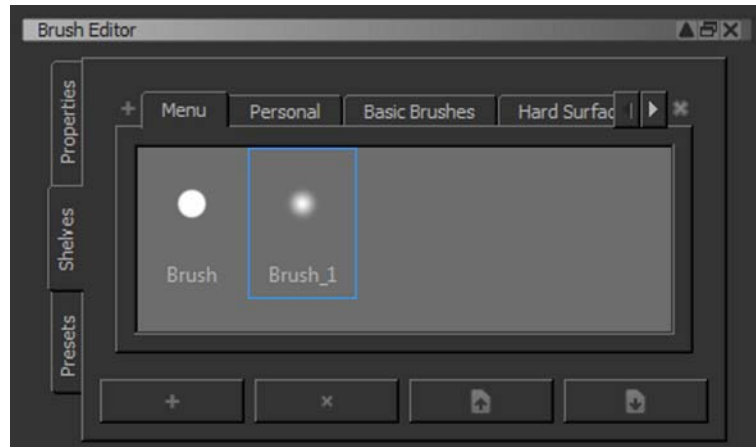
Sandbox *Select one of the preset brushes, and play around with the **Radius**, **Opacity**, **Profile**, and **Jitter** settings. Notice how each affects your brush strokes on the canvas or in the **Brush Editor** scratch pad.*

Customize a Brush

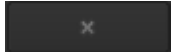
Having looked at the available preset brushes, and different properties you can change, let's (finally) try customizing a brush. This involves three steps: selecting a preset, changing its properties, and saving it to a shelf.

1. First select a preset to modify: click the **Presets** tab, then click a tab for a category of preset brushes, and click to select a brush.
2. [Review Properties You Can Change](#): click the **Properties** tab, and change the brush properties.
3. Finally, save it to a shelf: click the **Shelf** tab; click to select the **Personal** shelf, **Project** shelf, or any other custom shelf you may have created, and click the  button.

The modified brush displays in the selected shelf. By default it is named "Brush" (and then "Brush_1", "Brush_2" etc., incremented for every tool you add).



4. To rename the brush, double-click its name, type over a new name, and press **Enter**.

Tip To delete a custom brush: click to select it, then click .

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.

Sandbox 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.

Export and Import 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, right-click it 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](#)).*

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

Import a custom Photoshop 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 *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.*

Change 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 move the mouse right or left 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 move the mouse right or left to rotate clockwise or counterclockwise.
3. To change the **OPACITY** of your brush, press and hold **O**, and move the mouse right or left to increase or decrease the opacity.

*The alpha slider in the **Colors Palette** moves along with your mouse.*

4. To change the **SQUASH** of your brush, press **Q** and move the mouse right or left to squish the brush tip more or less.

Tip *To save any changes you make on the fly, see the procedure to [Customize a Brush](#).*

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



5 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 rotate the model, the source point remains in the same position in the window, but not the same position on the model.

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.

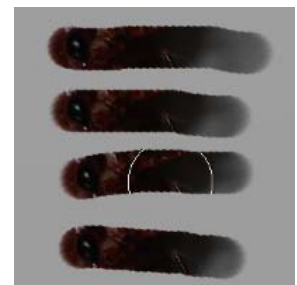
This example shows the difference between the two modes:



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.

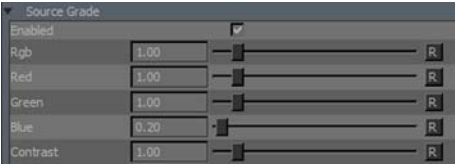
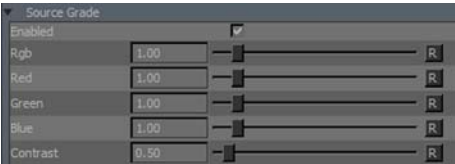
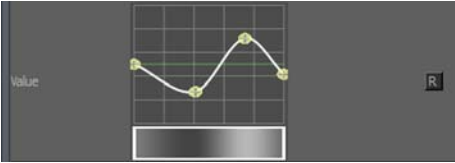
Tip *Mari copies textures you clone from an external image at the resolution of the source. Mari then paints these into the paint buffer at the buffer's resolution. For example, a 4k image provides 4k source texture for cloning, which Mari can then paint into the paint buffer at a maximum of 16k (the maximum size of the paint buffer). See [Controlling the Way Mari Applies Paint to Your Model](#) for details on paint buffer resolution.*

Color Grading the Image

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 channel turned down: 	
Colors at default settings, but contrast turned down: 	
Color and contrast at defaults, but the following value curve applied: 	

Steps for Clone Stamping

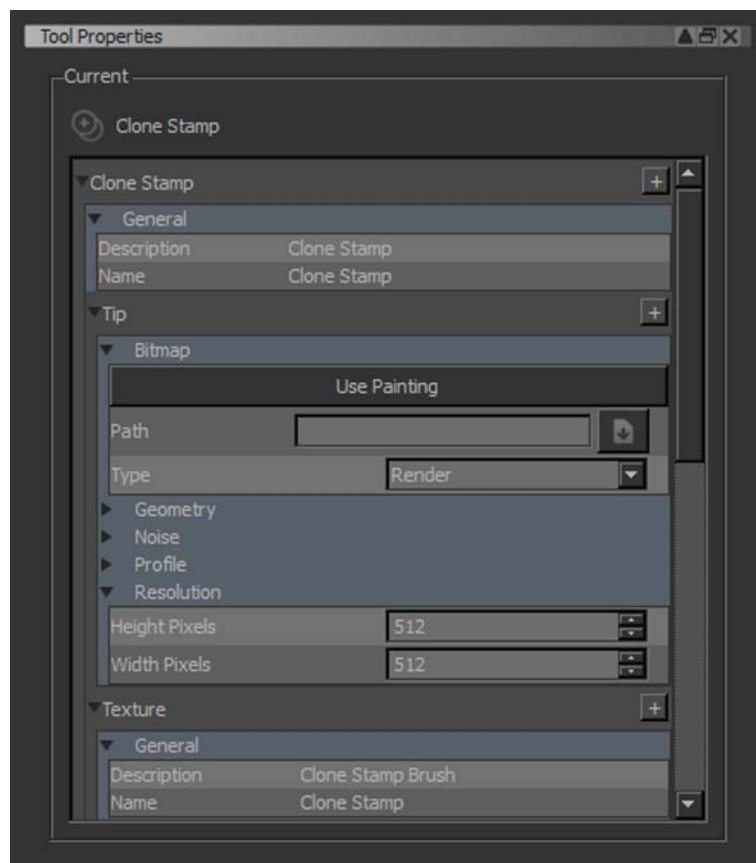
To paint using the clone tools, follow these steps:

1. [Set the Clone Stamp Options](#)
2. [Clone stamp](#)
3. [Apply a Color Grade as You Clone](#)

Set the Clone Stamp Options

1. Start Mari and open your project.
2. Click to select the **Clone Stamp**  tool.
3. Open the **Tool Properties** palette.

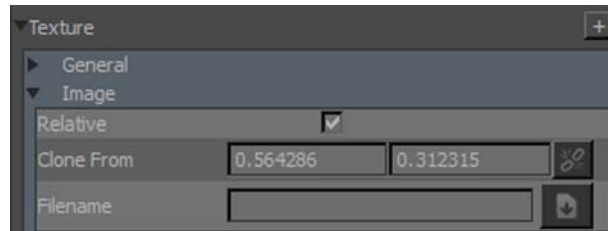
The current properties of the selected tool display. Scroll down to view them all.



4. To customize the brush you're using for your selected tool, set the options under the **Tip** and **Geometry** sections.

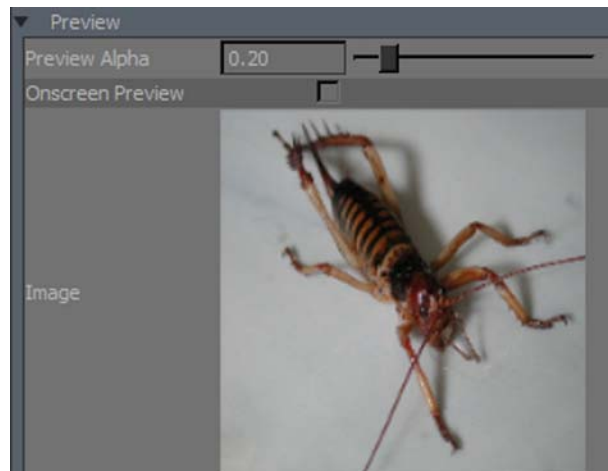
These are the same options available for brushes in general — see [Configuring Brushes](#).

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 pre-view 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

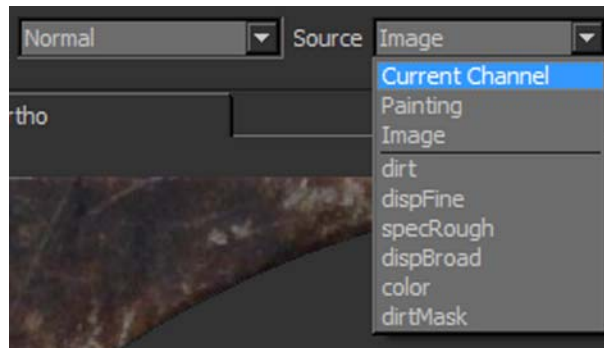
textures, as it saves you having to resize the source image preview from the huge default every time you switch.

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 channel:

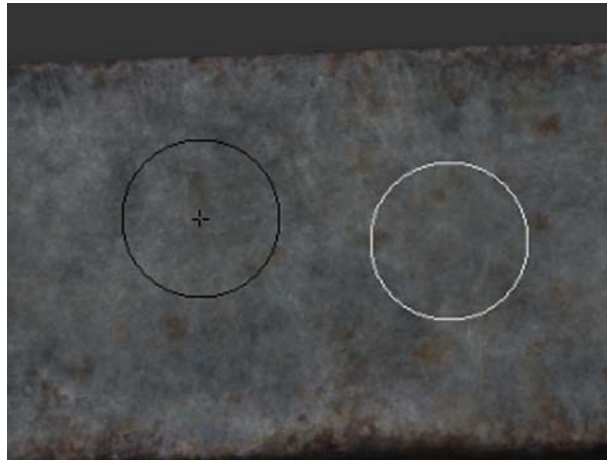
1. If you haven't already, select the **Clone Stamp**  tool.
2. In the **Tool Properties** toolbar at the top of the screen, select the clone source from the **Source** dropdown.



*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 Channel**.*

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 the channel you want to clone from.
4. Hold down **Ctrl** and click on the model's surface to set the source point.
Mari highlights the source point with a cross.



5. 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 blue 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**-click to set the source point. Paint normally:
 - As above, the white circle is your brush.
 - The green 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:

To clone from the surface straight up into the paint buffer directly over the clone point, select the channel to clone from as normal. When you set the source point, hold down **Ctrl+Meta** (the **Windows** key) and click

on the model. Paint normally. As you paint, Mari copies the area you paint from the channel surface into the paint buffer.

If you are cloning from the current channel, 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 channel, or to clone sections of paint from one channel 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+Alt**, clicking and dragging.*

Tip *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.

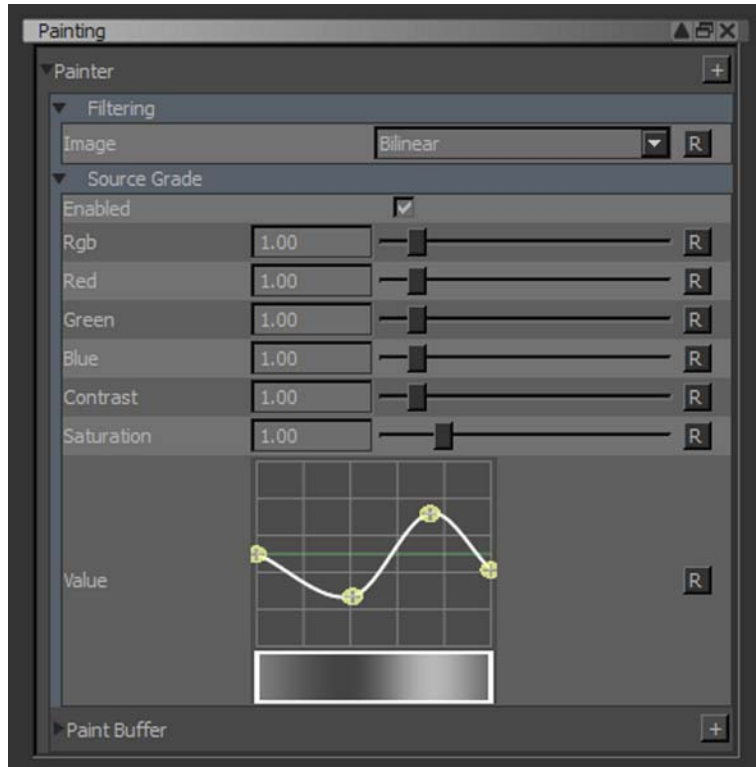
Tip *If you paint off the edge of the image, no clone data is available. You can choose to tile the source image. If this is on, when you paint off the edge, Mari repeats the image (so you always have source data).*

Use the ; (semicolon) key to toggle source image tiling on and off.

Sandbox *Experiment with the various options and modes of clone stamp. Try using a custom brush, with an unusual texture, lower alpha, and so on. Which settings work best for you?*

Apply a Color Grade as You Clone

1. Open the **Painting** palette and scroll down to the **Source Grade** section.



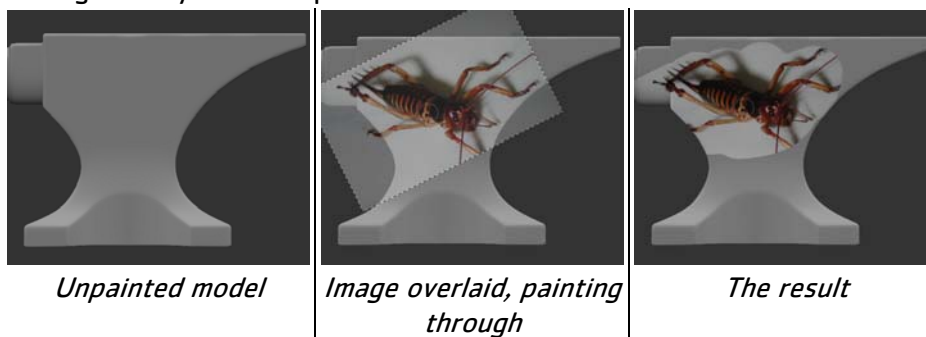
This section lets you control the color grade for your source image.

2. Click to check the **Enabled** box.
3. You can use the controls to:
 - 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.
4. Select the clone tool.
5. Paint some sample strokes, experimenting with the settings in the **Source Grade** section.

Sandbox *Use the sliders to change the color balance in the red, green, and blue channels. 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.*

6 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.



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.

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.

Tip *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:*

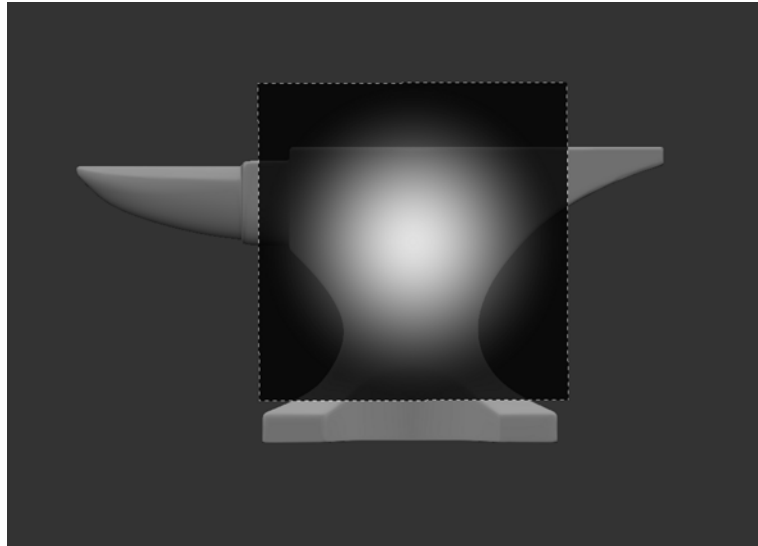
- **Nearest** — *preserves edge detail, but gives quite "blocky" textures.*
- **Linear** — *gives good results, but can produce square artifacts at extreme zoom. This is the default setting.*
- **Bicubic** — *provides more rounded results, slightly blurrier but without the square artifacts.*

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.*

About Using a Gradient

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:



Steps for Painting Through

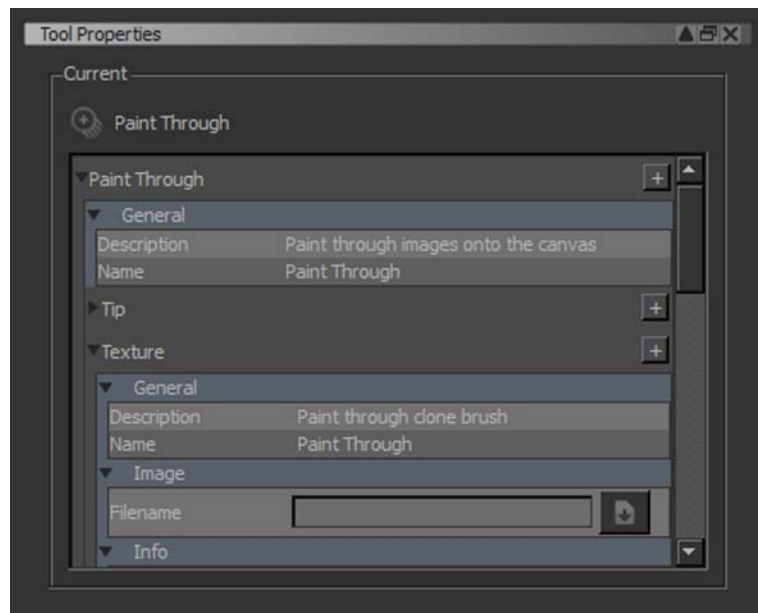
To paint through an image or color gradient in Mari, follow these steps:

1. [Set the Paint Through Options](#)
2. [Paint Through an External Image](#)
3. [Clone from a Floating Image](#)
4. [Using Tint and Stencil Options](#)
5. [Paint Through a Gradient](#)

Set the Paint Through Options

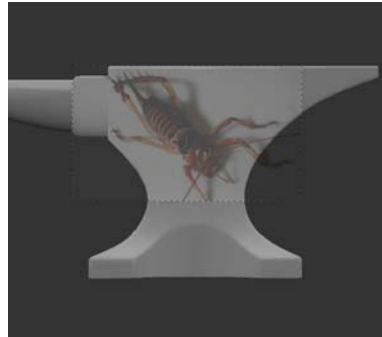
1. Start Mari and open your project.
2. Click to select the **Paint Through**  tool.
3. Open the **Tool Properties** palette.

*The current properties for **Paint Through** display. Scroll down to view them all.*

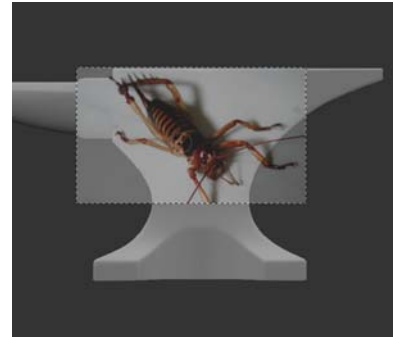


4. To customize the brush you're using for **Paint Through**, set the options under the **Tip** and **Geometry** sections.
These are the same options available for brushes in general — see [Configuring Brushes](#).
5. To change the prominence of the onscreen image that you're painting through, under **Texture** > **Preview**, set the **Preview Alpha**.

Higher values make the preview clearer, but make it harder to see the model underneath.



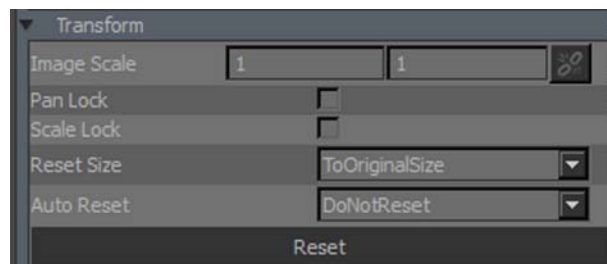
Preview alpha of 0.35



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.

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.

Paint 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.

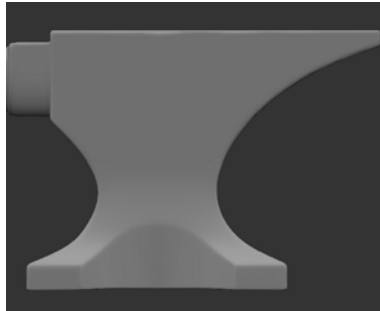
Note If you do not have the **Paint Through** tool active when trying to drag an image onto the canvas from the **Image Manager**, Mari automatically switches you to the **Paint Through** tool to continue.

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. Or, click on the white square in the middle of the image and drag it.
Resize the image, preserving the scale	Hold down Ctrl+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 , 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.



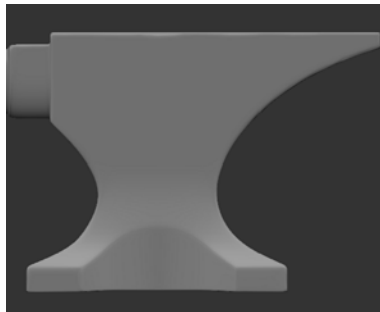
Model before painting



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

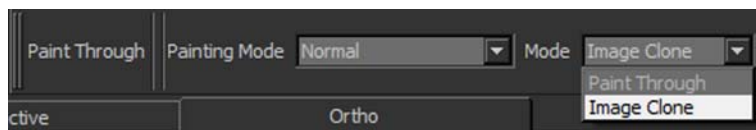
- Tip**
- *Painting with the brush lets you more subtly control how Mari applies the paint — for example, you can use a brush with a relatively low alpha, and “build up” the paint on the model using multiple strokes.*
 - *You can apply a color grade to the source image as you paint through. Set the color 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.*

Sandbox *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.*

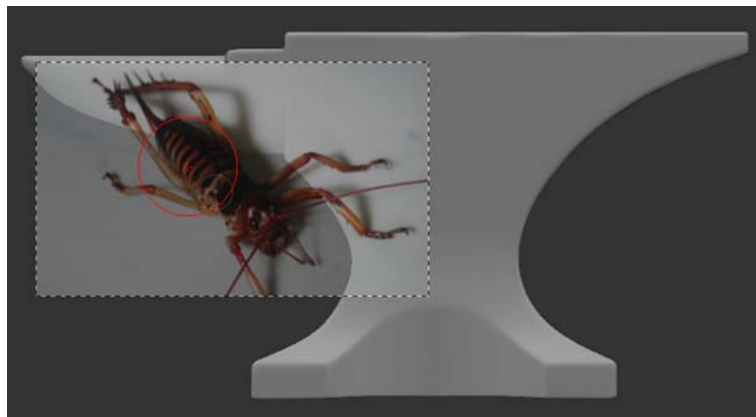
5. You can apply a **Tint**, which allows you to combine a color with the image to apply a tinted version of the image using **Paint Through**.
6. You can also apply a **Stencil**. The stencil allows you to paint through an image to create a design of the image on the model.
By default, there is **No Stencil** applied to the model.

Clone from a Floating Image

1. In the **Tool Properties** toolbar at the top of the screen, select **Image Clone** from the **Mode** dropdown:



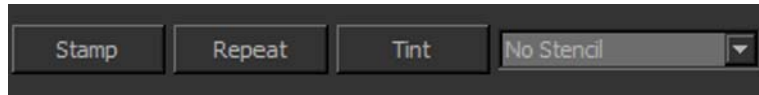
2. Make any other changes you want to the size, rotation, stretch, or position of the source image.
This is exactly the same as in the normal mode. You'll probably want to move the image next to where you'll be painting.
3. **Ctrl**-click on the floating image to set the source point.
The clone source preview displays in a red circle, while your brush is the white circle.



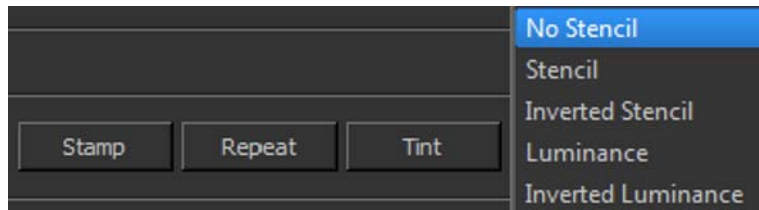
4. Paint normally.
Mari clones the texture as you paint, taking the resolution for the cloned texture from the source image. For example, a 4k image provides cloned texture at 4k resolution.

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 hotkey. 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.

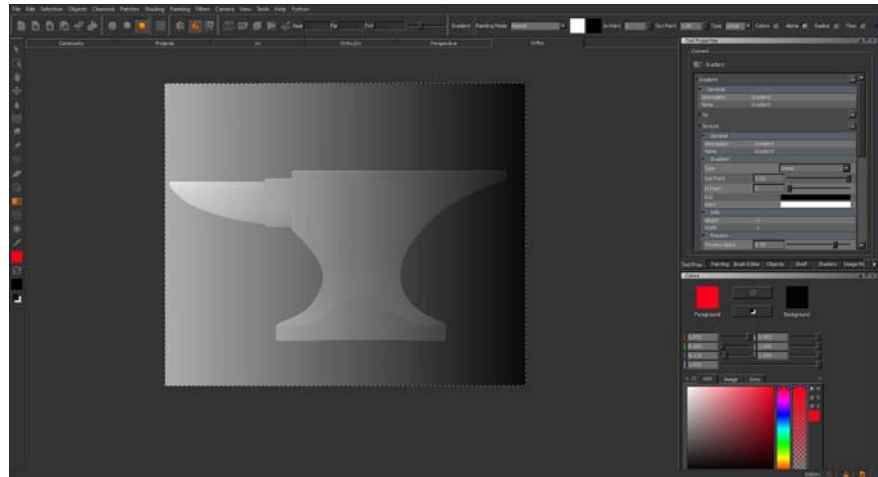
- By default, **No Stencil** is the chosen option, which simply tints the image color and does not change any other aspect.
- Selecting **Stencil** allows you to paint through an image to create a design on the model. **Stencil** ignores the color on the stencil and follows only the alpha value.
- Selecting **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.

Paint Through a Gradient

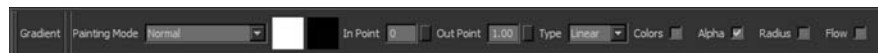
To paint through a color gradient (instead of through an image):

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 [Set the Paint Through Options](#).

4. Paint through the gradient, in the same way as described in [Paint Through an External Image](#).



7 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, [Mask 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

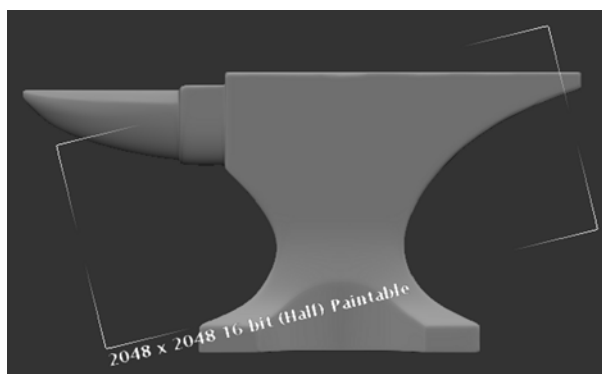
As you paint, Mari saves the painting since your last bake in the paint buffer. Then, when you bake, Mari saves the paint from the buffer, adds it to the surface underneath, and clears the buffer ready for more paint.

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 **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 channel — but until then you can edit the buffer, move the model around underneath it, or switch to another channel and bake the paint on there. 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.

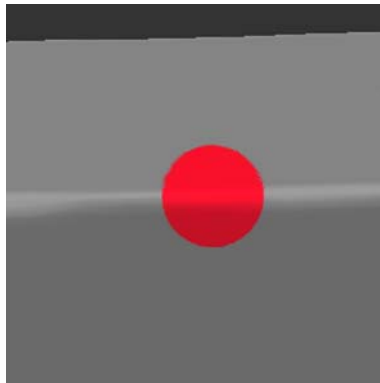
Mari includes the following mask types:

- a configurable **edge mask**, which masks the areas on the model that are oblique to the current view
- 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
- ambient occlusion mask
- depth mask
- backface mask
- fractal noise mask

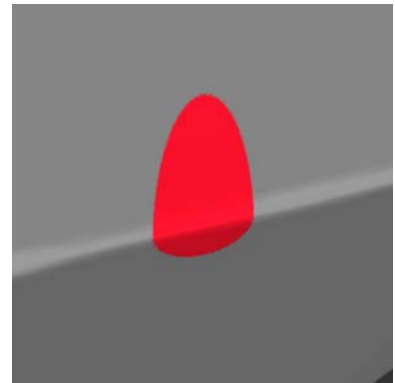
Edge Masks

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.

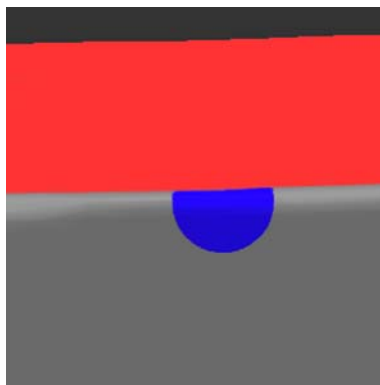


Circle painted front on

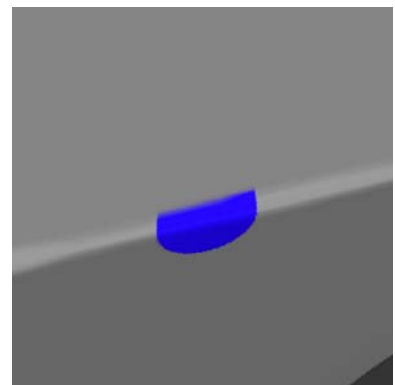


Rotated, you can see the falloff

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.



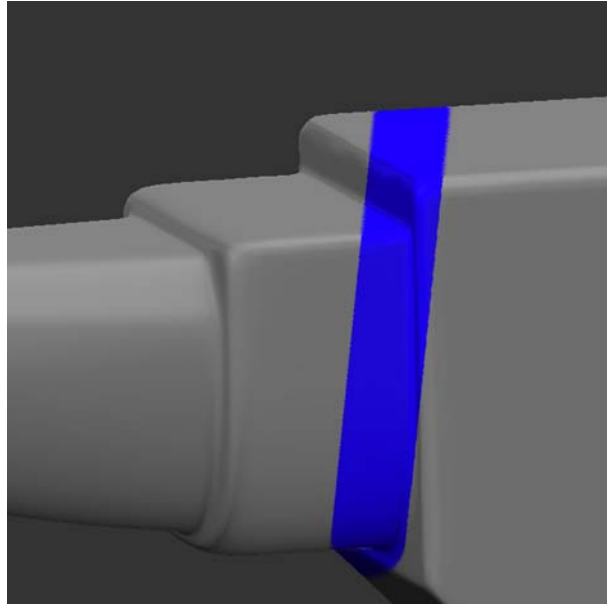
*Circle painted front on,
with edge mask.*



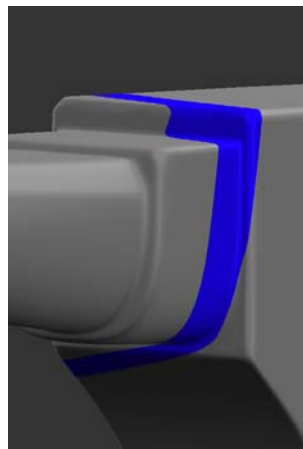
*Rotated, you can see the
mask prevented falloff.*

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.

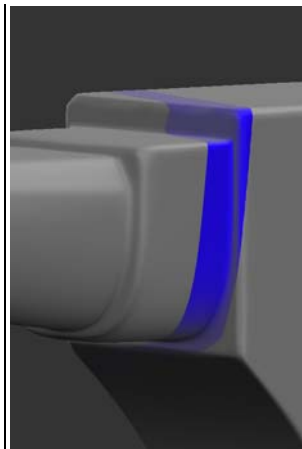
To see how this works, the following illustrations show the result of painting a stripe on the model, like this:



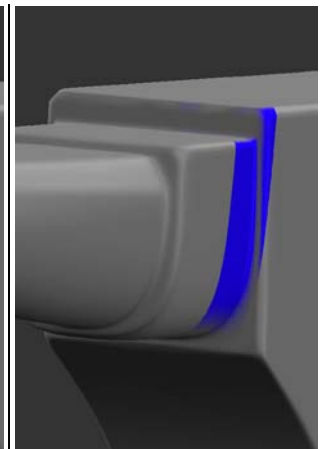
If we bake it, and then rotate to see how much “smeared” into the falloff area, you can see the results of using masks at falloff end/start points of 1.00/0.00 and 0.75/0.25:



No mask



*Mask Falloff:
End 1.00, Start 0.00*

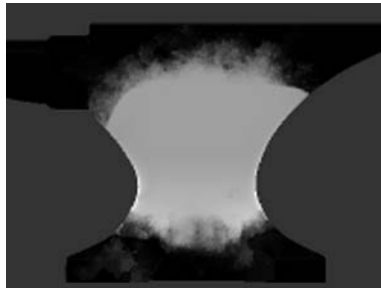


*Mask Falloff:
End 0.75, Start 0.25*

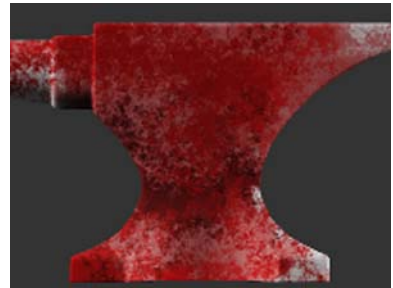
Tip *If you want to, you can set the end to a higher value than the start. This reverses the masking - the facing surfaces are masked and the oblique surfaces are paintable.*

Channel masks

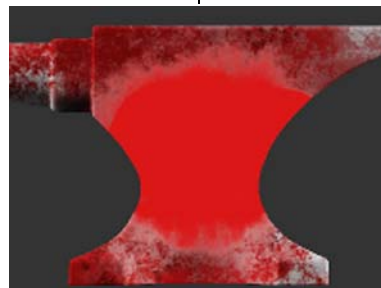
You can use a **channel mask** to manually define your own painting mask. You do this by painting into a channel and then using it to control where paint applies to the surface.



Channel mask.

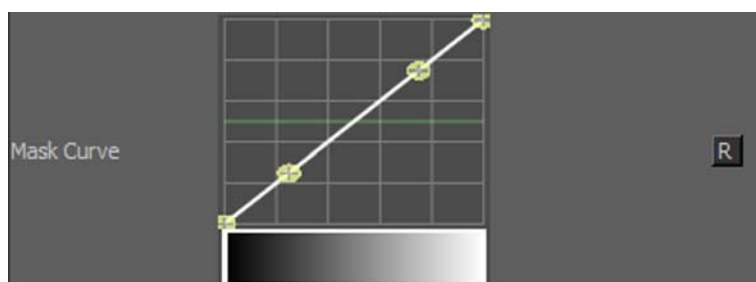


Paint buffer without channel mask.



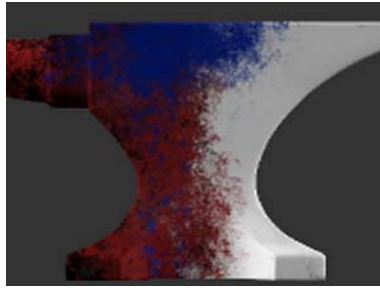
*Paint buffer with channel mask –
see the red masked area where
the white is on the mask.*

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% grey (0.5,0.5,0.5) creates a mask with 50% opacity.



Mask channel including colored mask data.

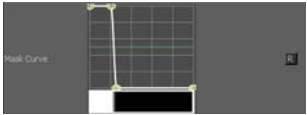
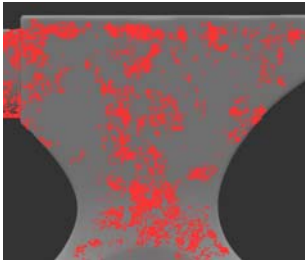
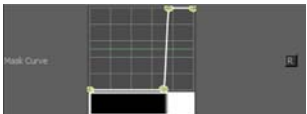


The resulting paint mask.

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:



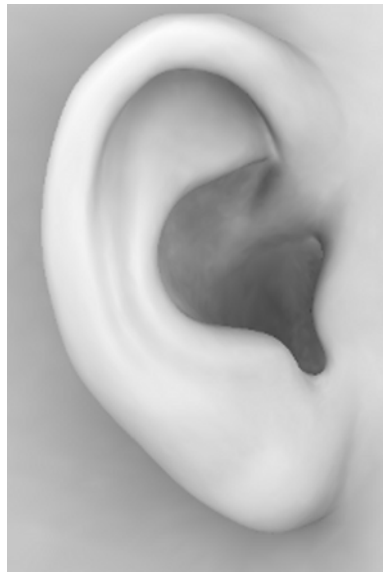
Setting	Result	Notes
		This paints into the dark areas – the cracks in the texture.
		This only unmasks the lightest (that is, the highest) points in the map.

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. If you prefer to paint your masks white on black, you can invert the mask input.*

Ambient Occlusion Mask

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.

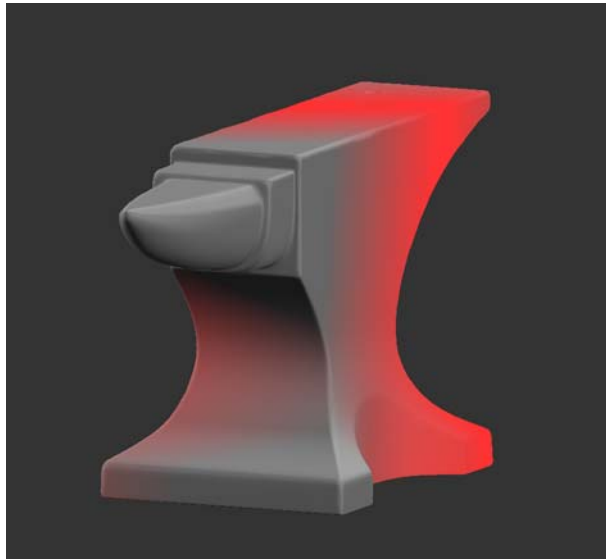


Ambient occlusion mask — the darker the area on the original, the more it's masked.

If required, you can invert the mask, so as to mask out all the areas not covered by the ambient occlusion.

Depth Mask

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

Backface Mask

Turning this on simply masks out the backfaces on your model – so you can't paint on them. There are no other configuration options.



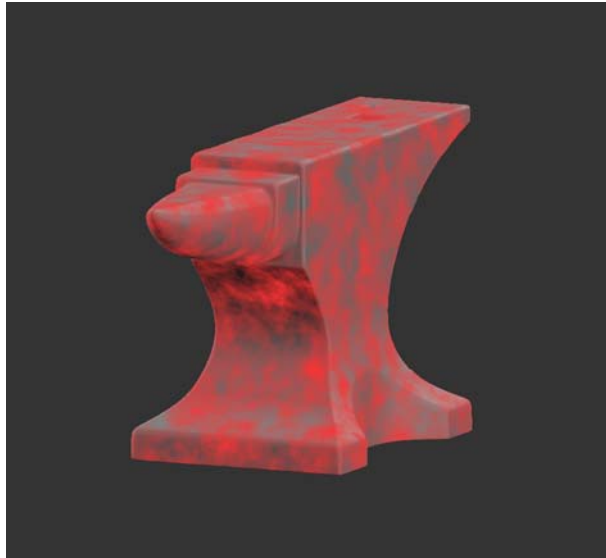
Backface mask off



*Backface mask on, masking the
inside of the lens*

Fractal Noise Mask

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 the fractal noise mask to mask where paint is applied to the surface of the object.

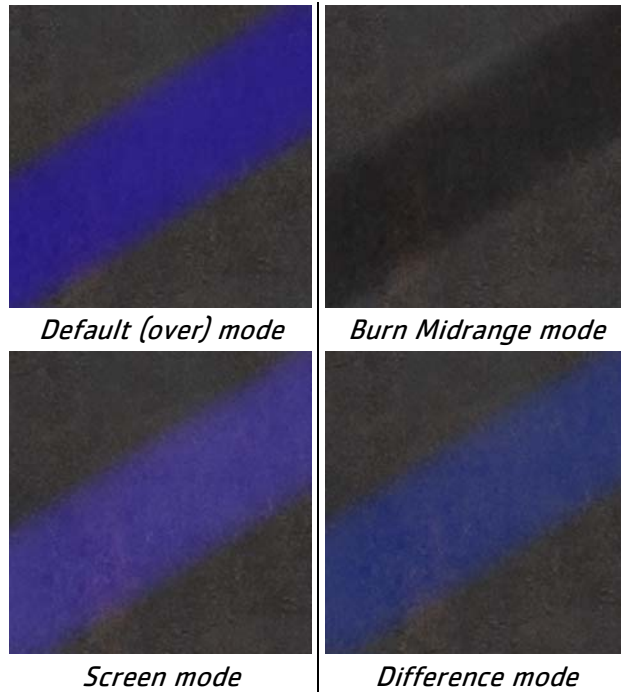
About Blend Modes

When you bake the paint buffer, Mari (by default) adds the contents of the buffer to the surface of the channel 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:



See [Paint Blending Modes](#) for details of all the blending modes available.

Steps for Controlling How Paint Applies to the Model

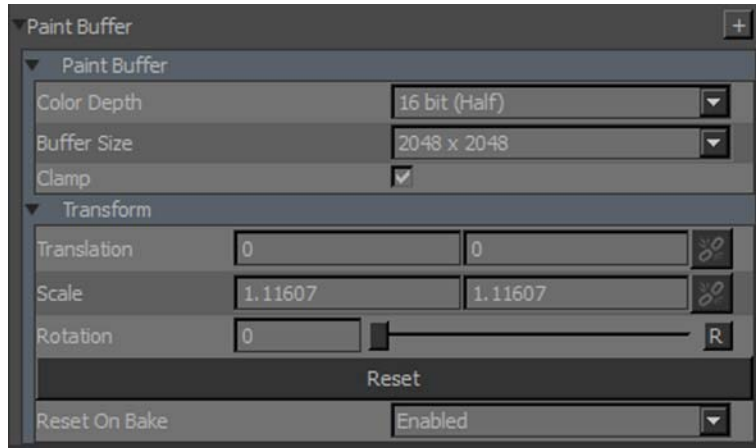
To control how your paint buffer works and how Mari bakes it to your model, follow these steps:

1. [Configure the Paint Buffer](#)
2. [Erase Paint](#)
3. [Mask What You Can Paint On](#)
4. [Set the Paint Blending Mode](#)
5. [Paint Buffer Symmetry](#)

Configure 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.

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.*

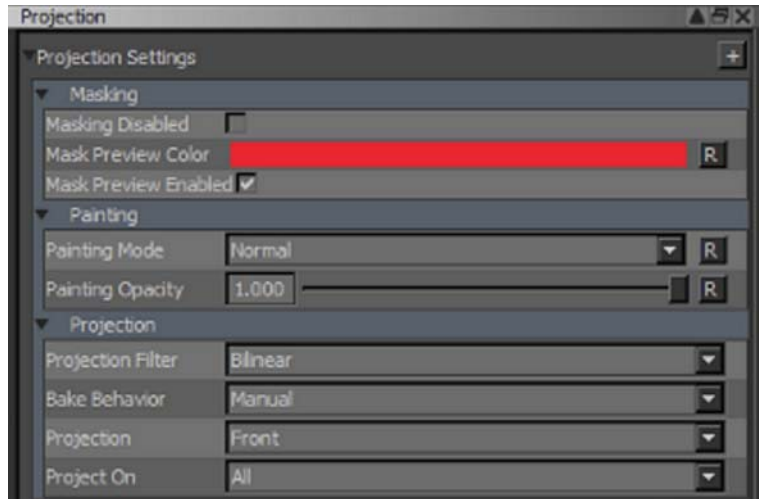
Mari supports a maximum Buffer Size of 16384 x 16384. However, this setting is not available for all graphics cards. The availability of this setting is determined by the maximum texture size supported by your graphics card.

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.

- **Transform (Translation, Scale, and Rotation)** — used to [Move or Transform Paint](#) in the buffer. But it's easier to use the **Transform Paint** tool.
- **Reset on bake** — controls whether Mari resets the buffer to its defaults after baking. If enabled (the default), 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.

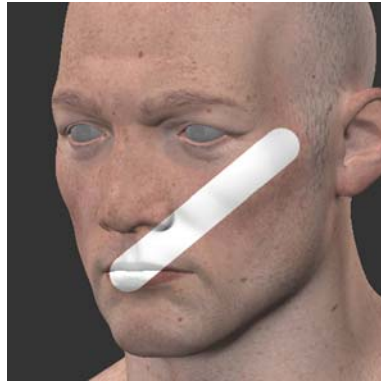
4. Open the **Projection** palette. This controls how Mari projects paint onto the model when you bake the paint, and mask settings.



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 **SelectedOnly**, 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.

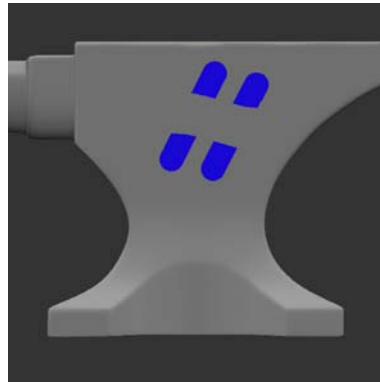
Sandbox *Paint a couple of example strokes onto your model, so you can see the effects as you work.*

Try:

- Setting [Bake Behavior](#) to **Manual** or **ClearOnly**, 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 **PageUp** and **PageDown** 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.

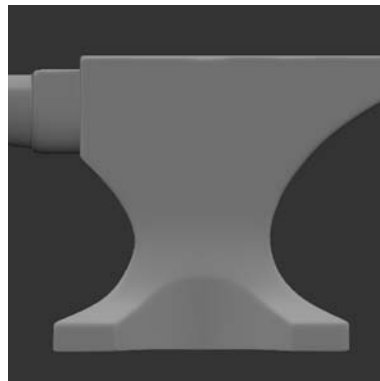
Erase Paint

1. To erase a section of paint, click on the **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+Shift+C**.*

Mask What You Can Paint On

1. Scroll to the top of the **Projection** palette.

*Masking options are under the **Mask Preview**, **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+M**. This 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). 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*

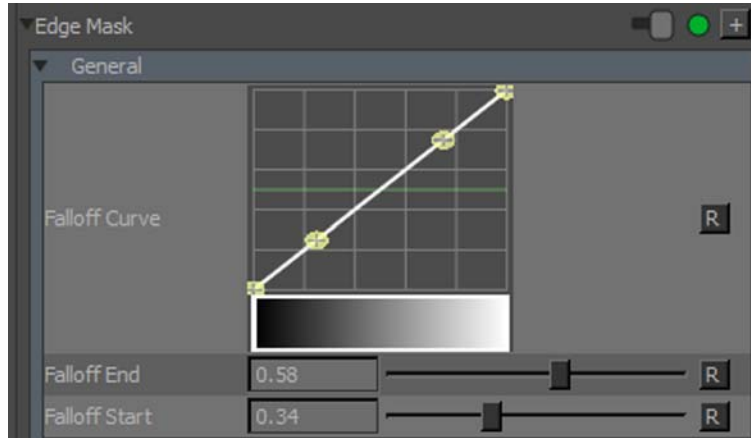
 *for depth masking*

 *for backface masking*

 *for masking disabled*

Edge Masking

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.

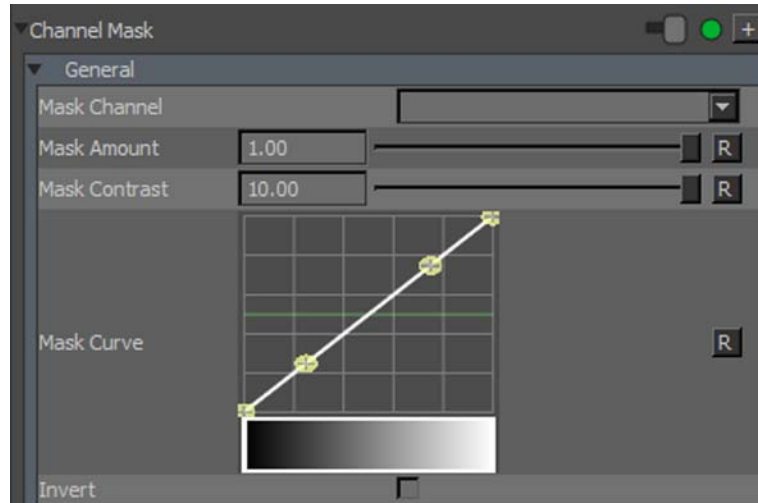
Enable the masking by turning on the toggle at the top of the section.

When enabled, the  icon displays in the status bar. Clicking this icon (as well as the checkbox in the palette) disables the edge mask.

Tip You can use shortcut keys to enable/disable the edge mask (**G**) and hide/show the masking (**,**).

Channel Masking

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'll want 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% grey 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. If you prefer to paint white on black, you can invert the mask input.

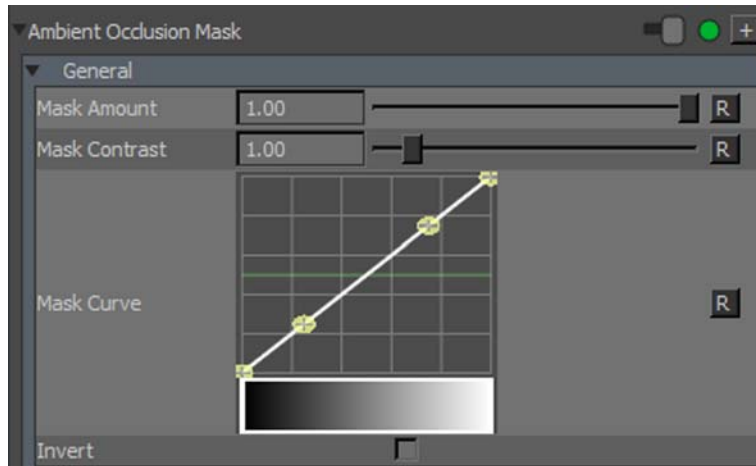
When enabled, the  icon displays in the status bar. Clicking this icon (as well as the checkbox in the palette) disables the channel mask.

Tip If you want to turn channel masking off, just select the blank entry from the **Mask Channel** list.

Ambient Occlusion Masking

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**:

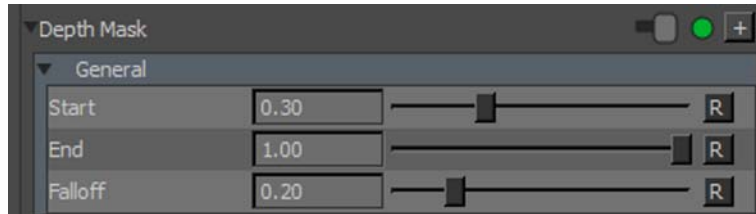


- **Mask Amount** — is the amount of masking to apply. This is from 0 to 1; higher values mask the areas that are covered by ambient occlusion, while lower values mask everything except the areas that are covered by the occlusion.
- **Mask Contrast** — controls how sharply defined the mask is. At 1.0, 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% grey 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. Clicking this icon (as well as the checkbox in the palette) disables the ambient occlusion mask.

Depth Masking

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. Clicking this icon (as well as the checkbox in the palette) disables the depth mask.

Sandbox *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.*

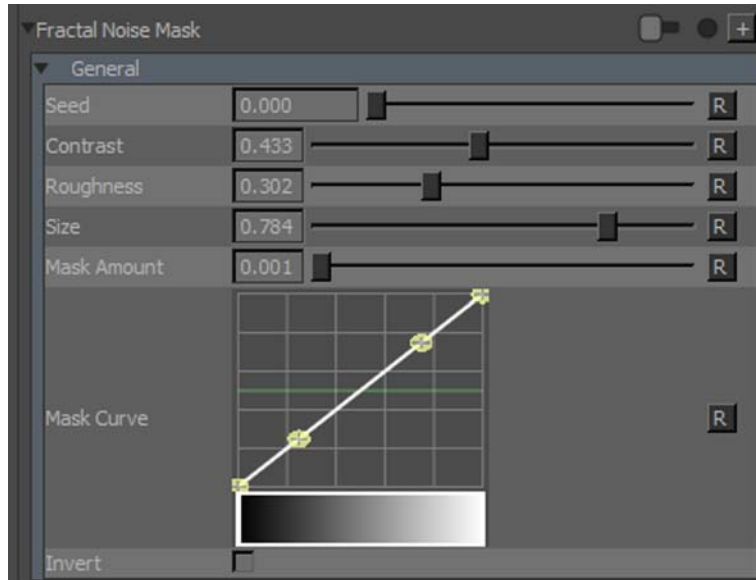
Tip *For details on creating new channels, see [Channels](#).*

Backface Masking

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. Clicking this icon (as well as the checkbox in the palette) disables the backface mask.

Fractal Noise Masking To turn fractal noise masking on, set the options under **Fractal Noise Mask**:

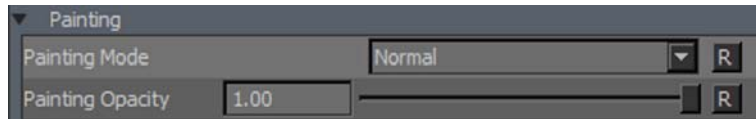


- **Seed** — changes the pattern of the fractal noise mask by moving the pattern in 3D space.
- **Contrast** — controls the level of contrast applied to the fractal noise features to derive the mask.
- **Roughness** — determines the roughness of the fractal noise features.
- **Size** — Determines the 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. Clicking this icon (as well as the checkbox in the palette) disables the fractal noise mask.

Set the Paint Blending Mode

1. In the **Projection** palette, scroll to the **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.

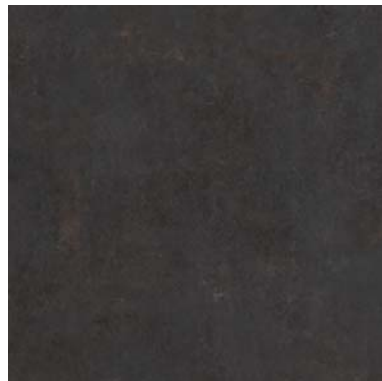
The paint blending modes available are similar to many other paint packages. If you aren't familiar with the various modes, see [Paint Blending Modes](#) for descriptions and examples.

As you change modes, the paint in the buffer updates to show the effect.

Sandbox *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 blending mode. For a list of available shortcuts, please refer to [Keyboard Shortcuts](#) on page 356.

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 channel's surface.	
Burn Highlights	Burns the channel based on the highlights in the paint buffer.	
Burn Midrange	Burns the channel based on the middle range (that is, not the lightest or darkest pixels) of the paint buffer.	

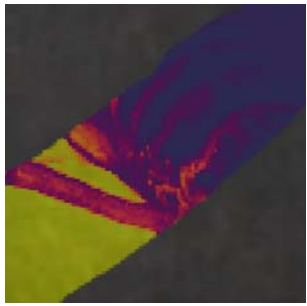
Blend mode	Description	Example
Burn Shadows	Burns the channel based on the contents of the paint buffer, with darker pixels being burned harder.	
Clear	Subtracts the opacity of the paint stroke from the current surface. For example, an 80% opaque stroke leaves a surface with 20% opacity.	
Color	Takes the luminance of the channel, and the color and saturation of the paint buffer.	
Color Burn	Burns the channel, using the contents of the paint buffer as input for the burn operation.	

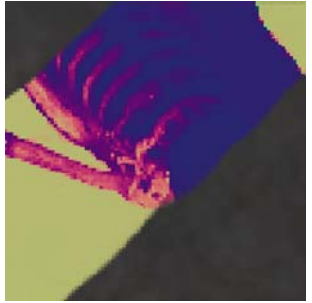
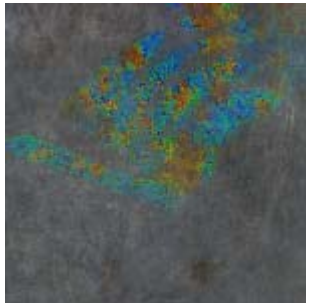
Blend mode	Description	Example
Color Dodge	Dodges the channel surface, using the contents of the paint buffer to control the degree of dodging.	
Darken	Gives each pixel the darker of two possible values: the current channel surface or the contents of the paint buffer (whichever is darker).	
Difference	Inverts the colors. Mari subtracts the values in the paint buffer from those on the channel surface.	
Dodge Highlights	Dodges the channel based on the highlights in the paint buffer.	

Blend mode	Description	Example
Dodge Midrange	Dodges the channel based on the middle range (that is, not the lightest or darkest pixels) of the paint buffer.	
Dodge Shadows	Dodges the channel 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 channel pixels, with the hue from the paint buffer.	
Inverse Difference	Subtracts the colors in the paint buffer from the channel surface, and then inverts the result.	
Invert	Uses the paint buffer as the input to invert the pixels on the base layer.	

Blend mode	Description	Example
Lighten	Gives each pixel the lighter of two possible values: the current channel surface or the contents of the paint buffer (whichever is lighter).	
Luminance	Takes the hue and saturation of the channel color, and the luminance of the paint buffer. (This is the opposite of the Color blend mode.)	
Multiply	Darkens the color. Mari multiplies the channel color by the color in the paint buffer.	
None	Doesn't add the paint to the surface.	

Blend mode	Description	Example
Normal	Replace values on the channel'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.	
Overlay	Uses the patterns and colors from the paint buffer, but keeps the highlights and shadows from the channel.	

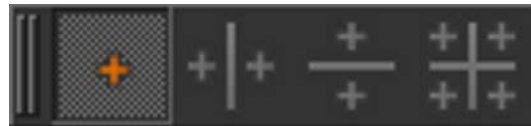
Blend mode	Description	Example
Paint Flow Vectors	Changes the paint buffer to tangent vector space. Keeps the vectors flat to the object's surface in 2D space. Unlike Nudge Flow Vectors , the Paint Flow Vectors mode blends between the vectors instead of adding them. This mode is best for use with flow 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.	
Pin Light	For light areas in the paint buffer, replaces pixels on the channel's surface that are darker than the corresponding pixels in the paint buffer. For dark areas in the paint buffer, replaces pixels on the channel's surface that are lighter than the corresponding pixels in the paint buffer. (Otherwise, leaves the channel surface unchanged.)	
Saturation	Takes the luminance and hue of the channel color, with the saturation from the paint buffer.	

Blend mode	Description	Example
Scale Vectors	This is most useful with the basic Paint tool, while preferably using greyscale paint. Scale Vectors uses the intensity of color to control the length of the normal in 3D space.	
Screen	Lightens the image. Mari divides the color on the surface of the channel by the value of the paint. (This is the opposite of Multiply.)	
Soft Light	In dark areas on the channel, burns the paint buffer onto the channel surface. In lighter areas, dodges it on.	
Sponge Desaturate	Uses the paint buffer as the input to desaturate the pixels on the base layer.	

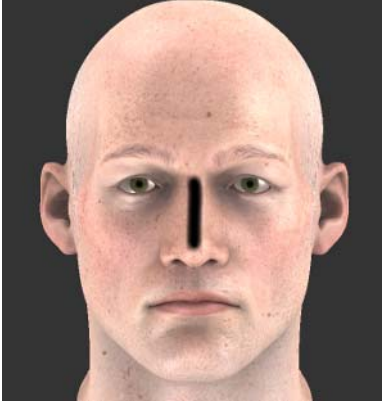
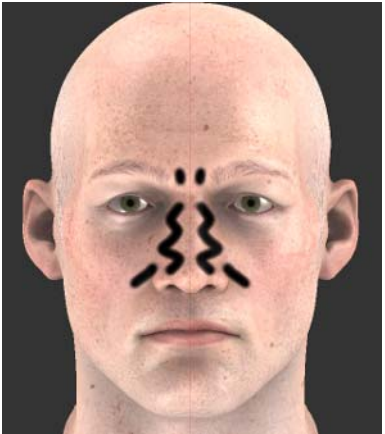
Blend mode	Description	Example
Vivid Light	In dark areas on the channel, adds the paint buffer with more contrast. In light areas, adds it with less contrast.	

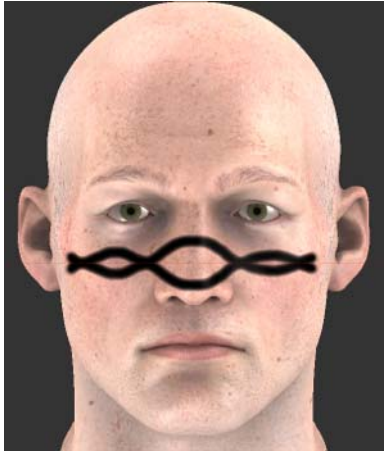
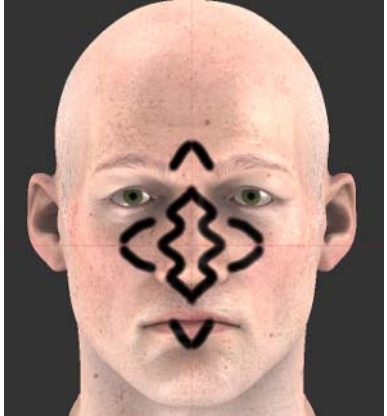
Paint Buffer Symmetry

You can mirror paint strokes in the paint buffer. On the **Paint Buffer Symmetry** toolbar, four options 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. The toolbar, shown below, appears in Mari alongside other toolbars such as **Lighting** and **Navigation**.



The four options on the toolbar 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 mirrors the same strokes on the other side.	

Symmetry Option	Description	Example
	Mirroring top and bottom divides the paint buffer horizontally. Paint strokes created on either side of the buffer mirrors the same strokes on the other side.	
	Mirroring four ways divides the paint buffer into quadrants. Paint strokes created in any of the four quadrants mirrors 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.

8 TRANSFORMING AND WARPING PAINT

Mari lets you transform and distort paint within the buffer with these four tools:

-  **Transform Paint**
-  **Warp**
-  **Slerp**
-  **Pinup.**

Tip See [Controlling the Way Mari Applies Paint to Your Model](#) for details of the how the paint buffer works.

Transforming the Paint Buffer

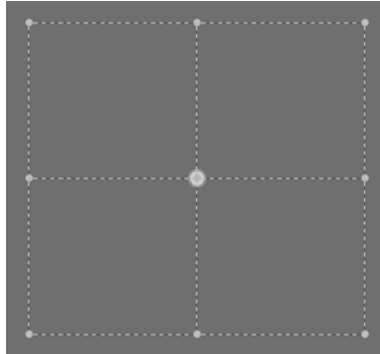
You can use the **Transform Paint** tool to resize and stretch the paint buffer as a whole. This lets you:

- resize the buffer (either proportionally or stretched)
- rotate it
- or just move it around.

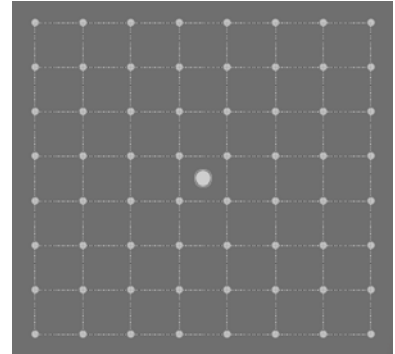
As you transform the paint buffer, its contents — that is, the paint as it appears on your model — changes too. For example, you can paint a stroke, then stretch it, rotate it, and position it exactly on your model, before finally baking it onto the layer.

Warping Within the Paint Buffer

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**.

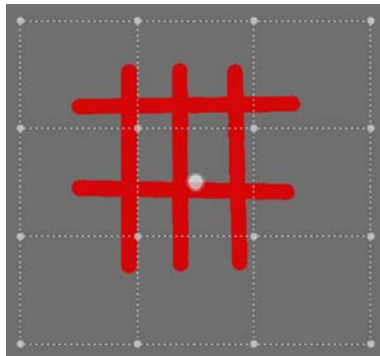


Warp box with
four control points.

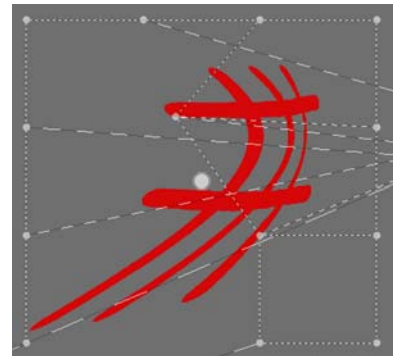


Warp box with
64 control points.

You can then warp the paint buffer under the box by moving the control points around.



Initial transform box
(16 control points).



Initial transform box
after some warping.

Pulling Paint Around with Slerp

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 wasp 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.

Pinning and Pulling with Pinup

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 affecting 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.*

Steps for Transforming and Warping Paint

To transform or warp paint in the buffer, follow these steps:

1. [Move or Transform Paint](#)
2. [Warp Paint Using a Grid](#)
3. [Pull, Smudge, Grow, and Shrink Paint](#)
4. [Distort the Paint with Pins](#)

Move or Transform Paint

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. Click to select the **Transform Paint** tool .

4. To transform the paint, you can do the following:

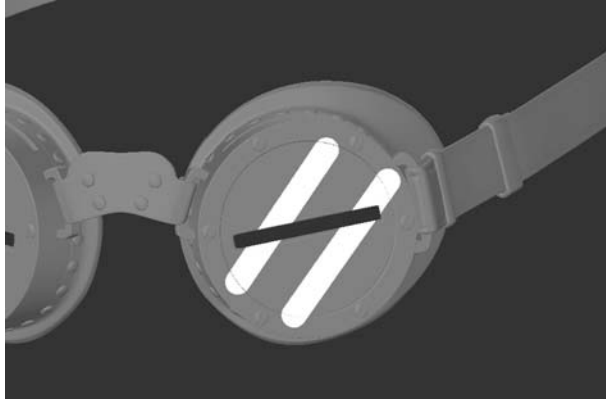
To...	You need to...	Example:
Move the paint buffer around	Click on the buffer and drag it.	
Resize the paint buffer, preserving the scale	Hold down Ctrl+Shift , click anywhere on the canvas and drag. Dragging to the left makes the buffer smaller, to the right makes it bigger.	
Stretch the buffer (resize 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, 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.

Sandbox 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.

Warp Paint Using a Grid

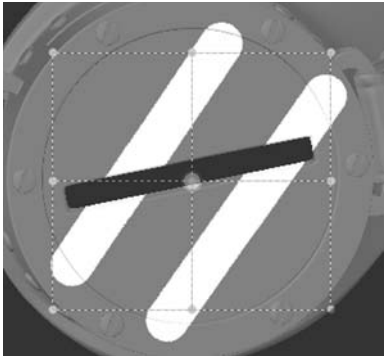
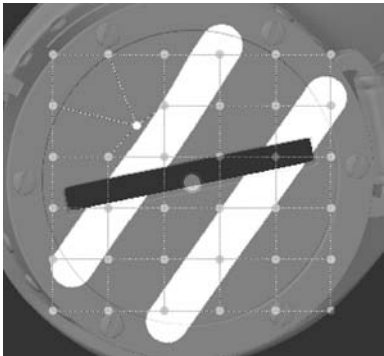
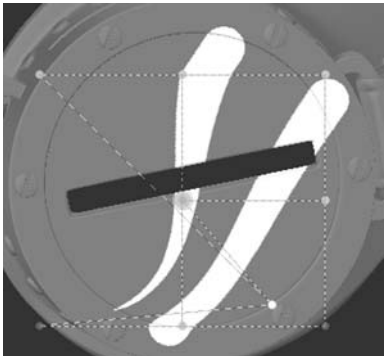
1. If necessary, paint another couple of strokes on the model to experiment with.

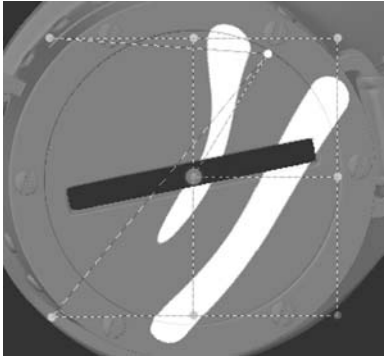


2. Click to select the **Warp** tool .
3. Click and drag to create a rectangle to warp.
The warp rectangle displays, with nine control points, and a central rotation pivot (the large point in the center).

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 (↑ ↓).	
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.	

To...	You need to...	Example:
Rotate around the central pivot	hold down the Windows (meta) key, then drag one or more points.	
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.*

Sandbox *Have a play with the warp tool. Try:*

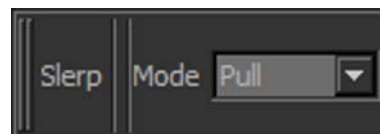
- *moving one point*
- *clicking and dragging to select two points, then moving them together*
- *using the Windows (meta) key to rotate points.*

Pull, Smudge, Grow, and Shrink Paint

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.	
make the paint in a specific area smaller	select the Shrink mode, click and drag.	

To...	You need to...	Example:
rotate your paint around	select the Rotate mode, click and drag.	
undo the distortion in a specific area of the painting	select the Erase 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.*

Sandbox *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.*

Distort the Paint with Pins

The **Pinup** tool lets you create your own control points to distort the paint buffer. You can use pins to pull the paint around like a rubber sheet, or to hold areas of the buffer steady while you distort other parts.

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 Windows (meta) 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+' (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.

Tip 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.

Sandbox Using pins is also hard to describe but quite easy to get the hang of. Try doing the following:

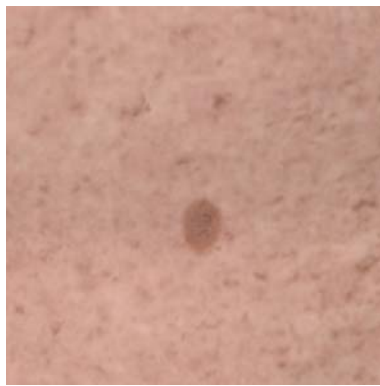
- *Create some pins*
- *Drag one pin around*
- *Select multiple pins and drag them around*
- *Rotate pins*
- *Move the point of rotation, and rotate pins around the new point of rotation*
- *Increase and decrease the strength of individual pins. See how stronger pins affect the texture more.*



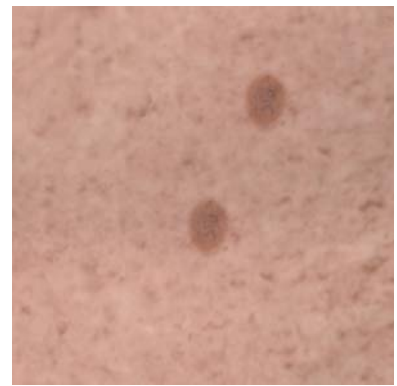
9 TOWING PAINT AROUND

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.

Steps for Towing Paint

To tow paint around on the surface, follow these steps:

1. [Select and Tow Paint](#)

Select and Tow 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.

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 tex-

ture, 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.*

Sandbox *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.*



10 COPYING AND PASTING PAINT

Mari lets you:

- copy and paste single patches quickly, or
- copy multiple patches at once.

In both cases, you can copy between channels. When copying multiple patches, you can copy to other objects in your project.

Copying a Single Patch

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.

Copy Textures Between Channels and Objects

Mari's Copy Multiple Textures function lets you copy entire patches between places in your project. You can copy patches:

- between objects
- between channels on the same object
- between patches on the same channel.

You can only copy textures on a patch-to-patch basis – you cannot copy one patch onto more than one other patch at once.

Using Offsets

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.

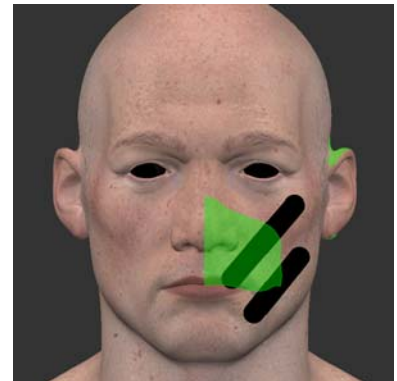
So, for example, if your model has a consistent numbering scheme, you can use this to paste from one side of the model to another. As you do this, Mari reverses the textures.

Example

The blacksmith model available with Mari. Here, we paint a couple of black stripes on the face, and then select patch 1002:

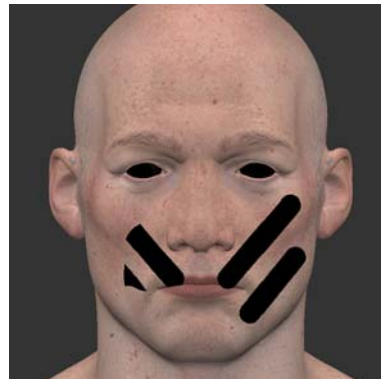


Black stripes painted



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.



Steps for Copying and Pasting Paint

To tow paint around on the surface, follow these steps:

1. [Copying and Pasting Single Patches](#)
2. [Copying and Pasting Multiple Patches](#)

Copying and Pasting Single Patches

1. Select your source patch.
This only works with single patches. You cannot copy multiple patches at once.
2. From the **Patches** menu, select **Quick Copy**.

3. Select your destination patch(es).

This can be on the same channel or a different channel on the same object.

4. From the **Patches** menu, select **Quick Paste**.

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.

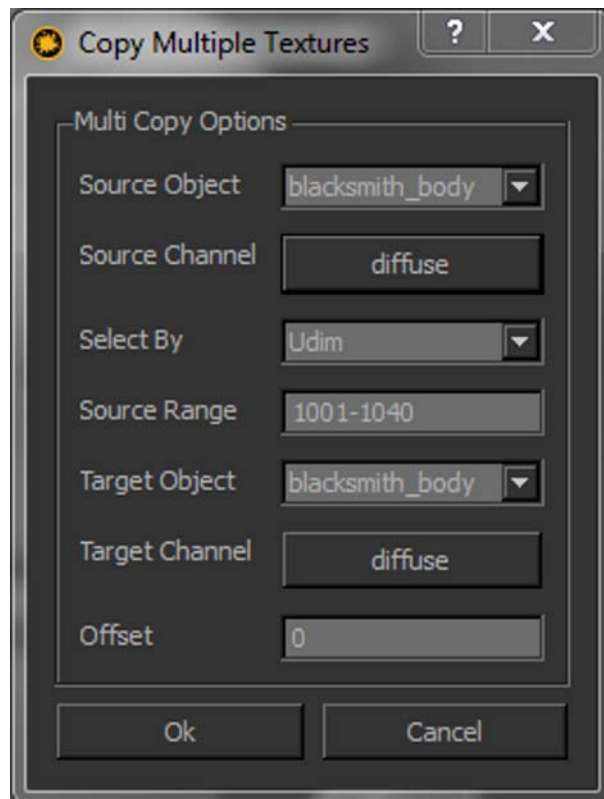
Mari pastes a copy of the source patch onto the destination.

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.

Copying and Pasting Multiple Patches

1. From the **Channels** menu, select **Copy Textures**.

*The **Copy Multiple Textures** dialog box displays:*



2. Set the details of where to take the textures from:

- **Source Object** and **Source Channel** set the location of the source patches.
- Set the **Select By** field to the patch numbering scheme you want to use to select the patches to copy.
- In the **Source Range** 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 **Target Channel** 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 channel.
 - With an offset of 20, patch 1001 is copied to patch 1021 in the target channel (which can be the same as the source channel).
5. Click **OK** to copy the textures.

Mari copies the textures to the designated object, channel, and patches.

11 EDITING PAINT ON CHANNELS

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 channel, while the **Clone Stamp** tool includes a mode to clone paint directly up off the surface of the channel into the paint buffer.

Blur

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](#).*

Pulling Paint into the Buffer

The **Clone Stamp** tool has a mode that clones the surface of the current channel 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 channel surface into the paint buffer.

Steps for Editing Paint on Channels

To paint a channel in Mari, follow these steps:

1. [Blur Paint on a Channel](#)

Blur Paint on a Channel

1. Start Mari and open your project.
2. Select a channel with some paint on it.
3. Select the **Blur** tool .
4. Click and drag to blur your paint.

Sandbox *Experiment with different brush settings to see how this affects the final result. Particularly, try changing the brush:*

- *size*
- *shape*
- *opacity (affects the strength of the blur).*

12 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). Just as these have a pattern that tends toward a particular direction, or lies 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.

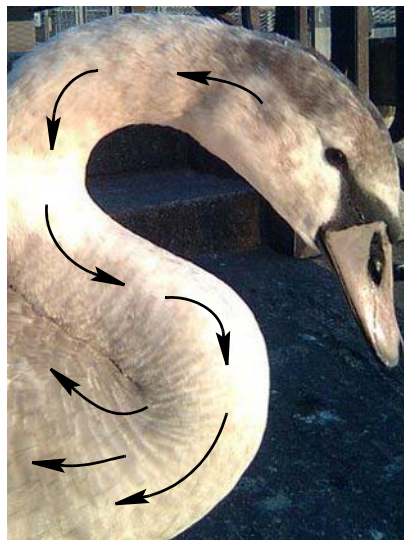


Figure 12.1: Arrows indicate the direction of the feathers on the swan's body, which vectors should mimic.

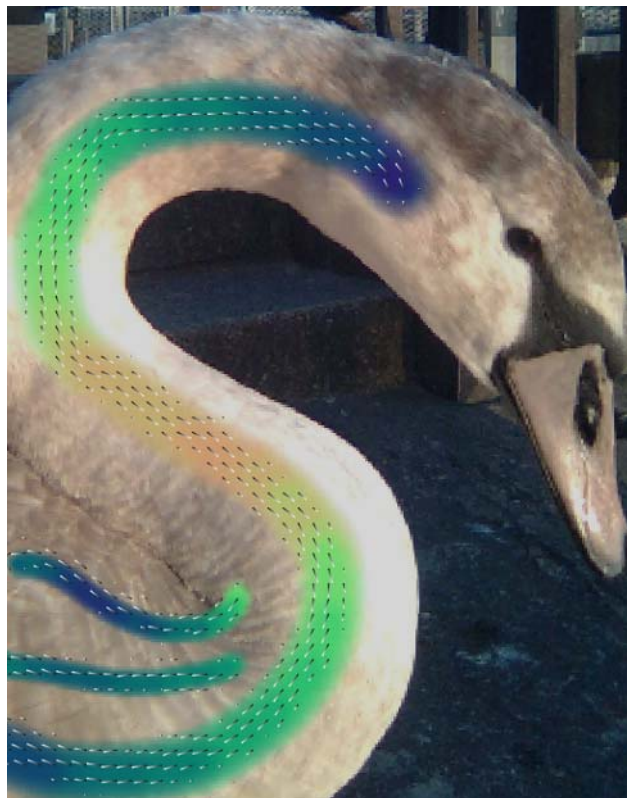


Figure 12.2: Arrows show the direction of painted vectors, which indicate the flow of vectors.

For example, in [Figure 12.1](#) 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 [Figure 12.2](#) the vectors have been painted on the image to show how this flow pattern might appear.

Vector Inspector

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 their properties and spacing. Like the paint buffer, the **Vector Inspector** can be moved around the canvas or rotated using keyboard shortcuts.



As seen above, the **Vector Inspector** shows the markers for the entire image, not just the vector brushing.

Vector-related Paint Blending Modes

Five 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 are available in the paint blending mode dropdown menu in both the **Paint** and **Vector Paint** toolbars, as well as the other paint tools, such as **Clone Stamp** and **Paint Through**.

The **Paint Flow Vectors** and **Paint Normal Vectors** modes are simply an alpha blend between the vectors. The **Nudge Flow Vectors** and **Nudge Normal Vectors** modes, on the other hand, 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 using black or white paint with the basic **Paint** tool.

The paint and nudge vector modes simply 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.

For more information on how to use these paint blending modes, see [Paint Blending Modes](#) on page 89 in the chapter on [Controlling the Way Mari Applies Paint to Your Model](#).

Vector Shaders

Three shader modules – **Diffuse Tangent to Screen**, **Diffuse Tangent to World**, and **Vector Data** – give you greater control over how you can visualize the **Vector Paint** tool. Among other functions, these shaders allow you to specify the properties for your vector visualization by adjusting the vector data or inverting it according to your needs.

The **Flow** shader module allows you to use your vector paint as surface flow over time. Select the vector data channel and 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. Enable the **Animation** attribute to see the flow over time.

For more information on what these shaders are and how to use them, see [Vector Brushing](#) on page 243 and [Shader Module Types and Examples](#) on page 255 in the [Shaders](#) chapter.

Vector Default Shaders

In addition to the shader modules listed above, there is also a method for quickly setting up default vector shaders, accessible from two locations in the UI. This method, described in [Adding Default Vector Shaders](#), creates two default shaders in the **Shaders** palette for creating flow maps and normal maps. The **Default Vector (Tangent Space)** shader is for viewing vector (colors) in tangent space, while the **Default Vector (World Space)** shader is for viewing vector (colors) in world space.

Vector 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.

Steps for Working with Vector Brushing

To use vector brushing in your Mari project, follow these steps:

1. [Painting Vectors](#)
2. [Modifying Vectors with Paint Blending Modes](#)
3. [Inspecting Your Vector Work](#)
4. [Adding Vector Shader Modules](#)
5. [Adding Default Vector Shaders](#)
6. [Changing the File Space](#)

Painting Vectors

Begin by starting Mari and opening your project.

1. Ensure that you are in a diffuse channel, then select the **Vector Paint**



tool.

2. If you want to see the directional markers while you paint, select the

Vector Inspector icon  in the **Vector** toolbar.

3. Select a brush from the **Brush Editor** palette. Once you have set up the brush options, paint in a direction across the surface of the model.

A specific color is chosen depending on the direction of your brush stroke.

Note 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)

Modifying Vectors with Paint Blending Modes

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, and
 - Scale Vectors.
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 **Projection** palette.

Note *As with other paint blending modes, the vector modes apply to paint that has not been baked down yet.*

Inspecting Your Vector Work

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 data that is piped along the **Vector Data** shader module. If there is no explicit **Vector Data** information, the **Vector Inspector** displays the unlit diffuse.*

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.
 - **Line spacing** - adjusts the space between the directional markers.
 - **Line length** - adjusts the length of the directional markers.
 - **Line width** - adjusts the width (thickness) 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 **ScreenSpace**, **TangentSpace** (no change), or **WorldSpace**. Screen space

is the most beneficial option for painting flow and normal maps. Most often **ScreenSpace** is the most beneficial option.

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 entry box 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 Shader Modules

To add vector shader modules to your project, follow these steps:

1. Open the **Shaders** palette. Create a new surface shader or select an existing shader.
2. Select **Add New Shader Module**.
*The **Add New Shader Module** dialog opens.*
3. Select one of the following vector-related shader modules:
 - **Diffuse Tangent to Screen**
 - **Diffuse Tangent to World**
 - **Vector Data**
 - **Flow**
4. Click **OK** and adjust the properties in the shaders palette.

Note *For a full description of the shader module properties, see the [Shader Module Types and Examples](#) in the [Shaders](#) chapter.*

Adding Default Vector Shaders

To create vector brush default shaders, follow these steps:

1. Open the **Shaders** palette to manage newly-created shaders and shader modules.
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.*

These shaders, **Default Vector (Tangent Space)** and **Default Vector (World Space)**, allow you to quickly plug channels into the shaders and start painting flow or normal maps. During creation, each of the shaders are given the **Vector Data** shader module. Additionally, the **Default Vector (World Space)** shader also creates a **Diffuse Tangent to World** shader module. The **Default Vector (Tangent Space)** shader also creates a **Suppress Blue (RGB Component Filter)** module that does not exist in the world space shader. By default, this module is not active.

Note that 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.

Changing the File Space

Changing the file space before exporting can only be done on channels and images. This converts the file space range from 0 to 1 (the Mari standard), to -1.0 to +1.0 (Vector standard) or -1.0 to +1.0 with inverted Y vectors (Vector inverted standard). 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)**.

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 > MrImage > Info | File Space**, click the field and select either **Normal**, **Vector**, or **Vector Y Flip** from the dropdown menu.

13 EXPORTING AND IMPORTING TEXTURES

Export is how your work moves on into the next step in the pipeline. Similarly, import is how you can pick up an existing set of textures to work on:

- Export saves out the paint on your model as a set of image files (one file per UV patch).
- Import loads images into the current channel on your model.

Export and import only affect a single channel, possibly across multiple objects, at a time. Within that channel, you can:

- export or import the entire channel (all patches)
- export or import only selected patches
- export only patches that have changed since last exported (not available for import).

Mari supports standard image formats for export and import (such as **.tif**, **.exr**, **.psd**, **.png**, **.jpg**, **.tga**, and **.dds**).

Importing Textures When Creating Channels

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](#) on page 208 for details.

Steps for Importing and Exporting Textures

To import and export textures in Mari, follow the instructions below:

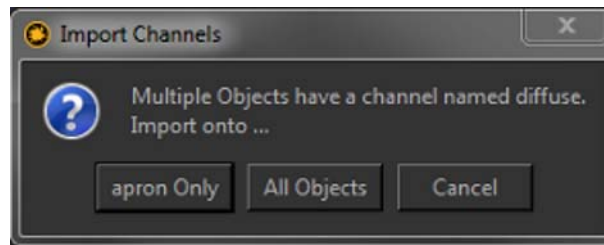
1. [Import Existing Textures](#)
2. [Export Textures](#)
3. [Export to Maya](#)

Import Existing Textures

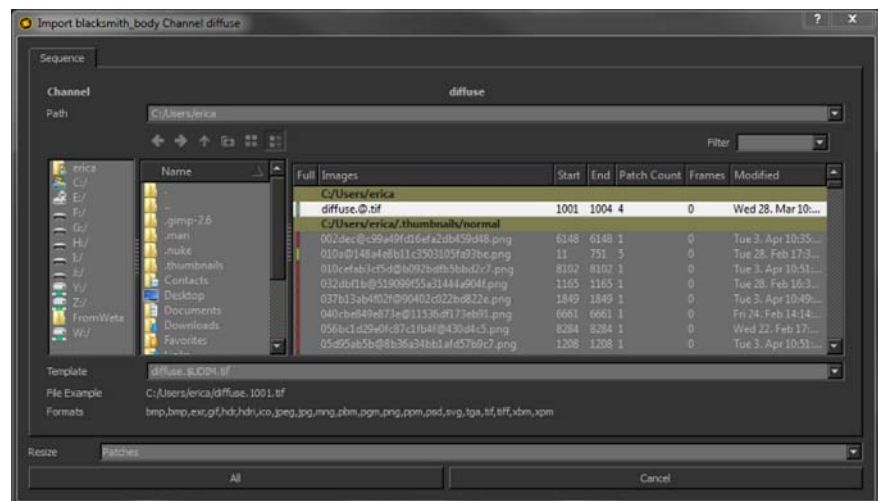
1. Start Mari and open your project.
2. Select a channel with some paint on it.
3. If you want to import only onto particular patches, select those patches.
From the **Channels** menu, select **Import**.

Tip *This is also available as an option from the right-click dropdown menu on the main canvas and in the **Channels** palette.*

If you have multiple objects, and other objects have channels with the same name as the channel you are importing onto, Mari asks whether to import only to the current object, or all objects:



4. Click to select where to import to.
*The **Import** dialog box opens:*



Note *If you are importing onto all objects in the project, this dialog box is for the first object in the project, **not** the object you had selected!*

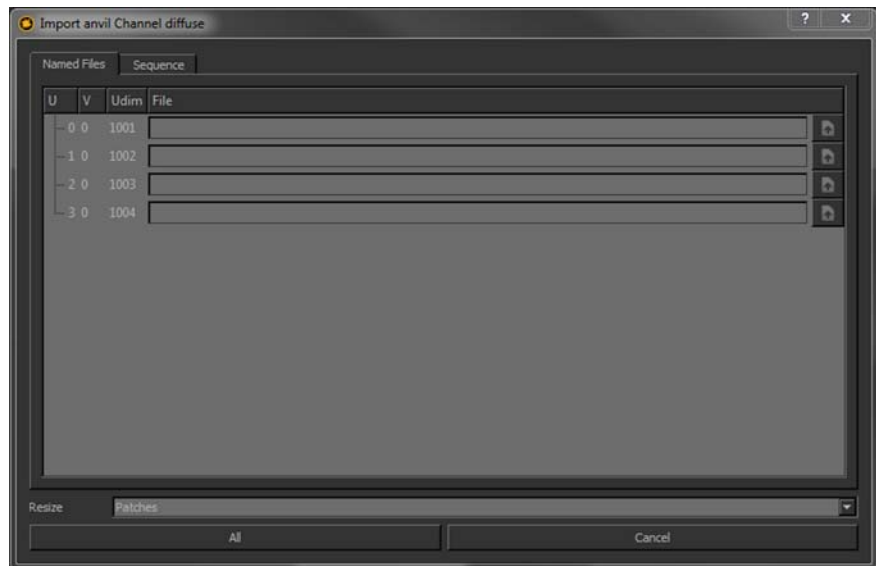
Now you can choose to [Import Individual Files](#), or [Import a Sequence of Files](#).

Import Individual Files

If the object has less than 20 UV patches, the **Import** dialog box has a **Named Files** tab. This lets you individually pick which files to load on each patch.

1. Click on the **Named Files** tab.

The tab shows all the patches on the object.



2. For each patch, set the file 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.

Tip *By default, if your model has more than 20 patches, the **Named Files** tab does not display.*

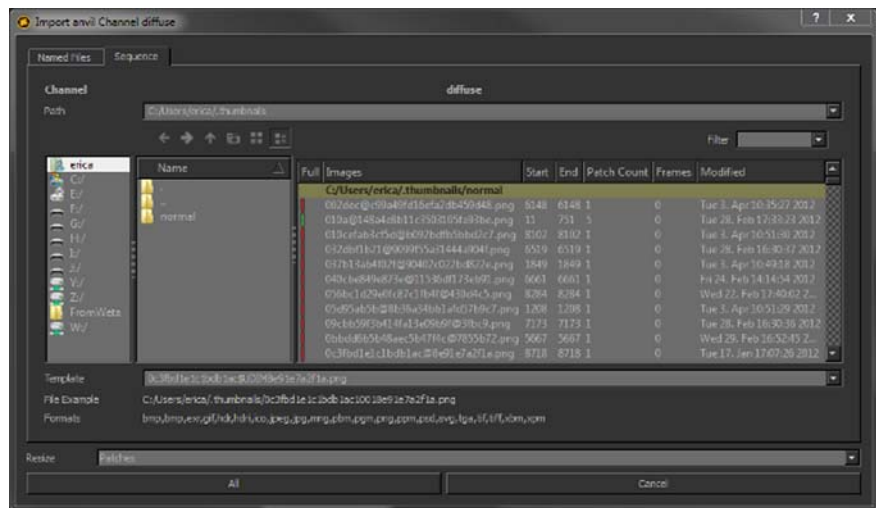
*To change this, set the **Patch Cutoff** preference to a number greater than the number of patches in the file. Alternatively, you can use the `MARI_NAMEDFILEIMPORT_SEQUENCELIMIT` environment variable to override the preference setting. Please see the section on [Environment Variables That Mari Recognizes](#) on page 351 for more information on environment variables.*

Import a Sequence of Files

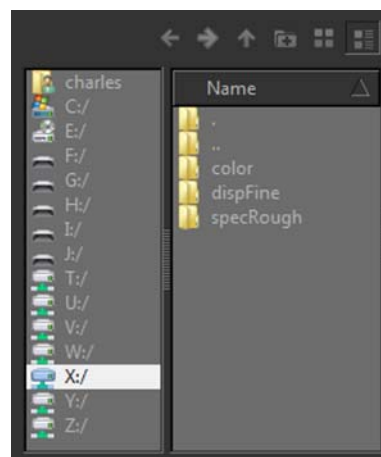
If your patches to import have a consistent naming/numbering system, you can import them as a sequence.

1. Click on 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 that this isn't the name of the files themselves, it's the directory in which the sequence of files exists.) 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:



3. When you've selected the path, the file area shows all the existing image sequences under that path.

4. You can use the **Filter** field to narrow down the list of textures shown. Enter filter text, and the display updates to only show the textures with that text in their name.
5. 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, you'll see the **File Example** update 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.

Note *Mari won't necessarily overwrite an existing sequence if you select an existing template. For example, if the template includes the "\$CHANNEL" tag, importing only overwrites an existing sequence for a channel with the same name.*

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.
- \$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.
- \$[METADATA VALUE] — any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project 1", Mari replaces the variable "\$PROJECT" with "Project 1").

Note *For more information on how the UDIM patch values are allocated, please see the section on [Using UDIM Values](#) on page 178.*

Note *If you choose to import a layered .psd file, layers are imported directly onto channels of the same name. If no existing channel name matches a layer name, a new channel with that layer name is created, and the image data is imported onto it.*

You can now [Set Resize Details and Import](#) the textures.

Set Resize Details and Import

1. From the **Resize** dropdown, select how to handle importing images onto patches of different resolution:
 - **Patches** — resizes the patch to match the resolution of the imported image.
 - **Images** — resizes the image to match the patch it's being imported onto.
2. Click to import the images. If you have one or more patches selected, you have the options of importing onto **All** patches in the channel or to the **Selected Only**.

Mari imports the images and loads them into the selected channel.
3. If you are importing onto all objects in your project, the **Import** dialog box displays, allowing you to import onto the next object in your project.

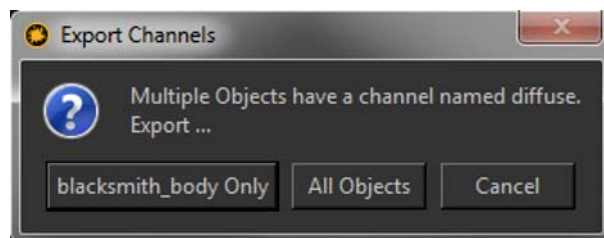
Export Textures

Once you've finished working on your textures, to export the files (for example, for use in the next stage of the pipeline):

1. If you only want to export some of the patches, select the ones you want to export.
2. From the **Channels** menu, select **Export**.

Tip *This is also available as an option from the right-click dropdown menu on the main canvas and in the **Channels** palette.*

If you have multiple objects, and other objects have channels with the same name as the channel you are exporting from, Mari asks whether to export only from the current object, or all objects:



3. Click to select where to export from.

*The **Export Channel** dialog box opens.*

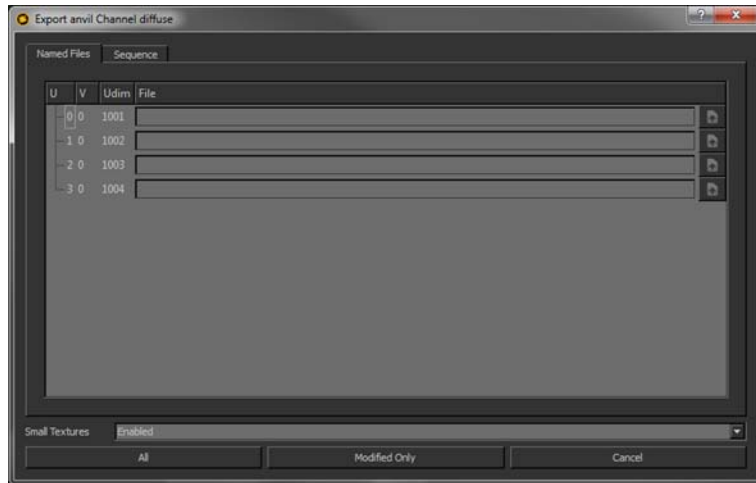
Now you can choose to [Export Individual Files](#), or [Export a Sequence of Files](#).

Export Individual Files

If the object has less than 20 UV patches, the **Export** dialog box has a **Named Files** tab. This lets you individually pick which patches to export to which files.

1. Click on 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 find an existing file to overwrite.

You can now [Set Small Texture Preferences and Export](#) the textures.

Tip *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](#) on page 351 for more information on environment variables.*

- Edit this line manually. As you edit, you'll see the **File Example** update 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.

Note *Mari won't necessarily overwrite an existing sequence if you select an existing template. For example, if the template includes the "\$CHANNEL" tag, exporting only overwrites an existing sequence for a channel with the same name.*

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.
- \$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.
- \$[METADATA VALUE] — any user-defined variables (for example, if a patch has a user attribute called "PROJECT", with a value of "Project 1", Mari replaces the variable "\$PROJECT" with "Project 1").

Exporting to a Layered .psd File

If you want to export the channels in your project to a layered .psd file, simply change the file extension to .psd in the **Export** dialog. This exposes a **Channels as Layers** window to the right of the dialog where you can select the channels you would like to export as layers. You can do this for both **Named Files** and a **Sequence** of files.

Note *When exporting multiple channels to a layered .psd file, all channels must have the same resolution and bit depth.*

The \$CHANNEL Template environment variable does not work when exporting multiple channels to a layered .psd file.

You can now [Set Small Texture Preferences and Export](#) the textures.

Sandbox *Experiment with the different Template variables, and see how the File Example changes to show how Mari generates exported filenames.*

Set Small Texture Preferences and Export

1. Select whether to enable **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.

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).

2. Click to export **All Textures** (the entire channel), the **Modified Only** (all patches that have changed since the last export), or **Selected Only** (only the currently selected patches).

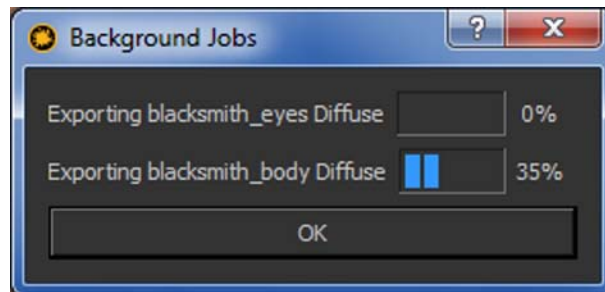
Mari exports your textures to disk.

3. If you are exporting from all objects in your project, the **Export** dialog box displays, allowing you to export from the next object in your project.

Mari exports your textures in the background.

4. To check the progress of the export click the  icon at the bottom right of the main window.

Mari shows the export progress.



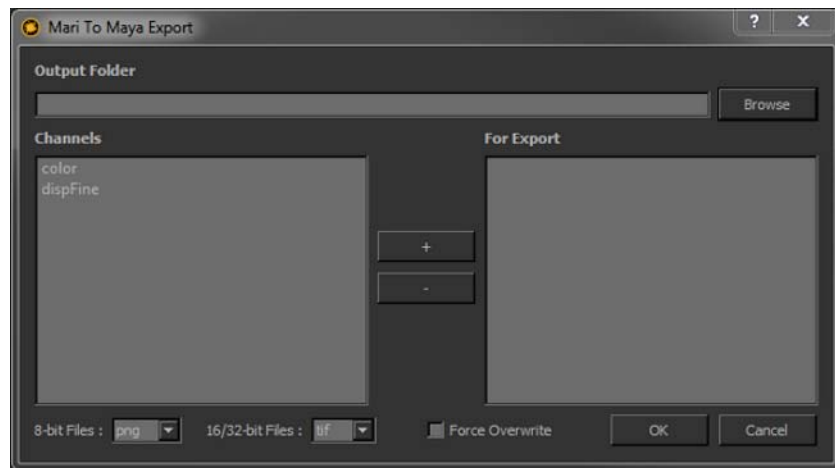
Export to Maya

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.



14 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 six shelves:

- **Menu** — items that you can select from the F9 pie selection menu control.
- **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.
- **Project** — items just for the current project.

You can also create any number of customized shelves.

Your shelves display in both the **Shelf** palette and on the **Shelves** tab in the **Brush Editor** (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 details on using the pie selection menu control, see [Using Shortcuts](#).

Steps for Using Shelves

To use shelves within Mari, follow these steps:

1. [Open Shelves](#)
2. [Add a Brush to a Shelf](#)
3. [Add a Color to a Shelf](#)
4. [Add an Image to a Shelf](#)
5. [Create a Custom Shelf](#)
6. [Modify Custom Shelves and Their Contents](#)
7. [Import and Export Shelves and Items](#)
8. [Create a New Shelf Palette](#)
9. [Add Any Mari Function to a Shelf](#)

Open 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.*



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

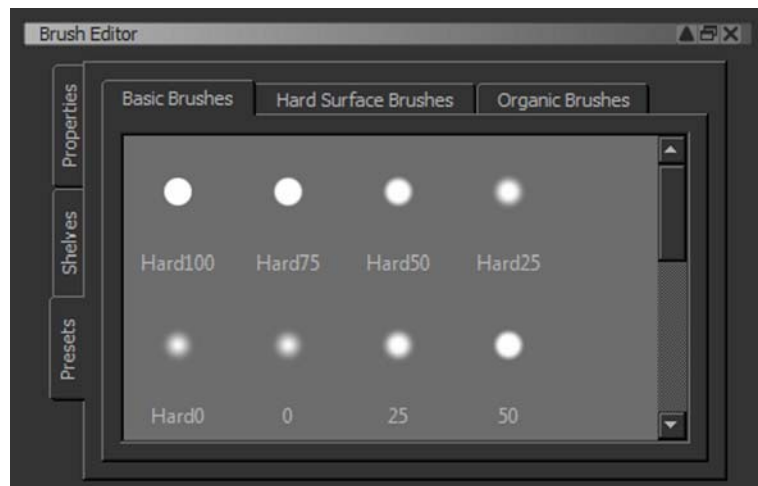
*As described above, the **Shelf** starts with six default shelves: **Menu**, **Personal**, **Basic Brushes**, **Hard Surface Brushes**, **Organic Brushes**, and **Project**. The **Menu** and **Personal** shelves are empty when you first start Mari. The **Project** shelf only displays when you have a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

Add a Brush to a Shelf

To add a brush to a shelf:

1. Open the **Brush Editor** palette:
 - from the **View** menu, select **Palettes > Brush Editor**, or
 - right-click in the toolbar area and select **Brush Editor** from the drop-down menu.

*The **Brush Editor** palette displays.*



2. On the **Presets** tab, click to display the tab in the **Brush Editor** with the brush you want to copy.
3. Click the shelf in the **Shelf** palette you want to copy the brush to (**Personal**, **Project**, or [Create a Custom Shelf](#)).
4. Drag and drop the brush from the **Brush Editor** to the **Shelf** palette.
You now have easy access to that brush whenever you want to use it.

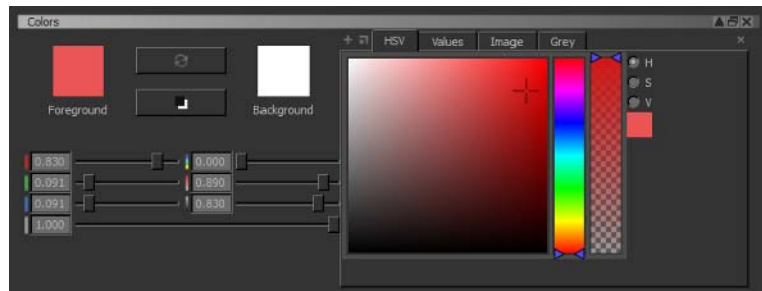
Tip *You can also change the properties of a preset brush, and then save it to your **Personal** shelf, using the **Brush Editor** — see [Configuring Brushes](#).*

Add 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. Click the shelf in the **Shelf** palette you want to copy the color to (**Personal**, **Project**, or [Create 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).*

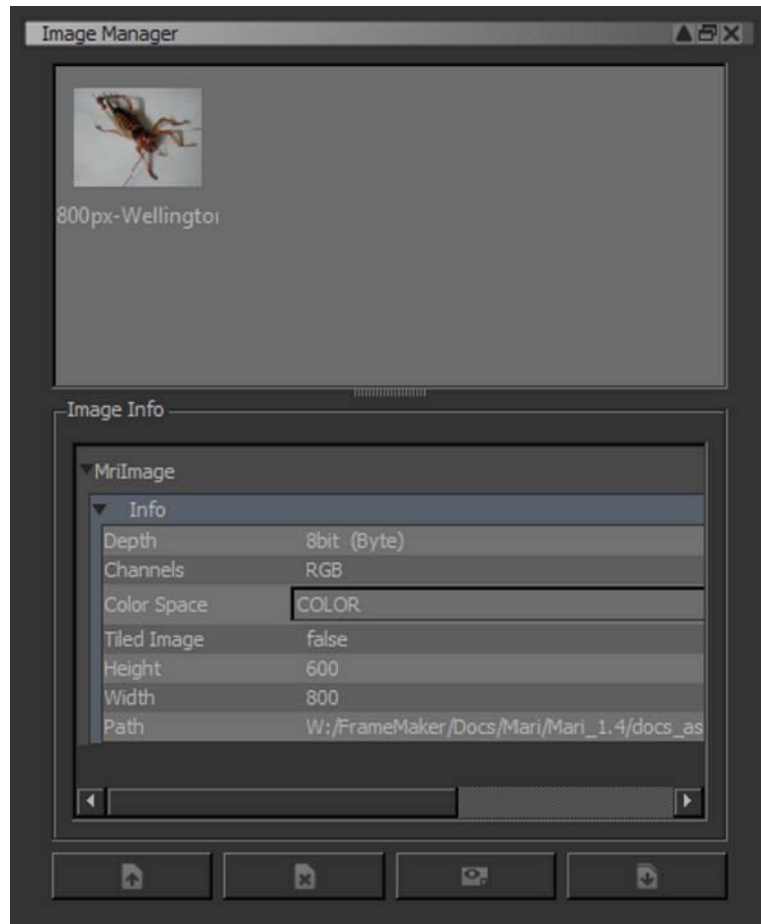
*You can display a large array of color swatches in a shelf by minimizing the icon size: right-click anywhere in the shelf, and select **Toggle Icon Size** from the dropdown menu. Icons switch from large to small (or vice versa).*

Add 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.*



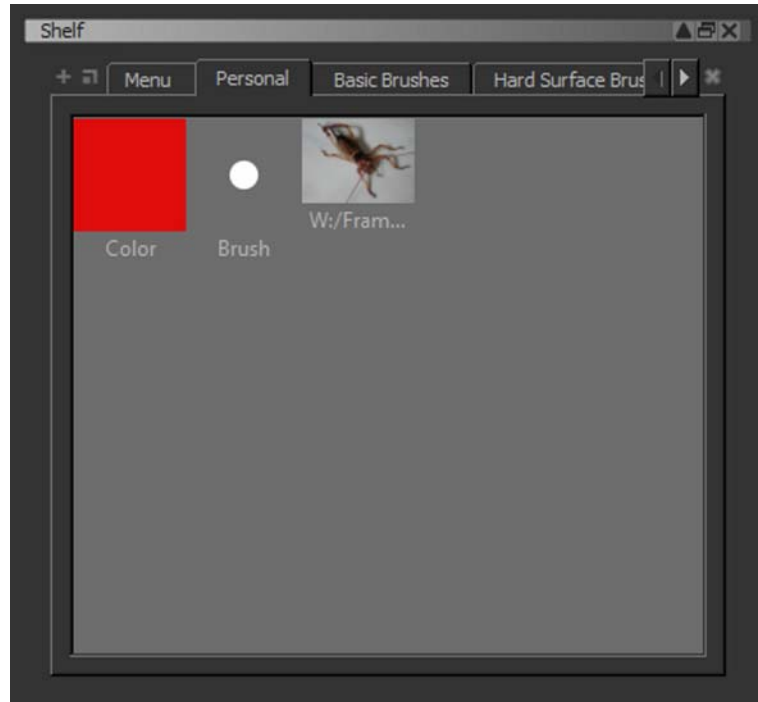
Tip *The default layout in Mari has the **Image Manager** palette already open. (Click to give it focus.) See [Managing Images in Mari](#) for instructions on loading images into the **Image Manager**.*

2. Click the shelf in the **Shelf** palette you want to copy the color to (**Personal**, **Project**, or [Create 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.*

*Below is an example of a color, brush, and image copied to the **Personal***

shelf.

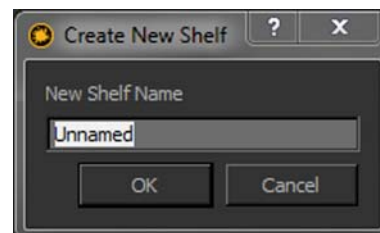


Tip *You can rename an item in a shelf, by double-clicking its name and typing over. You can also drag any item from any shelf to the **Personal**, **Project**, or custom shelves.*

Create a Custom Shelf

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

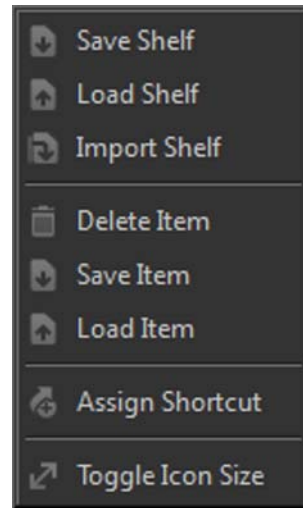
1. In the **Shelf** palette, click  at top left.
*The **Create New Shelf** dialog box displays.*



2. Type a name for the shelf and click **OK**.
*The new shelf displays at right in the **Shelf** palette.*

Modify Custom Shelves and Their Contents

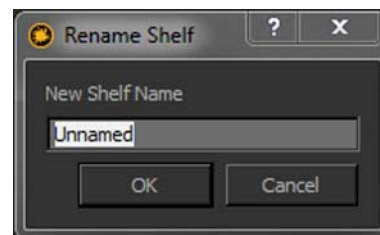
The **Shelf** palette includes a right-click dropdown menu with options to rename, delete items from, and assign shortcut keys to items in the **Personal**, **Project**, or custom shelves. (It also includes features to [Import and Export Shelves and Items](#).)



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.

Sandbox *As a learning exercise, try this:*

- *create a custom shelf*
- *rename it*
- *add an item to it*
- *rename the item*
- *delete the item*
- *and then delete the shelf!*

Import and Export 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 [Import a custom Photoshop brush](#) on page 42.*

Sandbox *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)*
- *delete both shelves!*

Create a New Shelf Palette

To turn any shelf into a custom palette:

- With the shelf selected, click  on the left.
Mari creates a new palette for that shelf.

Tip *Mari adds "spawned" palettes to the **View > Palettes** and **Palettes** menus.*

- If you rename a palette, Mari renames it in all places (**Shelf** palette, spawned palette, and menus).*

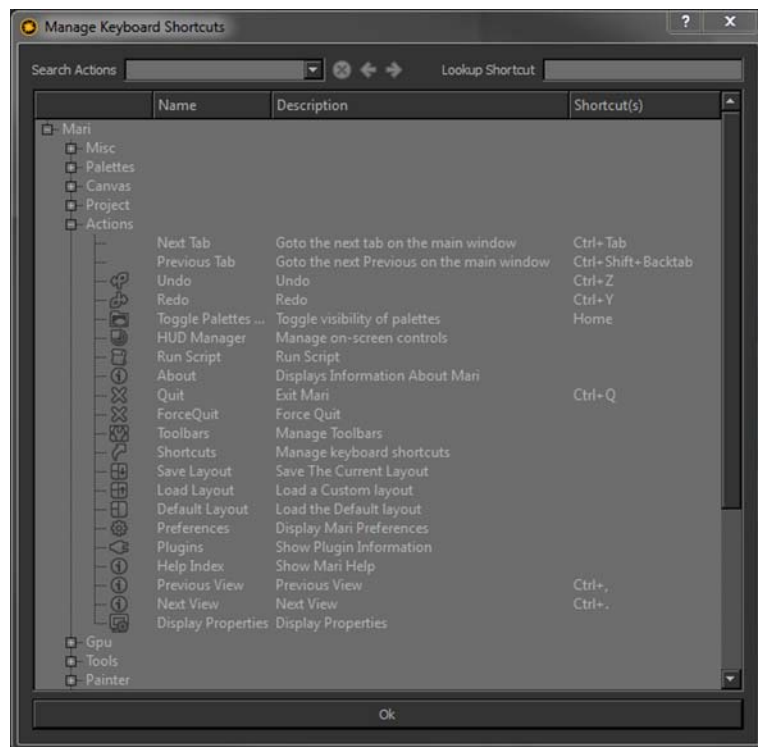
Sandbox *Spawn a couple of shelves and drag items between them. Try adding an item to the spawned palette and see how this affects the original shelf.*

Add Any Mari Function to a Shelf

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

- 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.



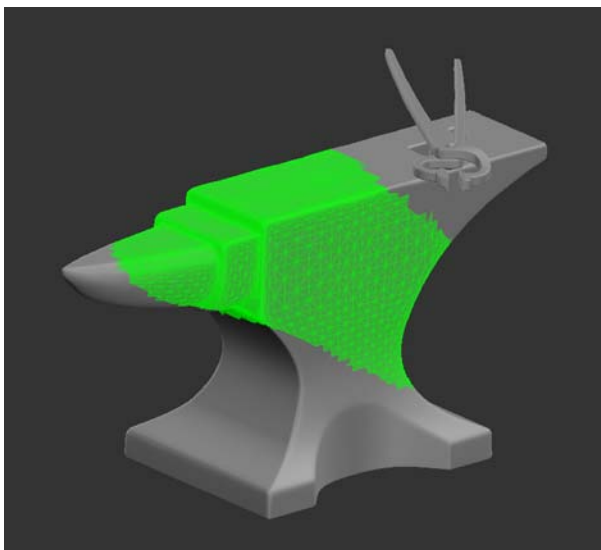
15 SELECTING ITEMS

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.

Selection Tool

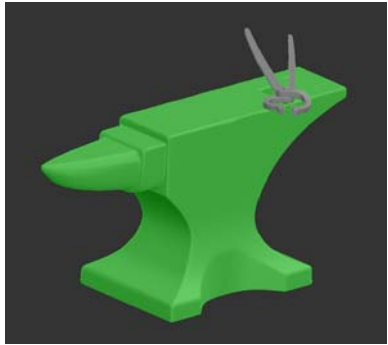
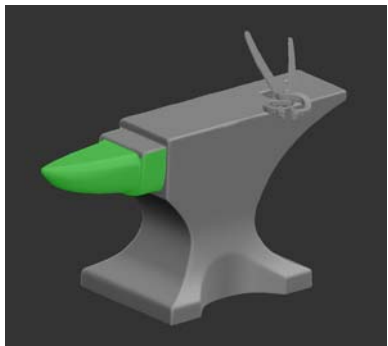
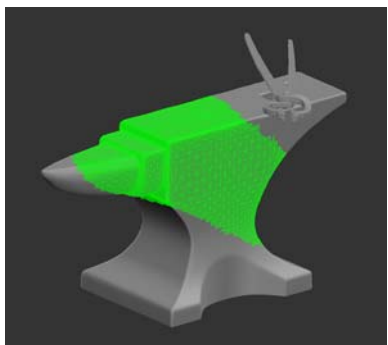
The selection tool lets you select areas on the objects in your project. Once you have made a selection, you can choose to show or hide the selected areas, or to lock them for editing.



Area selected on the model.

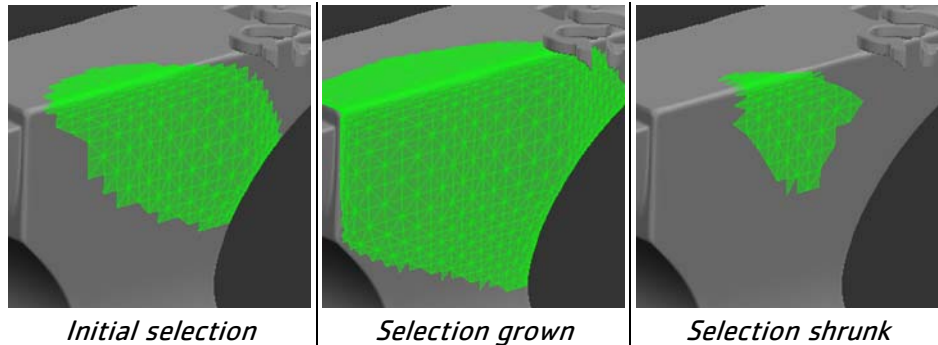
Selection Modes

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 areas on the objects. You can use standard marquee and loop select tools to select parts of your objects.	

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.



Smart Selection

The smart selection mode of the Select tool 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
- Edge Angles
- Orientation

Visibility and Locking

You can show or hide things based on whether they are selected. 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.

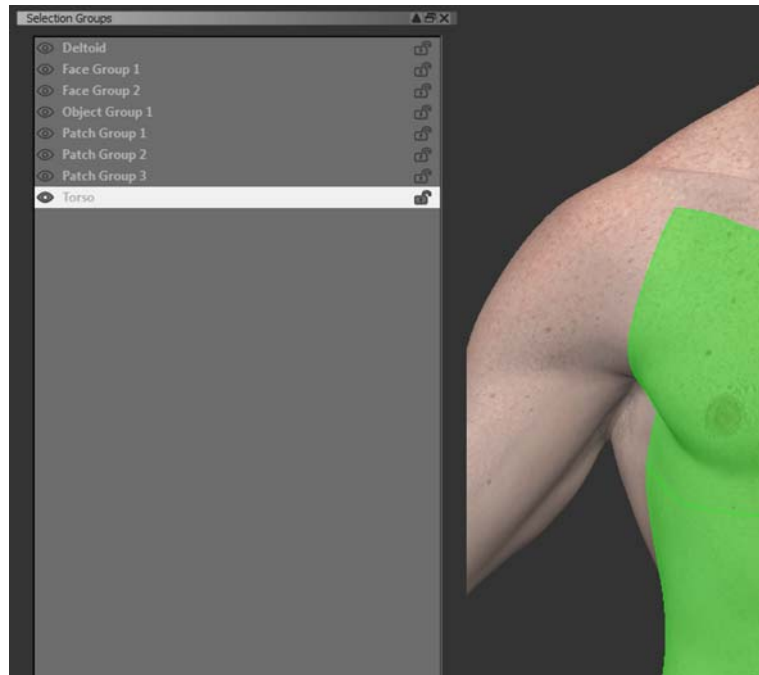
Selection and Painting

You can use the selection to control where paint can bake onto the surface of your model. If **Project On** (on the **Projection** palette) is set to **Selected**, then Mari only bakes paint onto selected areas.

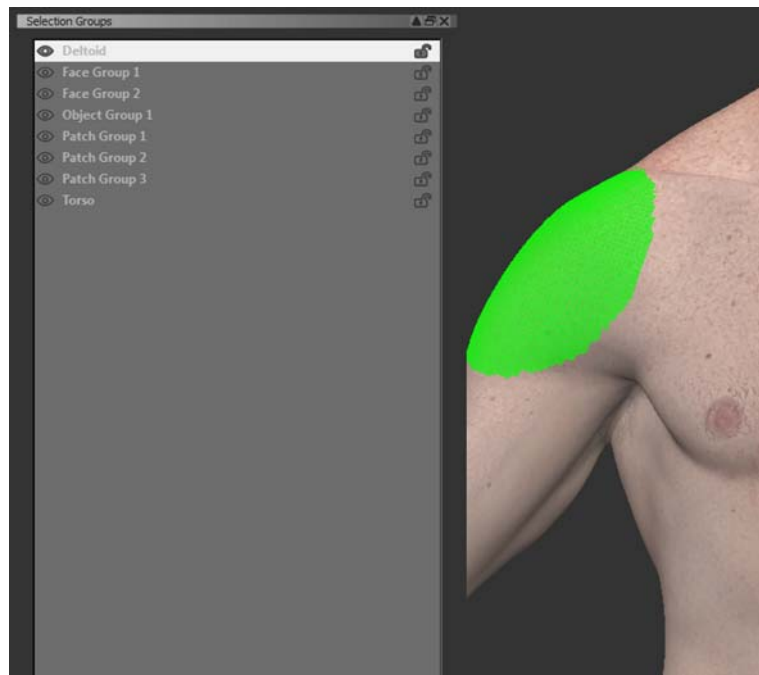
Saving Selected Areas

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.



Example of a selection group containing patches.

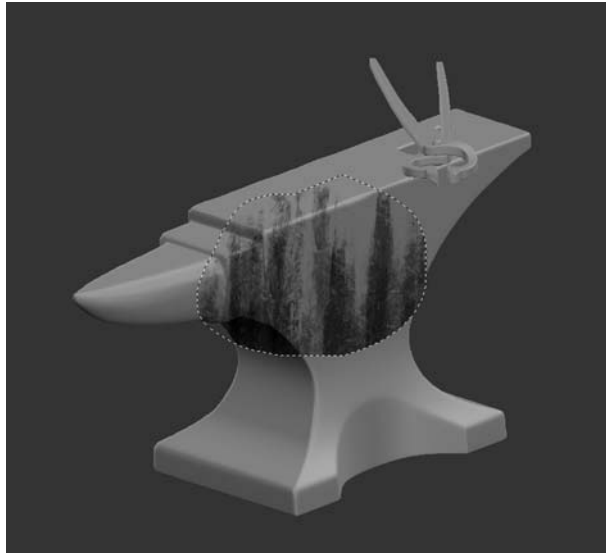


Example of a selection group containing faces.

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.

Marquee Select Tool

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 **Project On** is set to **Selected**, except that you're selecting areas on the screen rather than on your model. You can't use the **Marquee Select** tool to select areas to show/hide or to lock.*

Steps for Making Selections

To make a selection within Mari, follow these steps:

1. [Selecting Objects, Patches, or Faces](#)
2. [Saving Selections on the Model](#)
3. [Selecting Areas Onscreen](#)

Selecting Objects, Patches, or Faces

1. Start Mari and open your project.
2. Choose the **Select** tool .
3. You can use the **Selection** toolbar to switch selection modes.



*If you can't see the **Selection** toolbar, right-click on the toolbar area and choose **Selection** 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:

- **Facing** — whether to only select the (visible) front of the model, or to select “through” the model, including the back side.
- The selection tool shape – either:
 -  a square selection box,
 -  a lasso selection (allowing you to draw any shape of selection),
or
 -  smart selection (letting you select areas based on the shape of the model).

*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 invert the selection:

Hold down...	To...	Effect
Shift	Add to the selection	Areas that were unselected become selected, areas that were already selected stay selected.

Hold down...	To...	Effect
Ctrl	Remove from the selection	Areas that were selected become unselected, areas that were already unselected stay unselected.
Meta (Windows)	Intersect the selection	Areas that overlap with the current selection become unselected, areas that were unselected become selected.

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.*

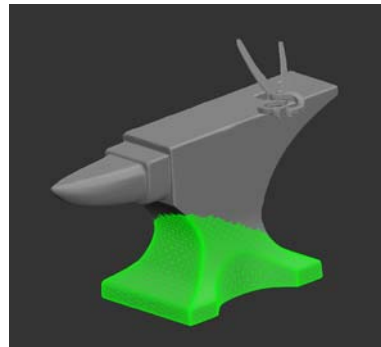
The **Smart Selection** mode works slightly differently from the other two modes. To use it:

1. In **Face Selection** mode, click .
2. In the tool properties toolbar, set the **Smart SelType**. This sets how to create the selection area:
 - **Connectedness** — 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 SelAngle**.
 - **Orientation** — selects the area that faces in the same direction as your initial selection.
3. Set the **Smart SelAngle** — 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.

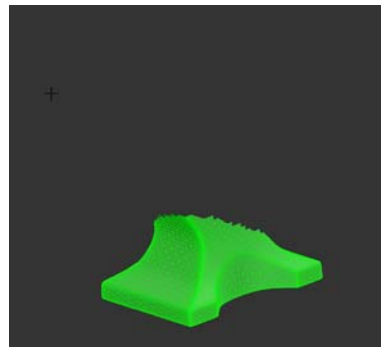
- For **Orientation**, this sets how far from the facing of your original selection the selection goes to. For example, if the angle is set to 30, this selects areas up to 30 degrees away from the facing of your original selection.
4. Click and drag to make the selection.
Mari creates a selection around the area you drag.

Hiding and Locking

1. In either the **Patch** or **Face** selection modes, select an area on your model.



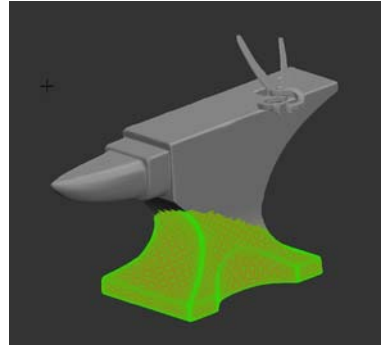
2. From the **Selection** menu, select **Hide Unselected**.
Mari hides everything except your selection.



You can clear the selection, and the other areas remain hidden.

3. To show the whole model again, from the **Selection** menu, select **Show All**.

4. To lock the selection so that you can't paint on the locked areas, from the **Selection** menu, select **Lock Selected**.



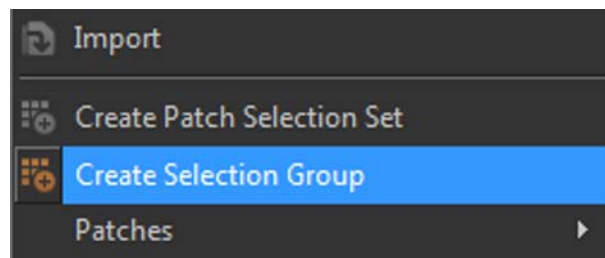
Mari locks the selection. Try painting a few strokes over the model — the paint does not "stick" to the locked areas.

5. To unlock the selection, from the **Selection** menu, select **Unlock Selected**.

Tip *You can also show/hide and lock selections from the right-click menu on the model.*

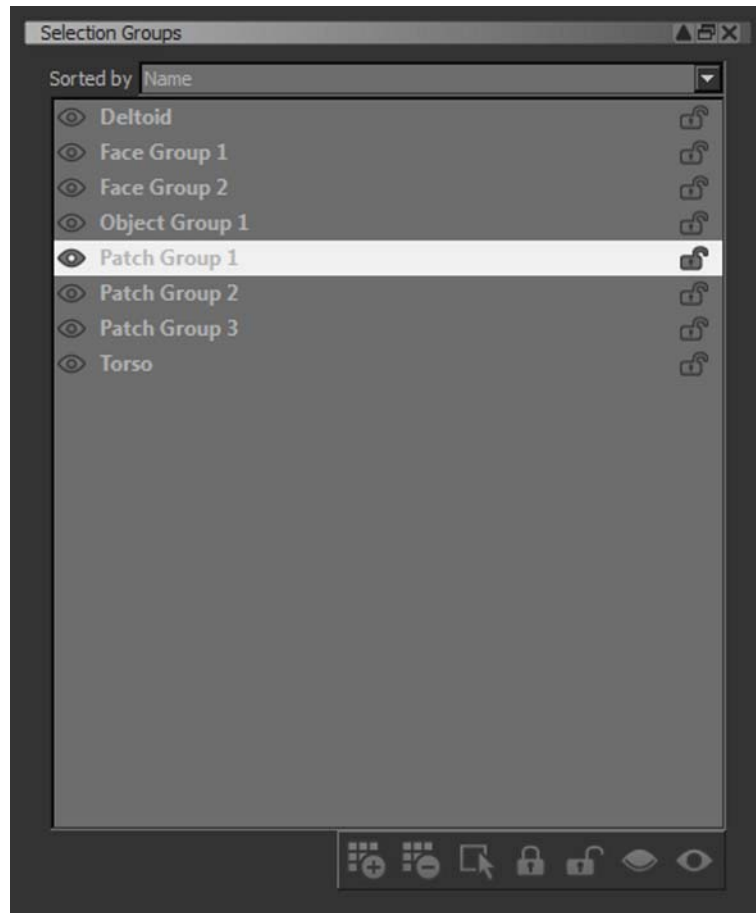
Saving Selections on the Model

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.
Object groups can be added to selection groups. You can create up to 200 of these groups.
5. 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.

6. 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 choosing **Create Patch Selection Set**. This can also be done in the **Patches** palette by right-clicking on your selection and choosing **Create Set**.

Selecting Areas Onscreen

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, or resizing it.

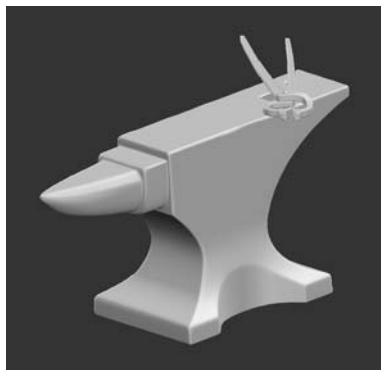
1. Paint a few strokes onto your model, so you can see the effects when you start drawing marquee selections.
2. Select the **Marquee Select** tool .
3. In the **Tool Properties**, select the tool type. You can select a rectangle, lasso, ellipse, or wand.



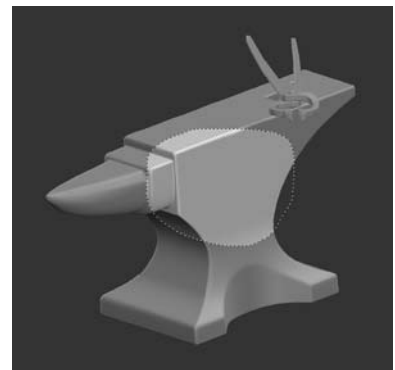
Tool	Lets you
	Draw a rectangular selection.
	Draw the outline of your selected area. You can draw any shape you want.

Tool	Lets you
	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.

4. Click and drag to create your initial selection.



Initial buffer, with unbaked paint



Marquee select — only the paint within the selection is visible

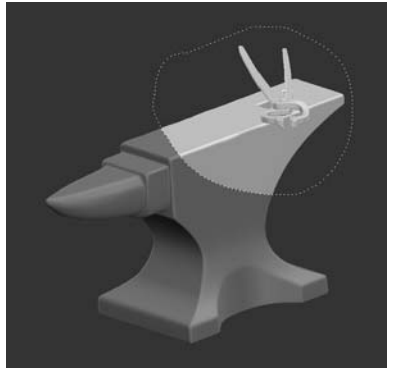
5. To edit the shape of the selection, change the **Mode** in the **Tool Properties**. You can set this to:

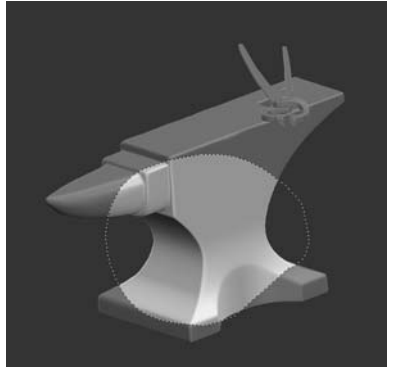
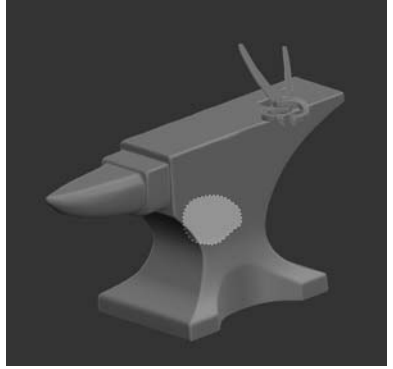
Mode	Does	Notes
Replace	the new selection replaces the existing selection.	In Replace mode (the default), you can: • hold down Shift, click and drag to add to the current selection, or • hold down Ctrl, click and drag to remove from the current selection.
Transform	lets you move your selection area around (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, click and drag to rotate the current selection, or hold down Shift+Ctrl , click and drag to scale the current selection.

6. 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.
7. 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	

To	Do this:	Example
rotate the selection	hold down Ctrl , click on your desired point of rotation, and drag	
scale the selection up or down	hold down Shift+Ctrl , click and drag	

16 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.

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.

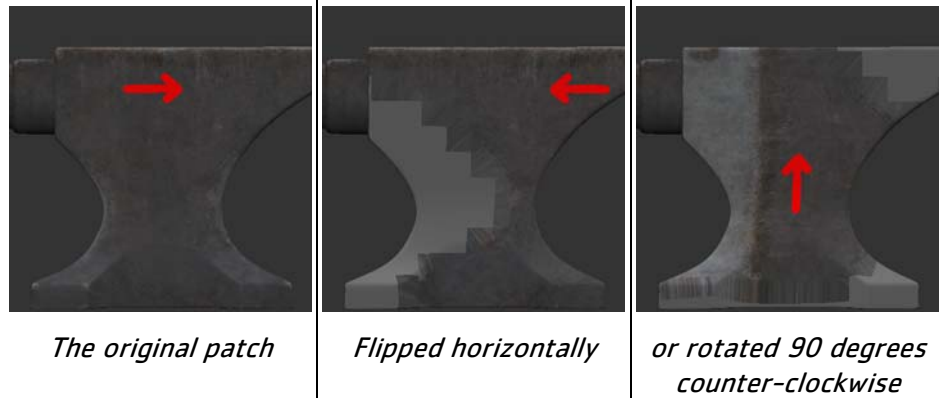
Tip *For more details on working with channels, see [Channels](#).*

Changing Patches as a Whole

Mari lets you change patches as 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.



Tip *Edits to patches only apply to baked paint. Unbaked paint in the paint buffer is not affected.*

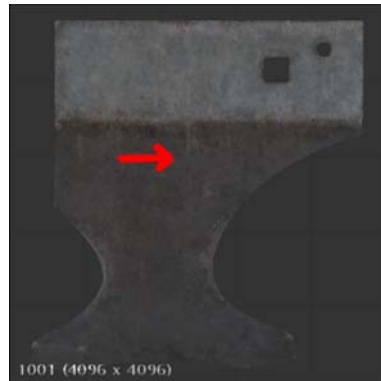
When you are copying patches onto other patches, unless your patches are identically shaped, you can see:

- 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.

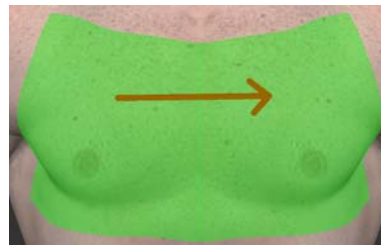


The original patch.



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.

Saving Images of UV Masks on a Patch

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.

Bleeding 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 demo 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.



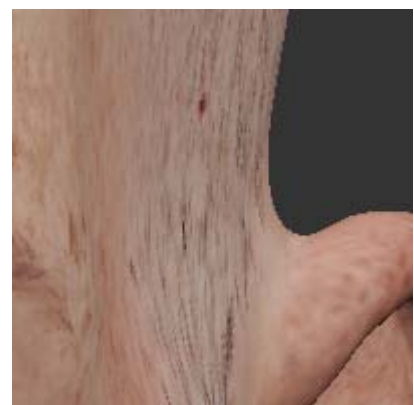
This is caused by a too small overspill edge.

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.

Note *If you check the **Bleed** checkbox when baking a shader, Mari uses the per tile method to generate bleed for the baked shader.*

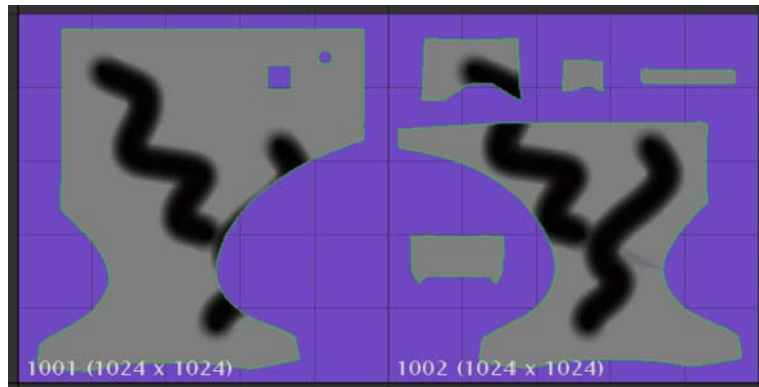
Edge bleeding per patch is different because it renders the entire patch first and then performs a full bleed on it. This avoids the conditions mentioned previously but requires much more memory and processor time, especially when bleeding large patches.

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.

Linked 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 to directly.*

Steps for Working with Patches

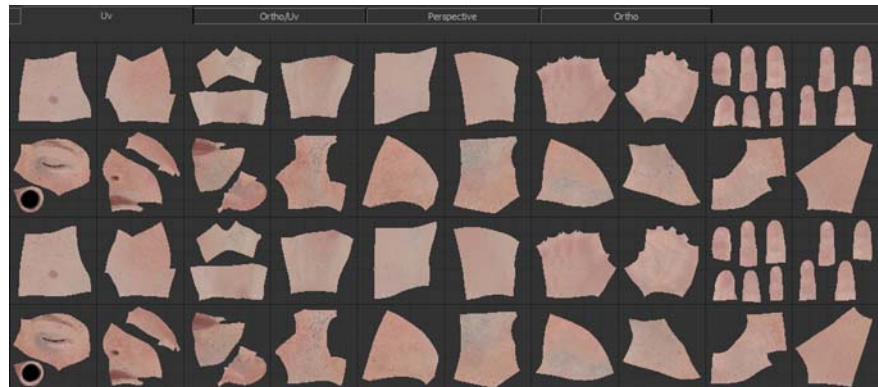
To manage the patches in your Mari project, follow these steps:

1. [View and Resize Patches](#)
2. [Using UDIM Values](#)
3. [Copy and Paste Patches](#)
4. [Extract Patches to the Image Manager](#)
5. [Fill Patches with Color](#)
6. [Flip and Rotate Painting](#)
7. [Mirror Painting](#)
8. [Save Images of the UV Masks](#)
9. [Bleeding Across Patch Edges](#)
10. [Link or Unlink Patches](#)

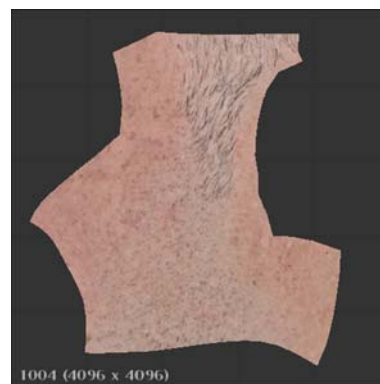
View and Resize Patches

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.



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).



Resize 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, 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.*

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:

$$\text{udim} = 1001 + u + (10 * v)$$

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:

$$\text{udim} = 1001 + 2 + (10 * 5) = 1053$$

(...)									
(0,1)									
(0,0)	(1,0)	(2,0)							(9,0)

Copy and Paste Patches

See [Copying and Pasting Paint](#) for details on how to copy and paste individual patches.

Extract 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**.

*Mari copies the patches to the **Image Manager**. See [Managing Images in Mari](#).*

Tip *You can only copy patches up to 8K resolution.*

Fill 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 in Mari](#) for details on selecting colors.

2. Select the patch(es) to fill.
3. From the **Patches** menu, select either:
 - **Fill Foreground** — to fill the patches with the foreground color
 - **Fill Background** — to fill the patches with the background color
 - **Fill Transparent** — to make the patches transparent.

Mari fills the patches with the selected color or transparency.

Flip and Rotate 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.

Mirror 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.

Sandbox *Experiment with flipping, rotating, and mirroring patches. Apply paint first to see what happens to the paint when the patch is changed.*

Save Images of the UV Masks

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 in Mari](#) for details on using the **Image Manager** palette.*

Bleeding Across Patch Edges

If you are seeing 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.

Link or Unlink Patches

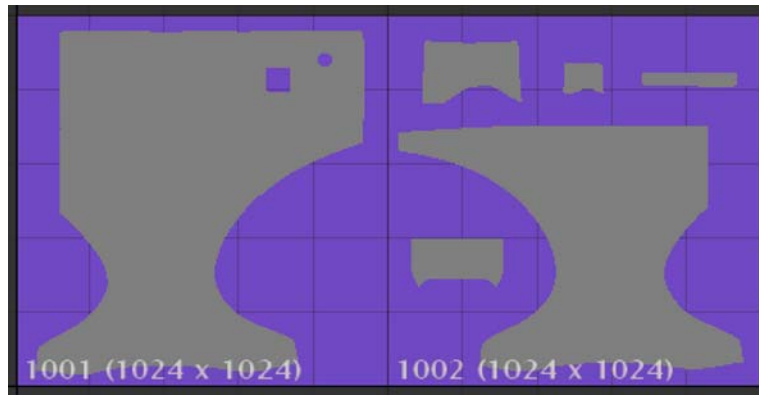
To link together two or more patches:

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 channel 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 patch or patches that you want to unlink from the linked selections.
2. Click the **Unlink Selected Patch Images** icon to unlink the selected patch from any other patches.

In UV view, the colored background no longer shows behind the unlinked patch. Any other linked patches still show the colored background behind them.

3. If you want to unlink all patches in the project, 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**.
4. 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.*

17 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 choose 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 and error warnings.

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.

You can choose between two sizes for the icons in toolbars and palettes. The default size makes it easy to see at a glance where each function is, while the small icons take up less space on screen.

Steps for Customizing Your Workspace

To customize your workspace, follow these steps:

1. [Choose and Organize Palettes](#)
2. [Choose and Organize Toolbars](#)
3. [Create Custom Toolbars](#)
4. [Configure the HUD](#)
5. [Save Your layout](#)
6. [Understanding the Status Bar](#)

Choose and Organize Palettes

To select and position palettes:

1. To choose 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 top, and toolbars below.

Tip *When you open a palette, it displays where it was the last time it closed.*

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 palette function: press the **Home** key (or select from the **View** menu) to toggle between showing and hiding the current palettes.*

Sandbox *Palette behavior is customizable. Experiment with the palettes to see how they can be docked, undocked, moved, and stacked.*

Choose and Organize Toolbars

To choose and position toolbars:

1. To choose 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).

Tip *The default toolbars are:*

- **Project** — common project functions (New project, Open, Save, Export, 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 or open the Brush Editor.
- **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.
- **Tool Properties** — lets you set the properties for the current tool.

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.

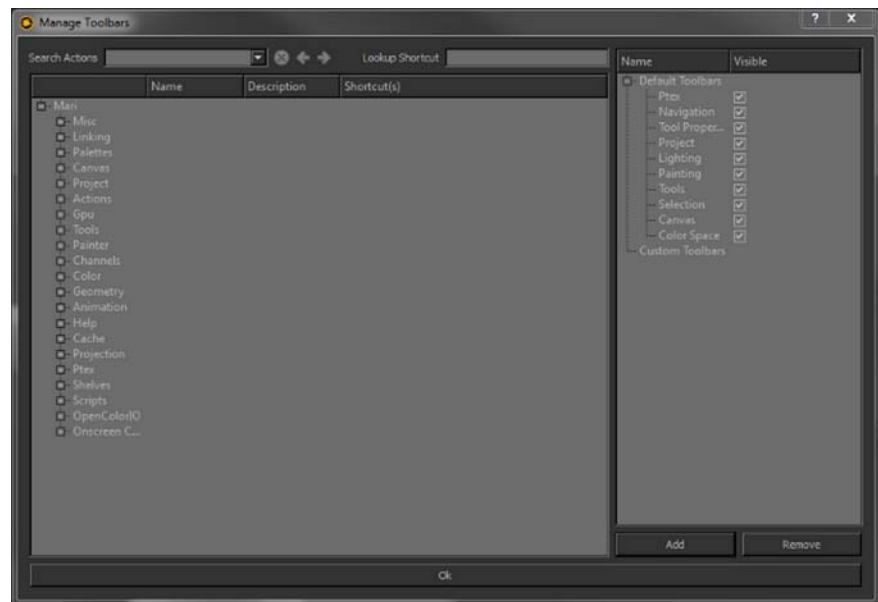
Tip *If the toolbar is too small to fit all of its items, a double chevron appears at the end (»). Click this to see all the tools in the bar, or resize and position the toolbar to see all items.*

Create 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.

Sandbox *Try this:*

- *Create a new toolbar.*
- *Add a few items to it.*
- *Resize it.*
- *Undock it, move it, and re-dock it somewhere else.*
- *Hide it from the display.*
- *Remove it.*

Configure the HUD

To configure the “heads-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**.*

Save 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 choose 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**.*

Sandbox *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, and features that can be toggled.

All icons display tooltips when the cursor hovers over the icon. Some can be clicked to toggle the displayed functionality. Though most icons are unique to a single error warning or feature, there are a few noted in the list below that are not.

Status Bar Icons

Icon	Description	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 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 color management is active. More information on color management can be found under Set how colors are displayed using the Color Manager on page 204.	

Icon	Description	Notes
	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 under the Projectors on page 294 chapter.	
	Notifies you that you currently have edge masking enabled. Clicking on this icon disables edge masking. More information on edge masking and the edge masking icon can be found under Edge Masking on page 84.	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. Clicking on this icon disables channel masking. More information on channel masking and this icon can be found under Channel Masking on page 85.	
	Notifies you that you currently have ambient occlusion masking enabled. Clicking on this icon disables ambient occlusion masking. More information on ambient occlusion masking and this icon can be found under Ambient Occlusion Masking on page 86.	
	Notifies you that you currently have depth masking enabled. Clicking on this icon disables depth masking. More information on depth masking and this icon can be found under Depth Masking on page 87.	
	Notifies you that you currently have backface masking enabled. Clicking on this icon disables backface masking. More information on backface masking and this icon can be found under Backface Masking on page 87.	
	Notifies you that you currently have fractal noise masking enabled. Clicking on this icon disables fractal noise masking. More information on fractal noise masking and this icon can be found under Fractal Noise Masking on page 88.	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 Mask What You Can Paint On on page 82.	
	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 cache 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 used in the current shader module. • 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 on which Mari is listening for commands.
	Notifies you that there was an error while Mari was listening for commands.	



18 MANAGING IMAGES IN MARI

Mari lets you use external images as a source for:

- clone stamping
- painting through
- picking colors in the **Color Manager**.

Mari has a palette, the **Image Manager**, which stores images 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.

When you open a project, the **Image Manager** loads its images from source files on the 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 also create new images in the **Image Manager** by:

- Cropping them from existing images,
- Running a script on an existing image, or
- Generating them via 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).

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.*

Steps for Managing Images

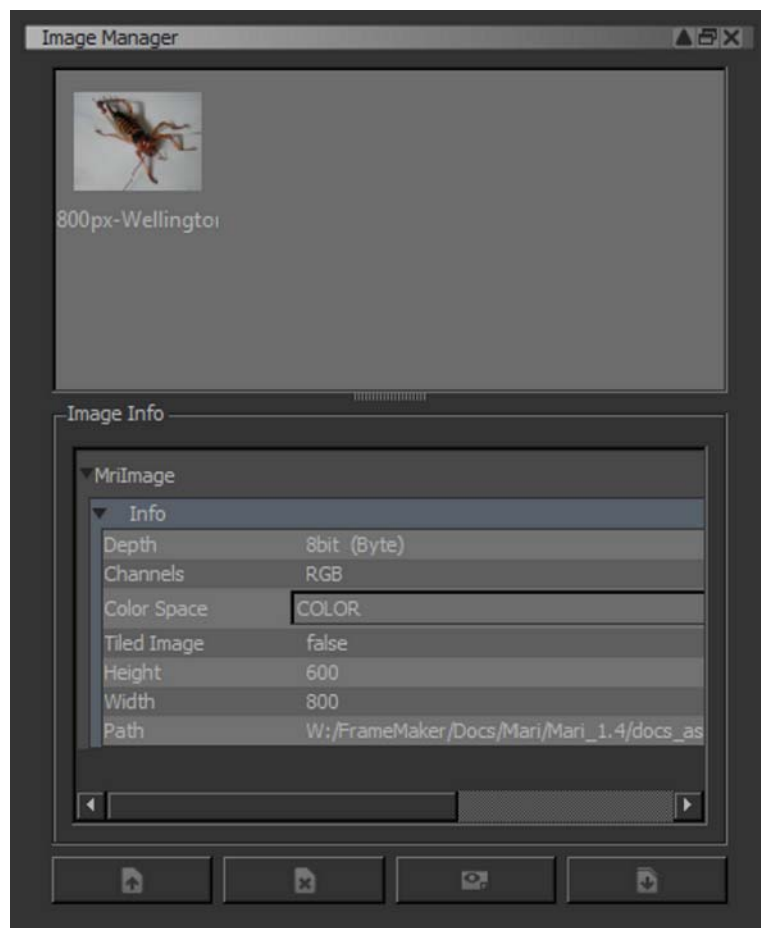
To manage the images in use in your project, follow these steps:

1. [Open the Image Manager Palette](#)
2. [Load and Select Images](#)
3. [Crop and Filter Images](#)
4. [Run Scripts on Images](#)

Open the Image Manager Palette

1. Start Mari and open your project.
2. Open the **Image Manager** palette:
 - right-click on the toolbar, and select **Image Manager** from the drop-down menu, or
 - from the **View** menu, select **Palettes > Image Manager**.

*The **Image Manager** palette displays, showing images currently in use in the project.*



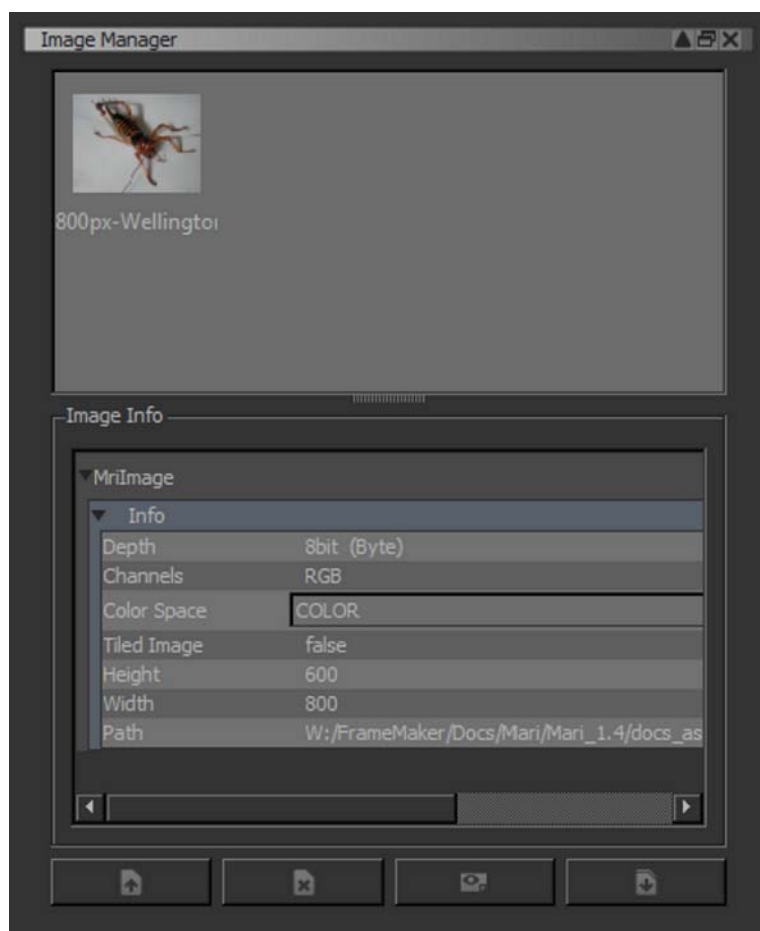
Load and Select Images

1. To load one or more images, click  .
*The **Open an Image** dialog box displays.*
2. Select the image(s) you want to load.

*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.*

3. Under **Advanced Options** select whether you would like to apply a **ColorSpace Conversion (Linear to sRGB)** to the image before you import it, and then click **Open**.
4. Click on a preview to view information about that image.



Information displayed includes:

- Color Depth
- Color Channels
- ColorSpace (which you can set to either **COLOR** or **SCALAR**)
- the image height and width in pixels

- the path for the source file (from which Mari loads the image when you open the project).
5. To remove images from the **Image Manager**, select the image (**Ctrl**-click or click and drag to select multiple images), and click .

*The image(s) disappear 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 .

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 (see the next section).

7. To save an image under a new filename, click .

*The **Save an Image** 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 filename when you open this project.

Tip *You can also **Open**, **Save**, **Save As**, or **Close** images by right-clicking on the **Image Manager** and selecting from the dropdown menu. There is an extra shortcut to open new images as well — double-clicking on an empty part of the **Image Manager**. And you can do all of these on multiple images at once (**Ctrl**-click or click and drag to select multiple images). You can even drag and drop images into the **Image Manager** from your web browser.*

Crop and Filter Images

1. To create a new image by cropping an image from the **Image Manager**, click  to open the image preview window.

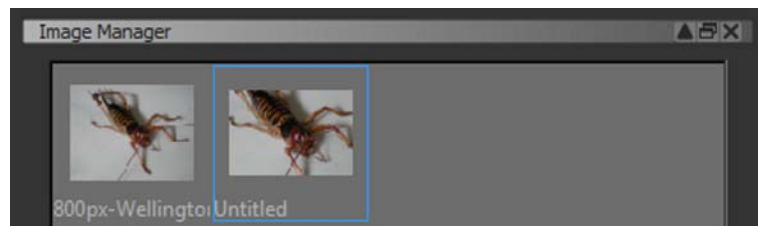
Tip *Clicking on the image in this window sets the foreground color to the color at the spot you clicked.*

2. Choose 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.

- Now click and drag to select the area to crop, choose the **Output Size** and click .

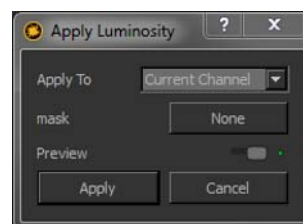


The selected area appears as a new image in the **Image Manager**.



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.*

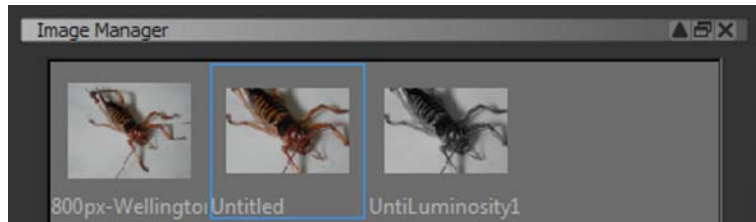
- To apply a filter to an image, click to select it in the **Image Manager** and, from the **Filters** menu, select the filter to apply.
*The **Apply** dialog box displays.*



- 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](#).*

- 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.

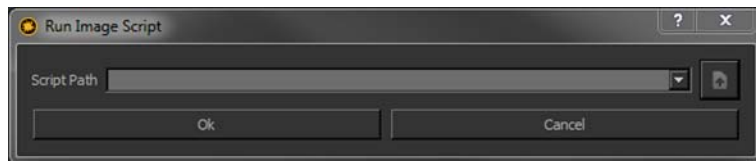
Filtering an image in the Image Manager breaks the connection between the image and the original file. If you want a copy of your updated image on disk, save it with a new filename.

Run 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.

1. To run a script on an image, right-click on the image and select **Run Script** from the dropdown menu.

*The **Run Image Script** dialog box displays.*

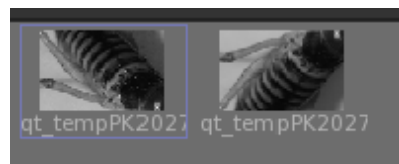


2. Type in the script command to run, or select from the dropdown list. The list holds the last 10 scripts run.

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**.*



Initial image to the left, modified image to the right.

Sandbox *Run some basic scripts on your rock images. For example, try:*

- *`/usr/bin/convert $FILENAME -flip $FILENAME` — to flip the image horizontally.*
- *`/usr/bin/convert $FILENAME -blur 10 $FILENAME` — to apply a slight blur.*

Tip *You can run scripts on multiple images at once. **Ctrl**-click or click and drag to select multiple images.*



19 MANAGING COLORS IN MARI

Mari lets you control both how colors display, and how you select them.

For color (not scalar) channels, you can use the color management functionality provided by the **Color Manager** or the **ColorSpace** toolbar to apply various color profiles (preset or custom), LUTs (look-up tables), or colorspace conversions to the painting displayed on the screen. Mari comes with a set of preset color profiles, and you can configure your own too. Mari also lets you choose a “split point” – where to split the display between corrected and uncorrected colors.

You can also have Mari display an onscreen histogram, showing the color distribution in your painting.

Note *The color management functionality provided by the **Color Manager** and the **ColorSpace** toolbar is for display purposes only. The color profiles, LUTs, and colorspace conversions applied here do not affect the paint on your model. To apply filters to the paint on your model, and bake the changes down, please refer to the [Filtering Images and Paint](#) chapter.*

Mari always has two colors selected, “foreground” and “background”. These display in the **Tools** toolbar, and on the **Colors** palette. By default, painting tools use the foreground color.

For picking colors, Mari provides:

- a standard “eyedropper”-style **Color Picker** tool.
- the **Pixel Analyzer** palette, which supplements the **Color Picker** with a range of colors you’ve selected, showing their average, minimum, and maximum color values.
- the **Colors** palette, which lets you choose colors interactively from a selection of pickers (including HSV, color values, greyscale, blends, and from images) or just by entering specific color values.
- the ability to select colors from images in the **Image Manager** or a shelf.

Tip *You can also change the color depth of a channel if you need to. See [Channels](#) for details.*

About the ColorSpace Toolbar

The **ColorSpace** toolbar is provided as an efficient way to apply colorspace conversions and LUTs to the painting displayed on your screen. While this functionality is also available from within the **Color Manager** (under the

preset **ColorSpace** color profile) the toolbar is conveniently placed to make quick adjustments while painting.

The **ColorSpace** 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 overall color management on/off via a button on the toolbar, and conveniently load custom color configuration files and custom LUTs. There are menus available for applying colorspace conversions, and a number of other controls for finely tuning them.

Steps for Managing Colors

To manage the colors you use in Mari, follow these steps:

1. [Pick Colors from Paint on the Model](#)
2. [Select Colors from a Picker or Image](#)
3. [Set How Colors Are Displayed Onscreen](#)

Pick Colors from Paint on the Model

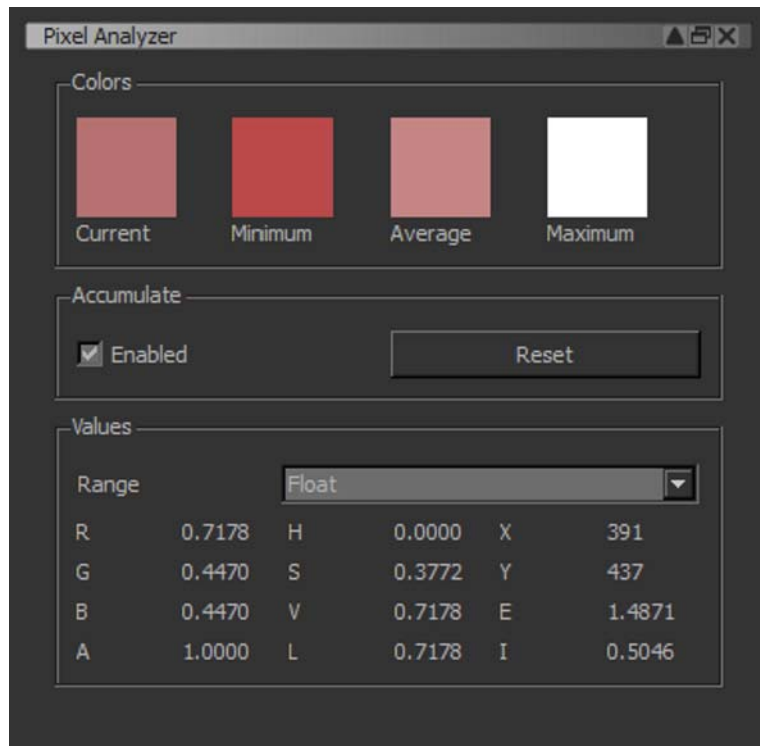
1. Start Mari and open your project.
2. To pick a single color from the paint already on the model, select the **Color Picker** tool , and click on the model.

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.*

3. Open the **Pixel Analyzer** palette:
 - right-click on the toolbar, and select **Pixel Analyzer** from the dropdown menu, or
 - from the **View** menu, select **Palettes > Pixel Analyzer**.

*The **Pixel Analyzer** palette displays details of the currently selected color.*



The **Values** section displays the color's exact numeric value, in the selected **range**.

4. Under **Accumulate**, click the **Enabled** checkbox.

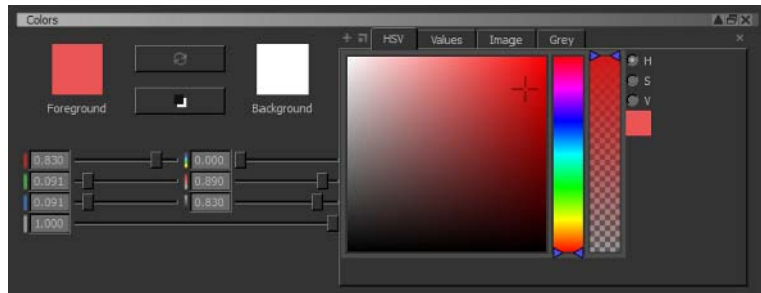
With this option enabled, Mari keeps track of the colors you select. As you click on the model, the **Colors** fields update to display the **Current** color selected, plus the **Minimum**, **Average**, and **Maximum** color values from the tracked series of clicks. (If you drag across the model, Mari 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.

Select 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:

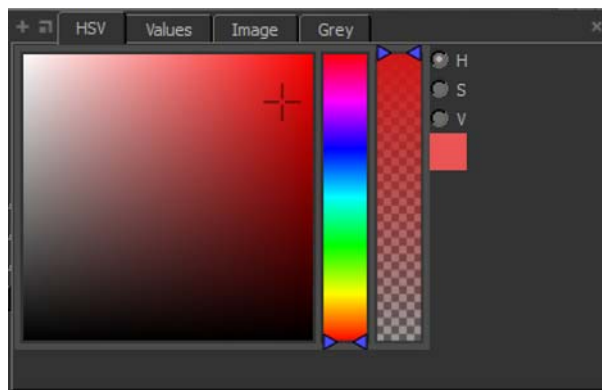
- manually set the numeric values (either type them into the entry box or move the sliders) for the various color attributes — red, green, blue, alpha; or hue, saturation, and value, or
- click a color field in the tabbed area.

The control dynamically updates as you change the settings.

3. Click to navigate through the tabs available for the color pickers.

These include:

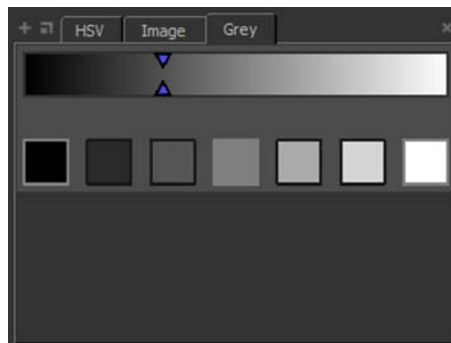
- **HSV** — to select hue, saturation, and value settings.



- **Values** — to pick from **Byte (8 Bit)**, **Half (16 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**).



- **Grey** — to select greyscale values (including predefined grey settings, at the bottom, in steps from black to white).



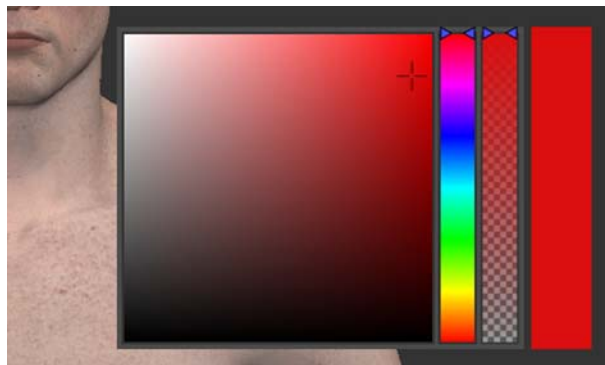
- **Blend** — to pick from a field blending four colors (select which colors by clicking the swatches in the corners to open another color picker control).



Tip *By default, Mari opens with one **HSV** tab, one **Image** tab, and one **Grey** tab. You can add additional color tabs:*

- *To open other tabs, including **Blend** 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 color picker menu when painting, press and hold **j**.
The color picker menu 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).



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.*

- Sandbox**
- *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.*

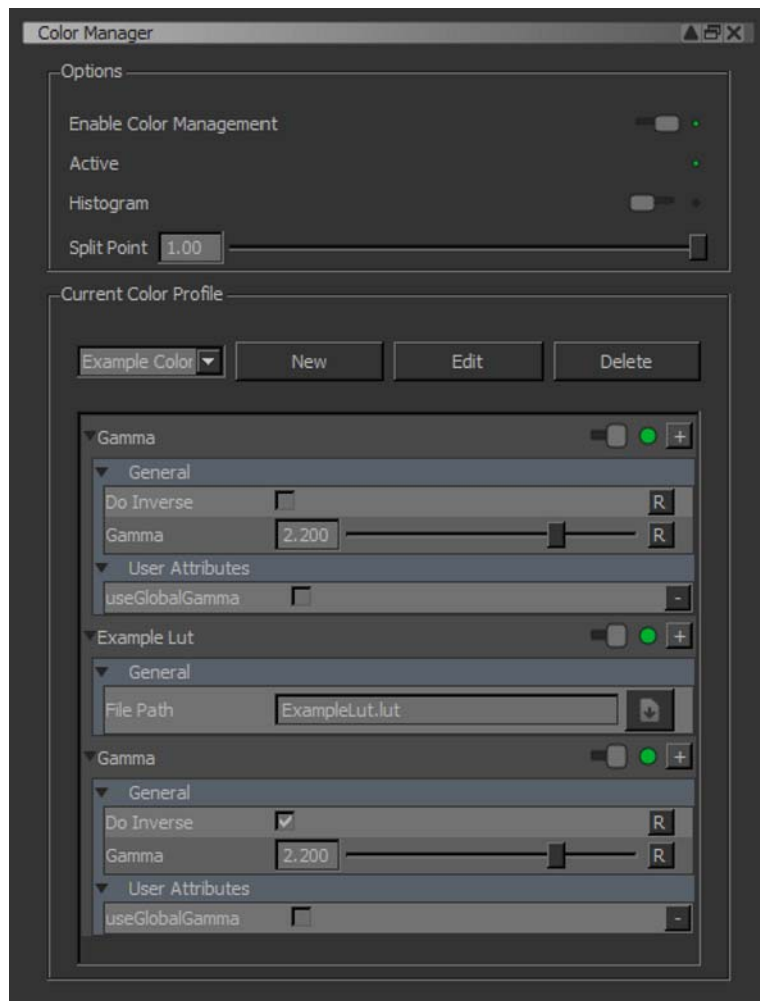
Set How Colors Are Displayed Onscreen

Mari sets how colors are displayed for each project separately. You can set this using either the **Color Manager** or the **ColorSpace** toolbar.

Set how colors are displayed using the Color Manager

1. Open the **Color Manager** palette:
 - right-click on the toolbar, and select **Color Manager** from the drop-down menu, or
 - from the **View** menu, select **Palettes > Color Manager**.

*The **Color Manager** palette displays color-display settings for the current project.*

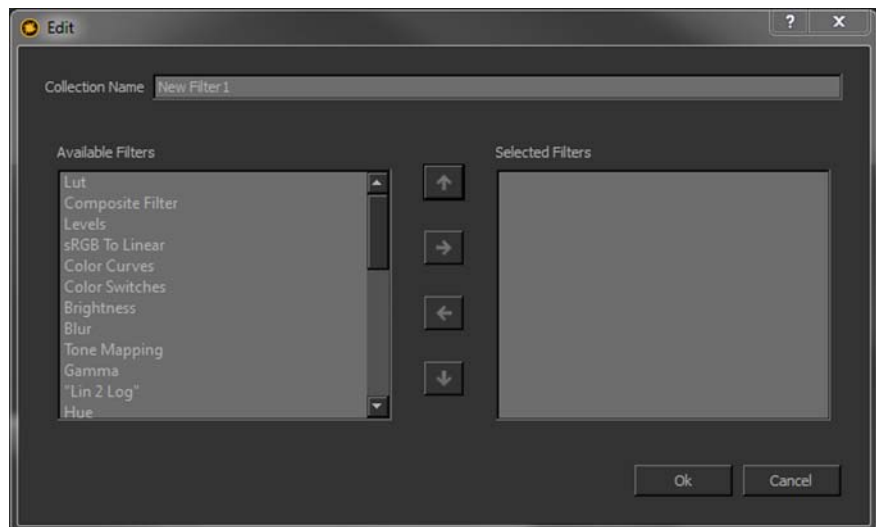


2. Under **Options**, select the main display options:

- **Enable Color Management**—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.

Tip *Color correction only applies to color channels. Scalar channels never use corrected colors. Even if color correction is on, scalar channels display the actual color values painted on the surface.*

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.*

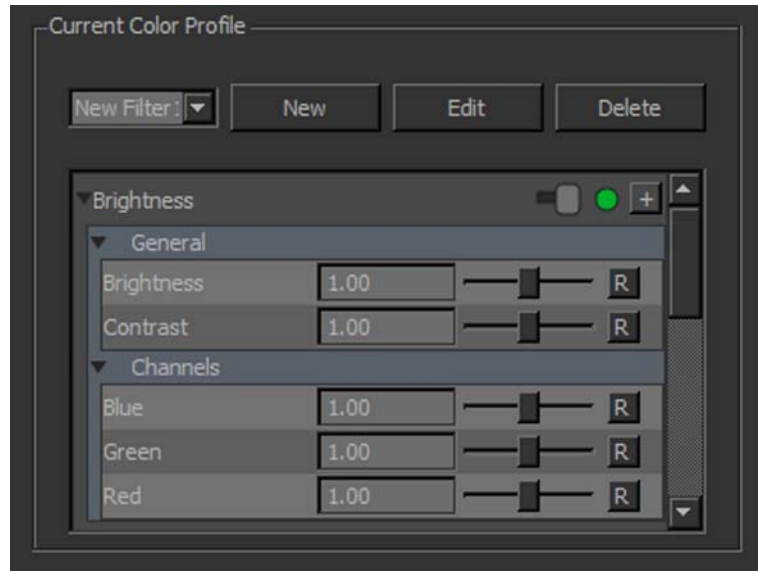


Note *You can delete color profiles from the list by selecting a profile and clicking **Delete**. You are restricted from modifying the **ColorSpace** profile however.*

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.



Tip For more details on Mari's filters, see [Filtering Images and Paint](#).

Set how colors are displayed using the ColorSpace toolbar:

1. Click  or select **Enable Color Management** in the **Color Manager** palette.
2. Select a custom OpenColorIO (.ocio) configuration file by clicking  or use Mari's default configuration.
For more information on OpenColorIO, including the OCIO file format, please visit <http://opencolorio.org/>.
3. Select the colorspace of the current scene using **Input ColorSpace** dropdown.
4. Set the display device used to view the scene using the **Display Device** dropdown menu.
5. Select the colorspace transform to apply to the scene from the **View Transform** dropdown menu.
6. Use the **Component** dropdown menu to see the individual color components in the scene.

7. Use the **Gain** controls to set the amount of exposure adjustment applied before the display transform by either entering a multiplier (exposure value), dragging on the slider, or adjusting the F-Stop value.
8. Finally, set the amount of **Gamma** correction applied after the display transform by entering a gamma level or by dragging the gamma slider.

Note *The custom OpenColorIO configuration file can also be changed in the **Mari Preferences** using environment variables. If there is no environment variable set up, Mari uses the path to the OCIO configuration for the project, or if this is incorrect, falls back to the default Mari is equipped with.*

If an environment variable is set up, it's important to note that Mari uses this file regardless of project settings. You cannot change the configuration path and we discourage attempting to change the setting. If the environment variable is incorrect, you'll be warned on startup that color management is disabled.

Sandbox *Try changing the color profile on a project. First select a couple of the other predefined profiles, then create one of your own. Tweak the properties on the filters you added and notice what happens. Then, load a custom color configuration file via the **ColorSpace** toolbar, tweak the properties again and notice how your changes affect the colors onscreen.*



20 CHANNELS

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 [Multiple Objects](#) for details on working with multiple objects.

Different Ways to Create Channels

When you create a channel, you set the color depth and patch size. You can change both of these. You can also resize individual patches within the channel. Channels are either color or scalar. Channels can be created:

- using the settings for existing or recently created channel 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 merging two channels
- by baking the paint buffer to a new channel
- by baking a shader (see [Shaders](#) for details)
- by copying a shader (see [Shaders](#) for details)
- from the alpha value or mask from another channel
- by taking snapshots of existing channels.

Presets and Templates

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 choose to 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. See the [Extending Mari](#) chapter for details.*

Merging and Blend Modes

When you merge two channels to create a new channel, you can optionally use a third channel as a mask, to control which sections of the source channel to use for the merge. The alpha value for the merged source is also fully controllable. Mari has several blending modes when performing a merge — a standard overlay merge, burn, dodge, hardlight, subtract, add, and so on.

You can also import or export the contents of a channel. See the *Getting Started Guide* for details.

Channels can be an important factor in building shaders (which set how Mari displays the model on the canvas). See [Shaders](#) for details.

Alpha and Masks

To make creating masks easier, Mari has two functions to easily convert between masks and alpha values:

- **Mask to Alpha** — creates a new channel using an existing mask channel (painted in black and white), deriving alpha values from the mask (where the alpha on a pixel is controlled by the shade on the original mask), and then combining those alpha values with the diffuse color from another channel.



Initial B/W mask channel

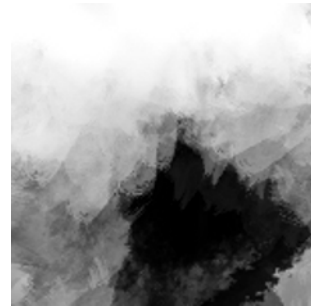
Diffuse color channel

*Result of **Mask to Alpha***

- **Alpha to Mask** — does the opposite. It converts a channel with alpha values to a black and white mask, where the shade on each pixel is controlled by the alpha on the original channel.



Initial channel with alpha values



Result of Alpha to Mask

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 [Create and Delete Individual Channels](#).

Snapshots

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.

Steps for Managing Channels

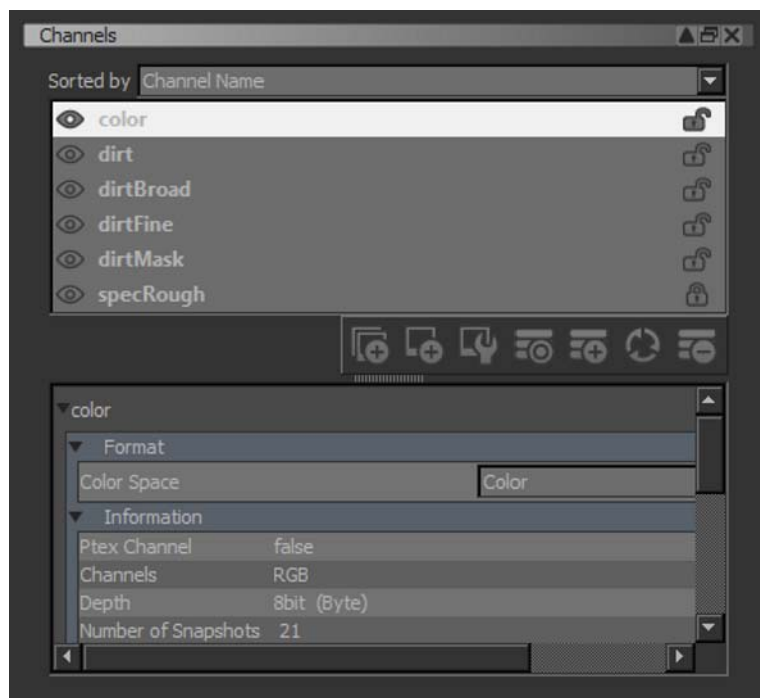
To manage the channels in your Mari project, follow these steps:

1. [Create and Delete Individual Channels](#)
2. [Create Multiple Channels from Presets](#)
3. [Navigate Between Channels](#)
4. [Set and Change Channel Properties](#)
5. [Take and Manage Channel Snapshots](#)
6. [Lock and Unlock Channels](#)
7. [Merge Channels](#)
8. [Convert Masks to and from Alpha](#)

Create and Delete Individual Channels

1. Start Mari and open your project.
2. Open the **Channels** palette:
 - right-click on the toolbar, and select **Channels** from the dropdown menu, or
 - from the **View** menu, select **Palettes > Channels**.

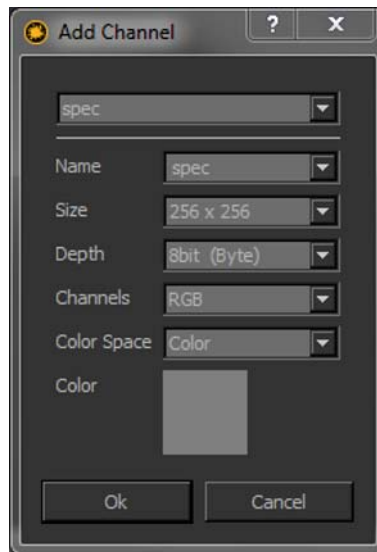
*The **Channels** palette displays, showing the channels in the current project.*



3. To add a single channel:

- right-click in the **Channels** palette, and select **Add Channel**,
- click the  button, or
- from the **Channels** menu, select **Add Channel**.

The *Add Channel* dialog box displays.



4. 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**, **Channels**, and **Color Space** populate automatically.

Tip *If you have never created a project or channel, the dropdown is empty. In this case, give the channel the name "Diffuse" and leave all the other values at their defaults.*

5. 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).
 - **Channels** — the number of color channels on the channel (one, two, RGB, or RGBA).
 - **ColorSpace** — color or scalar.

Tip *Scalar channels store absolute color values (for example, for painting masks). When you export a scalar channel, Mari does not apply the LUT — colors export precisely as painted.*

Note *Channels used for bump maps, normal maps, or flow maps should use a **Scalar ColorSpace** to avoid visualization of the maps changing.*

6. If you intend on cloning from, or painting through, either 16-bit or 32-bit images, the **Depth** needs to be set to either 16-bit or 32-bit, 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.

7. If you want to, select the initial **Color** for the channel.

Tip *We recommend that you create:*

- *mask channels with a black background — so you can paint the mask data white*
- *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).*

8. Click **Ok** to add the channel to the object. If you have more than one object in the project, you can choose to add the channel to the current object or all objects in the project.

*Mari creates your new channel, and displays it in the **Channels** palette. When first created, channels are a flat color (as selected in the **Add Channel** dialog).*

Tip *You can paint the channel a different flat color if you want: from the **Channels** menu, select **Fill Foreground** (to fill the channel with the foreground color) or **Fill Background** (to fill with the background color).*

9. To create a copy of a channel, either:

- right-click on the channel in the **Channels** palette, and select **Duplicate Channel**, or
- from the **Channels** menu, select **Duplicate Channel**.

Mari creates your new channel as a copy of the original — including all the painting on the model.

10. To delete a channel:

- right-click on the channel in the **Channels** palette, and select **Remove Channel**,

- with the channel selected, click , or
- from the **Channels** menu, select **Remove Channel**.

Mari asks for confirmation. 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.

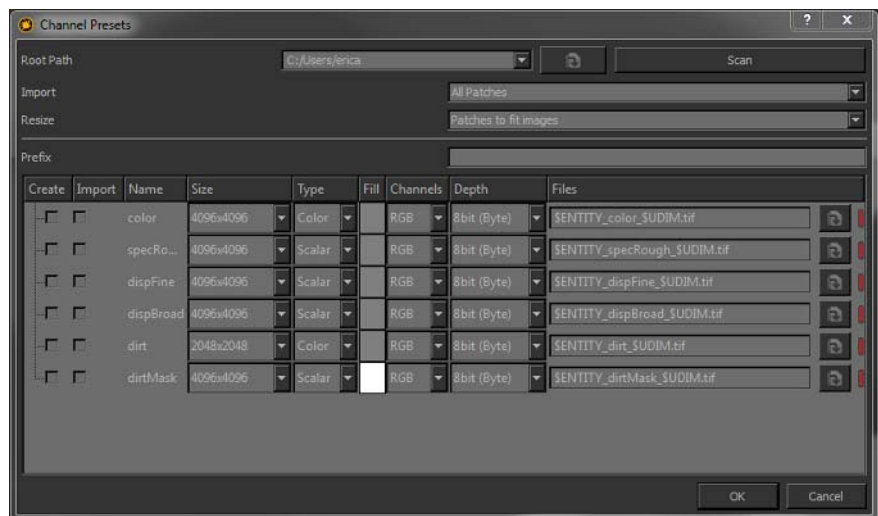
Sandbox *With a sample project loaded, try this:*

- *Create a channel called "TEST", with features: RGBA, bit, and color.*
- *Then change its color.*
- *Duplicate it.*
- *Then delete both new channels.*

Create Multiple Channels from Presets

- To create multiple channels from presets:
 - right-click in the **Channels** palette, and select **Channel Presets**,
 - click on the  button, or
 - from the **Channels** menu, select **Channel Presets**.

*The **Channel Presets** dialog opens.*



- 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. Select a **Category**. The categories control the groups of channels that appear in the list.

*For example, if you select the **Vehicle** category, the list changes to show all the channels that apply when painting vehicles.*
6. Choose 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 choose to **Create All**, **Create None**, **Import Existing**, or **Import None**.
7. You can change any selected channel's **Size**, **Type**, or **Fill**.
8. 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 open an **Open** dialog box and find the texture files.
9. Once you are happy with the channels and textures selected, click **OK**.

Mari creates the channel (and imports selected textures).

Tip *Depending on the number of channels and the size of the textures, the import could take a long time. Consider going for a cup of tea!*

Navigate Between Channels

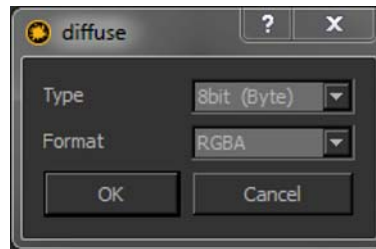
1. To view a particular channel, click on the channel in the **Channels** palette.
2. To move to the next or previous channels, from the **Channels** menu, select **Next Channel** or **Previous Channel**.

Tip *Mari has the following keyboard shortcuts to navigate between channels:*

- Press **Page Up** to move to the next channel and **Page Down** to move to the previous.
- Press **T** to toggle between the last two channels viewed.

Set and Change 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 **Format** for the channel, and click **OK**.
3. To switch a channel's type between color and scalar, select the channel in the **Channels** palette.
The bottom half of the palette displays the details of the channel.
4. Change the **ColorSpace** to **Color** or **Scalar**.

Take and Manage Channel Snapshots

You can take snapshots of the current channel or all channels in a project.

Snapshot Current Channel

1. To take a snapshot of the current channel, either:
 - select the channel in the **Channels** palette and click ,
 - select **Channels > Snapshots > Snapshot Current Channel**, or
 - from the **Snapshots** palette, with **Channels** set to **Current**, click



*The **Snapshot Channel** dialog box displays.*

2. Enter a name for the snapshot and click **OK**.

Snapshot all Channels

1. To take a snapshot of all channels in the project, either:

- click  in the **Channels** palette,
- select **Channels > Snapshots > Snapshot All Channels**, or
- from the **Snapshots** palette, with **Channels** set to **All**, click



*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.*

Use the Snapshots Palette

Once taken, snapshots are added to the **Snapshots** palette. To open the **Snapshots** palette:

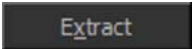
- Right-click on the toolbar, and select **Snapshots** from the menu,
- from the **View** menu, select **Palettes > Snapshots**,
- from the **Channels** menus, select **Snapshots > Manage Snapshots**, or
- click  in the **Channels** palette.



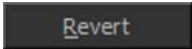
*The **Snapshots** palette displays, showing the existing snapshots for the current channel and for the entire project. Information about the snapshot such as when it was taken and by which user, along with a thumbnail preview, is also displayed.*

There are a number of options available for managing existing snapshots. From the **Snapshots** palette you can **Extract**, **Revert** or **Delete** snapshots.

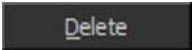
To extract a snapshot into a new channel:

1. Select a snapshot and click .
*The **Extract Snapshot** dialog displays.*
2. Enter a name for the new channel and click **OK**.
Mari creates a new channel with the given name.

To load an existing snapshot:

Select the snapshot and click .
Mari loads the selected snapshot into the current channel.

To delete a snapshot:

1. Select a snapshot and click .
*The **Delete Snapshot** dialog displays, asking you to confirm.*
2. Click **OK**.
*The snapshot disappears from the **Snapshots** palette.*

Autosnapshots

When saving a project, Mari automatically takes a snapshot of all available channels and adds it to the **Snapshots** palette. This feature is enabled by default and 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**.

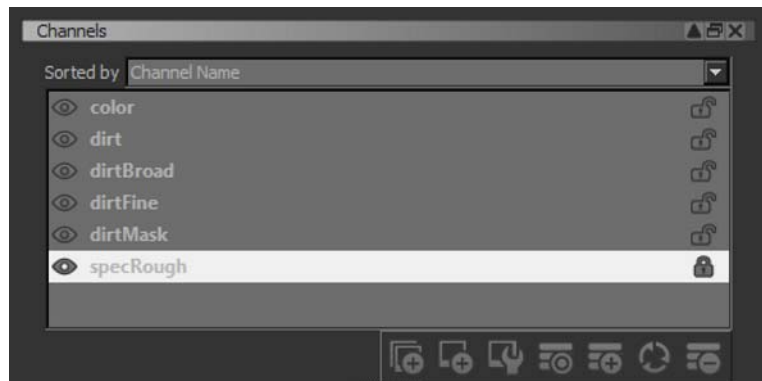
3. When **Enabled with Limit** is selected, Mari limits the number of autosnapshots that are stored in the **Snapshots** palette to the number specified by the **Autosnapshot Limit**. If the number of autosnapshots stored exceeds this limit, the oldest autosnapshot is removed from the palette.

Note *The **Autosnapshot Limit** does not affect the number of regular channel snapshots that can be stored in the **Snapshots** palette.*

Lock and Unlock Channels

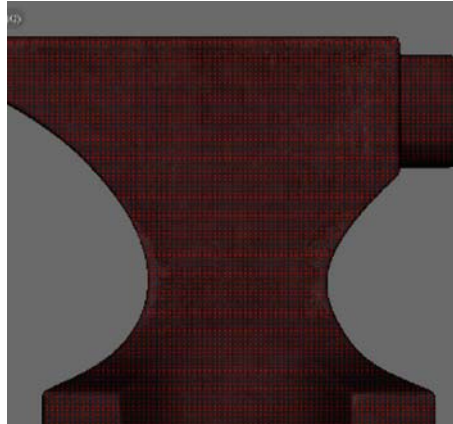
1. To lock the current channel, from the **Channels** menu, select **Lock > Lock Channel**.

The current channel is locked.



*The channel's listing in the **Channels** palette changes to show the "closed lock" icon .*

In addition, the model appears with an overlay of small squares to make it obvious that the channel is locked and cannot be edited.



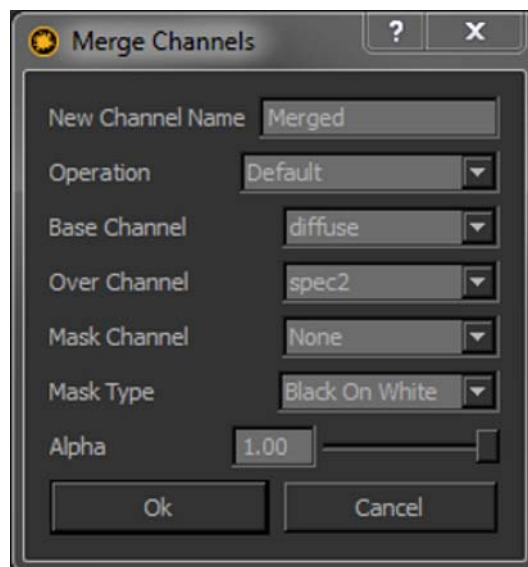
2. To unlock the current channel, from the **Channels** menu, select **Lock > Unlock channel**.

*Mari unlocks the current channel, changes its listing in the **Channels** palette to the "open lock" icon, and removes the overlay from the display.*

3. You can also lock and unlock channels directly in the **Channels** palette:
 - right-click on the channel, and select **Lock Channel** or **Unlock Channel** from the dropdown menu, or
 - click on the lock icon to the right of the channel.

Merge Channels

1. From the **Channels** menu, select **Merge Channels**.
*The **Merge Channels** dialog box displays:*



2. Set the following properties for the merge:
 - **New Channel Name** — Enter a name for the new channel to create based on the merge.
 - **Operation** — Select a type of merge from the list (see below).
 - **Base Channel** — Select a channel to use as the basis for the new channel (with the **Over Channel** placed on top).
 - **Over Channel** — Select a channel to merge on top of the Base Channel.

The channel merge **Operations** available are the same as the paint blending modes. For examples of each mode, see [Paint Blending Modes](#).
3. If you want to merge using a mask, set the **Mask Channel** and **Mask Type**.

Mask channels control which areas of the over channel to use in the merge. In merging, Mari examines the mask channel. Where Mari finds a specific pattern in the mask channel, it uses the corresponding area in the over channel for the merge. Mari ignores the rest of the over channel during the merge, using only those areas the mask data selects.

You can select to use either the black pixels or the white ones as the source of the mask data.

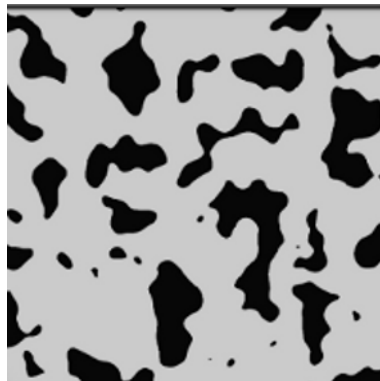
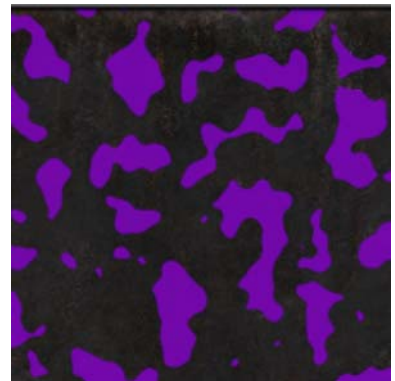
Here's an example of a default merge of two channels, using a mask with Black on White and with an alpha of 80%:



Base Channel



Over Channel

*Mask Channel**Merged Channel*

Tip *Ensure that your mask channel is fully painted. Mari always considers unpainted areas as part of the mask, no matter which **Mask Type** you're using. (It's a good idea to fill the channel before you start painting a mask channel)*

4. Set the **Alpha**, and click **OK**.

Mari applies that alpha value to the over channel when placing it on top of the base channel. (An alpha of 1 is solid color, 0.01 is extremely translucent.)

Tip *When using an over merge, using an alpha of 0 gives you the contents of the **Base Channel**; 1 just gives you the contents of the **Over Channel**.*

Sandbox *Review the different available [Paint Blending Modes](#), and try channel merges with three or four of them in your test project. How do the different channel merges affect each other?*

Convert Masks to and from Alpha

You can:

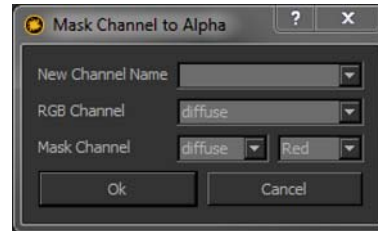
- [Convert a Mask to Alpha](#)
- [Convert Alpha Values to a Mask](#).

Convert a Mask to Alpha

You need two channels at the **same resolution**:

- one to provide the diffuse surface
 - one to act as the mask, determining the final alpha values.
1. Either:
- right-click on the **Channels** palette and select **Mask to Alpha**, or
 - from the **Channels** menu, select **Mask to Alpha**.

The **Mask Channel to Alpha** dialog box displays.



2. Enter a name for the new channel to create.
3. Select the color input channel from the **RGB Channel** dropdown.
Mari takes the color for the final channel from this input.
4. Select the mask input from the **Mask Channel** input, and the color channel (red, green, or blue) to use as the mask.
This lets you paint mask channels in color, and use the different color channels to get different masks from a single channel.
5. Click **OK**.
Mari creates your new channel. At each pixel, the alpha value comes from the color value of the selected color channel in the mask texture. For example, a pixel with RGB values of (0.4, 0.5, 0.2) gets an alpha value of 0.4 if you select the red channel, 0.5 if you select green, and so on).

Convert Alpha Values to a Mask

1. Select a channel with **RGBA** color channels (not just RGB).
2. Either:
 - right-click on the channel in the **Channels** palette and select **Mask to Alpha**, or
 - with the channel active, from the **Channels** menu, select **Mask to Alpha**.*Mari creates a new channel, named after the old channel with the word "Mask" at the end (for example, **layer1** becomes **layer1Mask**). The new channel is black and white, with the color at each pixel taken from the alpha value of the corresponding pixel in the original channel (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 black. If necessary, use the **Paint** tool in **Clear** mode to erase some of the paint off your channel.*

21 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.

Ptex Support in Mari

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
- baking a custom shader into a Ptex channel
- 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, shader modules 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 Modelling Requirements in Mari

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 modelling 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.

Minimum Face Size

Within **Preferences**, it is possible to set the **Minimum Size** for face textures via **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.

Paint Preservation

Geometry versions from **.obj** files can be added to a Ptex project, provided all the versions are topologically equivalent. This preserves the paint that has already been applied to the existing version.

Steps for Using Ptex

To use Ptex in Mari, follow these steps:

1. [Create a New Project](#)
2. [Set the Ptex face Texture Options](#)
3. [Manage Ptex Faces](#)
4. [Export a Ptex Channel](#)
5. [Import a Ptex Channel](#)
6. [Paint Preservation with Multiple Ptex Geometries](#)
7. [Shader Baking to a Ptex Channel](#)

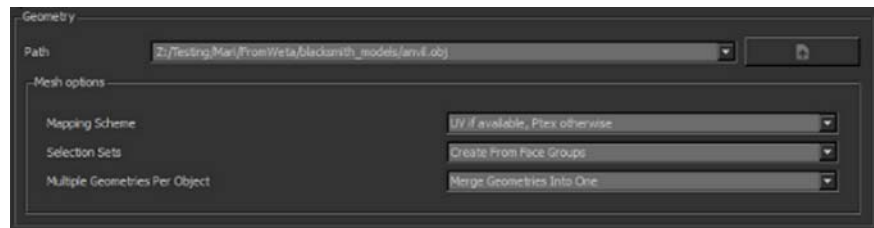
Create a New Project

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.

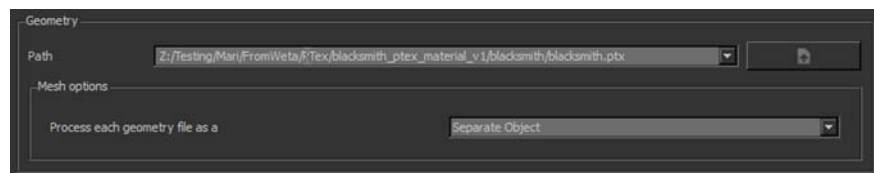
To create a new project, using either one file or a set of files, follow the steps below:

1. Start Mari and select **File > New**.
*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 Sets**, and working with **Multiple Geometries Per Object**.
 - **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 Sets:** If you want to select a specific object group in the geometry while you work, this option allows you to set up selection sets 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 choose to **Merge Geometries Into One**, Mari combines the individual geometries into a single geometry, while choosing **Create Separate Geometries (Ptex Only)** creates individual geometries.



New Project dialog: .obj mesh options



New Project dialog: .ptx mesh options

- If you choose to load a .ptx file, the **Mesh Options** only provide the option for selecting **Separate Object** or **Separate geometry in one Object** for the **Process each geometry file** 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.

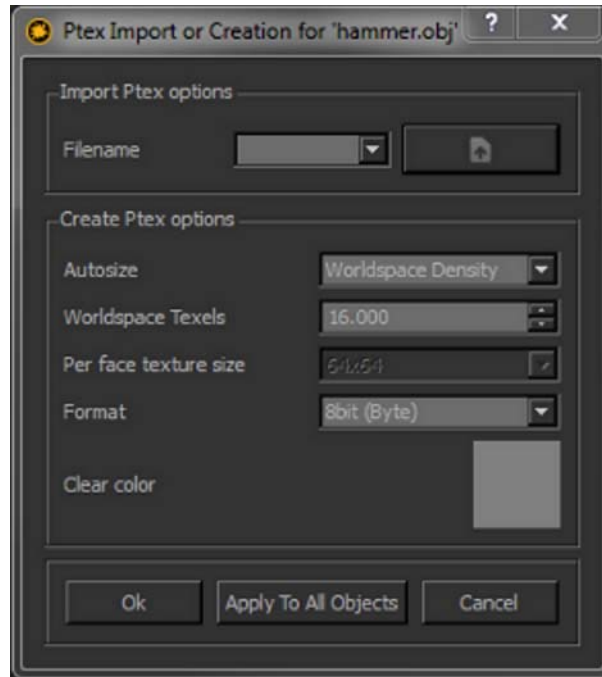
Note *Even if you choose to **Create Separate Geometries (Ptex Only)**, Mari only displays a single object on the canvas. However, when you export a channel you have the option to export individual .ptx files for each geometry.*

*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.*

- Click **OK**.

Set 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.

Import 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.

Create 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**.

Manage 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
- - 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.*

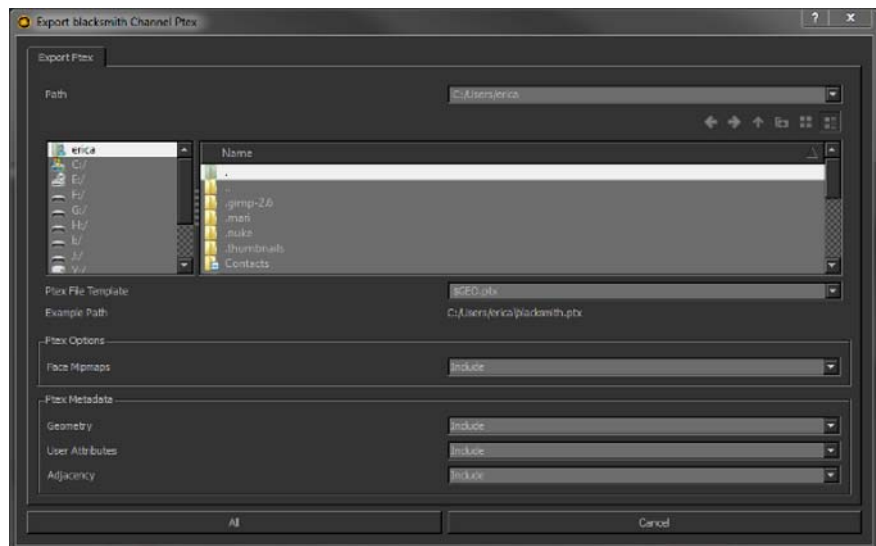
You may notice a slight pause after adjusting the resolution of a selected face. This happens while Mari regenerates the texture's proxy on the CPU.

Tip *These options are also available in the menu bar under **Ptex**.*

Export a Ptex Channel

- To export a **.ptx** channel, select:
 - **Channels > Export**, or
 - Right-click on a channel and select **Export**.

*The **Export Channel** dialog displays*



- Enter the Ptex export path in the **Path** field or manually browse to the location.
- Set the **Ptex File Template** for Mari to use when generating the Ptex filename. You can edit this line manually. As you edit, you'll see the **Example Path** update 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.

Sandbox *Experiment with the different **Ptex File Template** variables, and see how the **Example Path** changes to show how Mari generates exported filenames.*

4. In the **Ptex Options** panel set whether Mari includes or excludes **Face Mipmaps**.
5. In the **Ptex Metadata** panel set whether Mari includes or excludes the following information with the file when exporting:
 - **Geometry**
 - **User Attributes**
 - **Adjacency**
6. Click **All**.

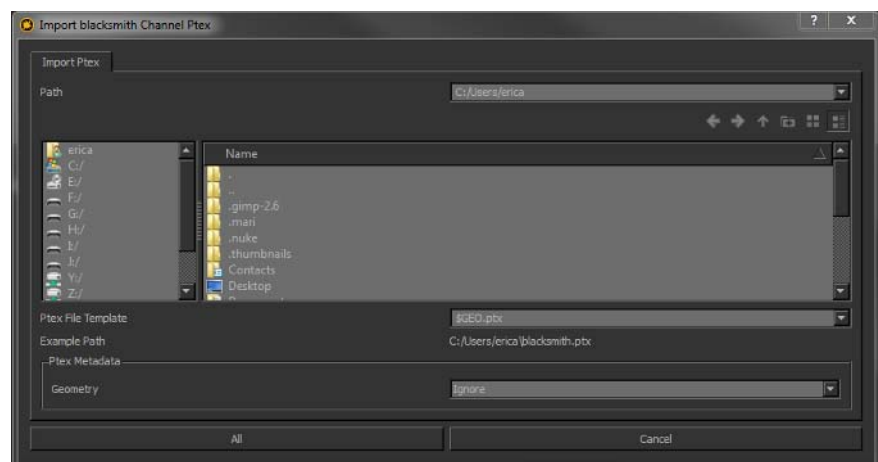
*Mari exports your textures to disk in **.ptx** format.*

Import a Ptex Channel

To import a **.ptx** file to a channel, follow the steps below.

1. Select:
 - **Channel > Import**, or
 - Right-click on a channel and select **Import**.

*The **Import Channel** dialog displays.*



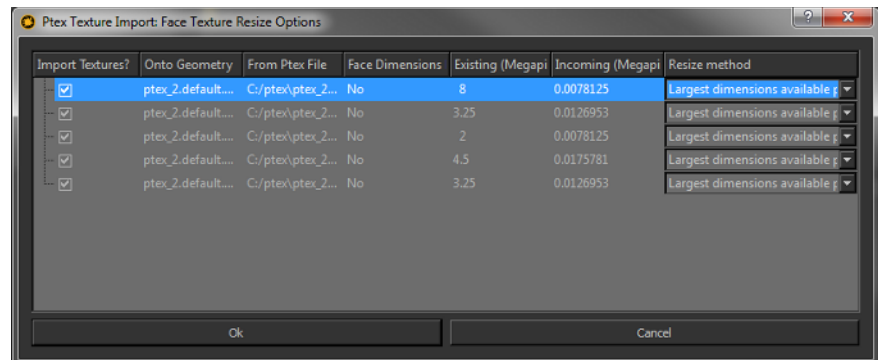
2. Enter the Ptex import path in the **Path** field or manually browse to the location.
3. You can manually select the Ptex files to import or you can set the **Ptex File Template** to automatically search for Ptex files that match the object, channel or geometry variables. You can edit this line manually. As you edit, you'll see the **Example Path** update 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.
4. In the **Ptex Metadata** field you can set whether Mari should:
 - **Ignore** the geometry metadata, or
 - **Import as New Version**.
 5. Click **All**.
Mari imports the images and loads them into the selected channel.

To import a texture to a channel from multiple **.ptx** files, where multiple geometries are in an object, follow the steps below.

1. Select:
 - **Channel > Import**, or
 - Right-click on a channel and select **Import**.

*The **Import Channel** 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** environment 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.*

Note *The **Ptex Texture Import: Face Texture Resize Options** dialog is also displayed when importing new geometry versions from **.ptx** files.*

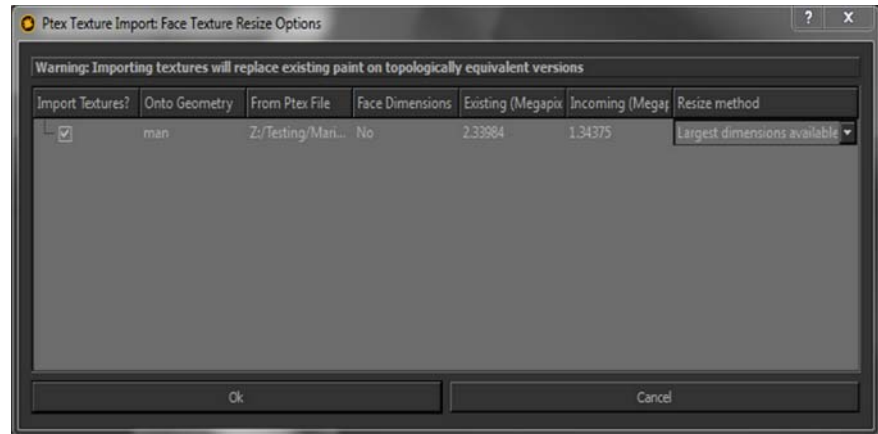
Paint Preservation with Multiple Ptex Geometries

Though **.obj** files can be added to Ptex projects with topologically equivalent versions, 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 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.



Shader Baking to a Ptex Channel

You can change the bleed settings for shader baking in Ptex. For triangular-only meshes, a shader bake without bleed generally results in visible artifacts around faces in the channel. Experimenting with the bleed settings can work to counteract the visibility of these artifacts. There are fewer artifacts for quad meshes and the results can vary depending on the bleed settings.

You are encouraged to change the bleed settings for the shader baking in order to achieve the results you want for your project.

Refer to [Bake a Shader Component onto a Channel](#) in the Shader chapter for more information on applying the bleed option.



22 SHADERS

Shaders control how Mari displays the model on screen. Mari uses shaders to build the surface of the model and how it reacts to light. Mari *always* uses a shader to generate the surface displayed on screen.

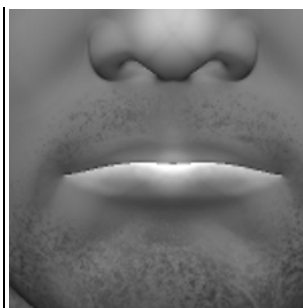
The default shader for a new project builds the surface by rendering the current channel as diffuse color. You can create custom shaders, which can use data from multiple channels or other inputs (colors or images) to render the surface of the model in Mari.



Color channel, plus...



*displacement channel,
plus...*



*specular channel,
equals...*



...this. Here the composite 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.*

*If you want to blend channels together, use the **Merge Channels** function. See [Channels](#) for details.*

Shaders include:

- basic lighting details, including ambient, diffuse, and specular light intensities
- a series of modules, each describing one operation the shader performs when generating the shaded surface.

Mari generates the shaded surface by going through the modules in the current shader, in the order specified. Each module takes an input from one or more sources (usually, but not always, channels) and adds it to the current surface. The final shaded surface as you see it on screen is the final result of this chain of modules.

For example, you could set up a shader with the following chain of effects:

1. Create the initial shaded surface from the diffuse channel.
2. Add content from another channel to the shaded surface, using a third channel as a mask.
3. Display displacement preview information, using a channel as the displacement source, with a user-configurable displacement magnitude.

As with channels, you set up shaders individually for [Multiple Objects](#) in the project.

Tip *As with channels, lights, patches, and projectors, you can organize shaders into sets.*

Available Modules

The types of shader modules available in Mari include:

- applying and previewing displacement (using displacement preview and bump mapping)
- adding channels
- displaying diffuse channels, optionally using masks to control which areas display, or multiplying values in one channel by another channel
- adjusting specular display, using channels as the inputs

- environment mapping, such as spherical or cube maps
- projecting cubic or spherical image textures using projection and masked projection shaders.

This tutorial includes a full list of Mari [Shader Module Types and Examples](#).

Tip *When working on masked textures, we suggest using at least three shaders:*

- *The default shader, which shows the contents of the current channel. This lets you see exactly how the mask looks.*
- *A mask shader, which shows the effects of the current mask on the diffuse channel. This lets you see how the mask works.*
- *A composite shader, which shows a preview of the final model with all shader modules applied (all masked textures, specular, displacement, and so on).*

You can then assign shortcut keys to these shaders. This lets you quickly switch between painting the mask directly in the channel, painting while watching the effect on the masked texture, and occasionally loading the composite shader to check how the mask works with the overall textures in the project.

Shader Limits

Most modules do not require texture slots. The following modules are examples of modules that use a varying number of them:

- **Diffuse Color Curves** and **Masked Diffuse Color Curves**,
- **Tiled** and **Masked Tiled**, and
- **Triplanar** and **Masked Triplanar**.

To prevent you from using too many texture slots, the **Add New Shader Module** dialog disables any of the modules that would cause the texture slot limit to be exceeded if applied. These appear grayed-out and list a message in the description of the shader detailing how many texture slots are required, as well as how many are available.

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 are used. You can turn off these features to free up more texture slots. See [About Masking](#) and [Object Shadows](#) for more information.*

Shading and Channels

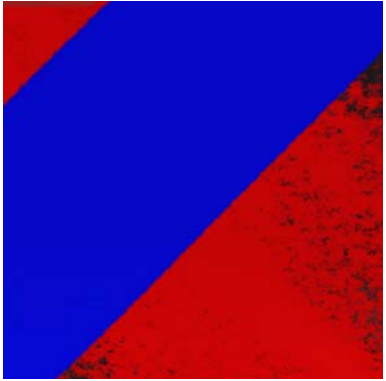
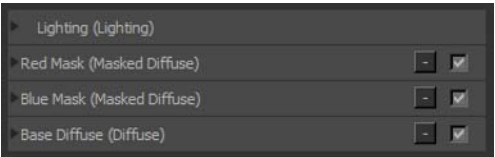
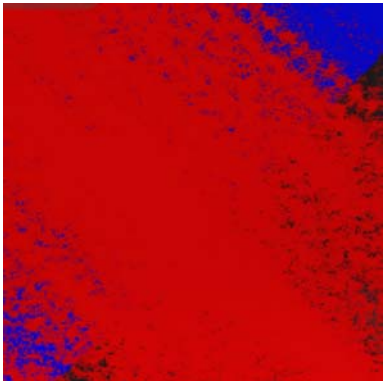
Using sophisticated shaders can make it hard to remember which channel you're painting on. Since Mari has a default shader which just shows the contents of the current channel, you can always switch back to that (to check what you're doing!).

Tip *A useful way to keep track of which channel you're editing is to add it to the Heads-Up Display on the Mari canvas. From the **Edit** menu, select **HUD Manager**. Under **Project**, put a check in the box next to **Current Channel**. Mari now shows the name of the channel you are currently painting as part of the background information on the canvas. For more details on configuring the HUD, see [Configuring the User Interface](#).*

Ordering Modules

Mari applies the shader modules 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.

For example, the following table illustrates a basic shader with two Masked Diffuse modules: one to show red stripe, the other to show a blue one. The order of the two Masked Diffuse modules determines which texture appears on top. The table shows the results of swapping the order of the shader modules.

Shader order	Result
	 <i>The diffuse texture (bricks) is on the bottom, then the grass, then the dirt. Finally, Mari applies the lighting.</i>
	 <i>The diffuse texture is on the bottom, but this time the grass is on top of the dirt.</i>

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.

Cube Mapping

Cube maps provide a context for reflecting images of surrounding environments on the model, similar to the **Environment Sphere** shader module. The difference is that the environment is modelled 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 **Environment Cube** shader module. As with the environment sphere, the cube allows you to modify the blend amount, and the falloff starting and ending points to create the exact level of reflection you want on your model.

Projection Shaders

Cubic and spherical projection shaders allow you to import a texture into the scene and project it onto the geometry. Using multiple **Projection Cube** or **Projection Sphere** shaders 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 four projection shaders:

- Projection Cube
- Projection Sphere
- Masked Projection Cube
- Masked Projection Sphere

Masked projection allows you 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 shader 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.*

Baking Shaders onto Channels

Large shaders with multiple shader input models have a processing overhead. Each shading module takes a bit more time to process, and as your shaders grow they can eventually slow Mari down.

To prevent this, you can bake a shader onto a channel. That is, take the

shaded surface produced by a shader, and either create a new channel with it or overwrite an existing channel. This gives you a way of getting the shader functionality of blending together your channels, without slowing Mari down to constantly re-render the channels you've already finished painting. You can think of this as being like flattening the layers of an image in Photoshop® down to a single layer.

You can also bleed shader baking. For triangular-only meshes, a shader bake without bleed generally results in visible artifacts around faces in the channel. This does not happen as frequently with quad meshes, but you are still encouraged to experiment with the bleed settings to achieve the results you want.

Creating Layered Shaders

If the channels in your project were imported from a layered **.psd** file, there is a really quick way to create a layered shader consisting of a module for your base layer, and a number of diffuse blend modules for all the subsequent layers in the original **.psd** file. Every module in the new shader uses the corresponding **.psd** layer as an input texture.

Vector Brushing

Three vector brush shader modules exist for viewing vector maps. These shader modules – **Diffuse Tangent to Screen**, **Diffuse Tangent to World**, and **Vector Data** – 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 shader modules together with the **Vector Painting Brush**, refer to the *Vector Brush* chapter in the *Mari User Guide*.

The **Flow** and **Masked Flow** shader modules allow 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.

A Python example script for setting up default vector shaders can be found under **Python > Examples > Setup Vector Brush** or as an icon  on the **Vector Painting** toolbar. This script automatically creates two new **Default Vector** shaders with the **Vector Data** shader module already set up. For more information on using the **Default Vector** shaders, refer to [Adding Default Vector Shaders](#) on page 131.

Steps for Using Shaders

To manage the shaders in your Mari project, follow these steps:

1. [Open the Shaders Palette](#)
2. [Create a Custom Shader](#)
3. [Set the Shader Details](#)
4. [Bake a Shader Component onto a Channel](#)
5. [Create a Layered Shader](#)
6. [Copy an Existing Shader to Another Object](#)
7. [Save, load, and delete shaders](#)

This chapter also includes a reference table of [Shader Module Types and Examples](#).

Open the Shaders Palette

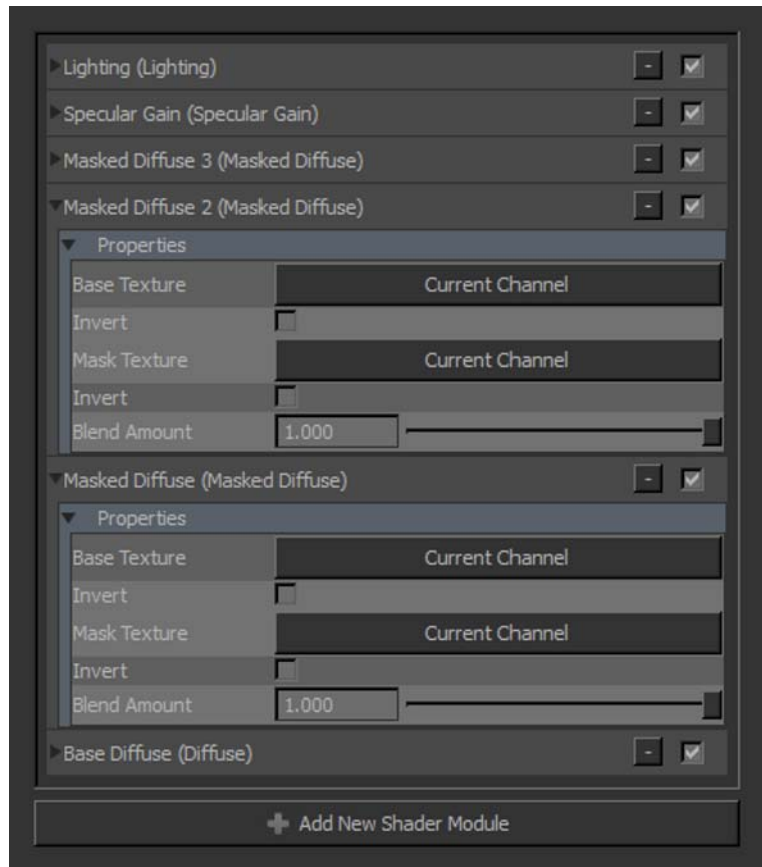
1. Start Mari and open your project.
2. Open the **Shaders** palette:
 - right-click on the toolbar, and select Shaders from the dropdown menu, or
 - from the **View** menu, select **Palettes > Shaders**.

*The **Shaders** palette displays, showing the shaders defined for your current project.*



3. Click a shader in the list to see its details.

When you select a shader, the bottom of the palette shows the details.



Create a Custom Shader

1. To add a new shader, either:
 - right-click an empty space in the **Shaders** palette, and select **Add Shader**,
 - from the **Shading** menu, select **Add Shader**, or
 - in the **Shaders** palette, click .

The new shader appears in the list.

Tip *You can also duplicate an existing shader by selecting **Duplicate Shader** from the **Shading** menu or the right-click menu in the **Shaders** palette.*

2. Give the shader a more meaningful name by double-clicking it in the list and typing the new name.

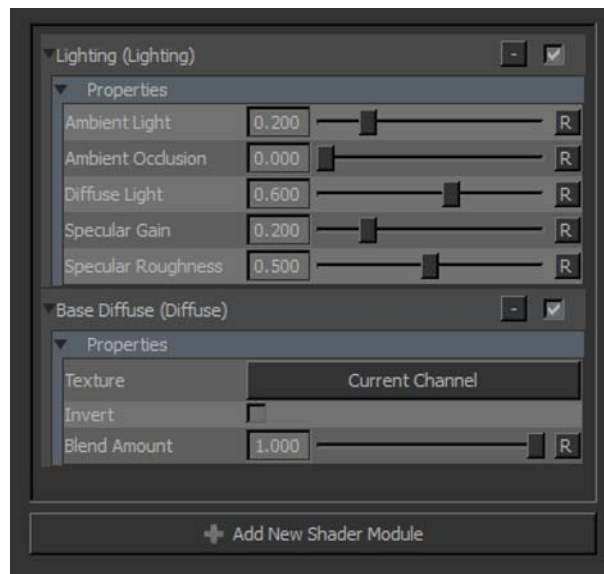
Tip You can set a keyboard shortcut for the shader by right-clicking it and selecting **Assign Shortcut** from the dropdown menu. See [Using Shortcuts](#) for details.

Tip You can right-click any of the shaders, or within the shader dialog, to access the dropdown menu for more shader channel options. Right-clicking an existing shader also selects it.

Set the Shader Details

1. Select the shader you want to customize from the list at the top of the **Shaders** palette.

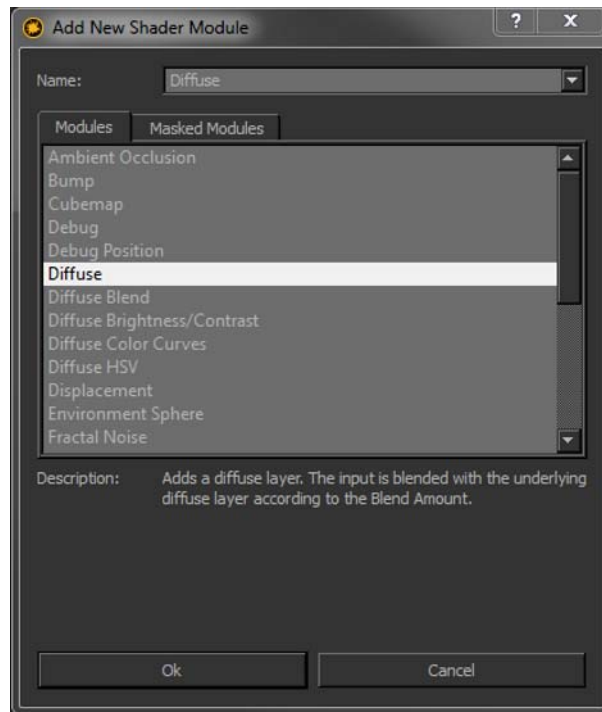
The bottom of the palette displays that shader's details.



Tip You can edit only the lighting details on the **Default Shader** (the one shader all Mari projects start with).

2. You can build up your shader by adding, removing, and moving modules. To add a module, click the **Add New Shader Module** button.

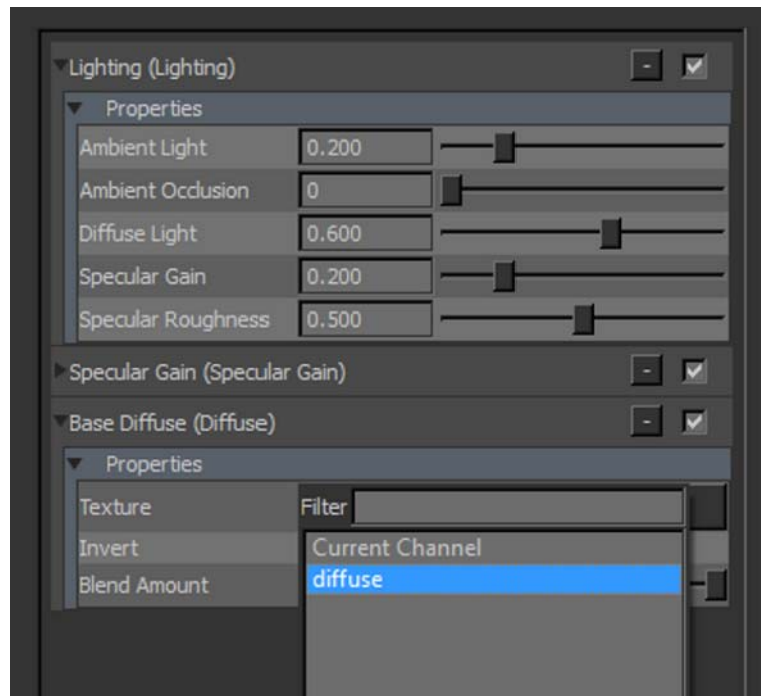
*The **Add New Shader Module** dialog box displays.*



3. Enter a **Name** for the module (recent names are available in the dropdown list), select a [module tab](#) and a [module type](#), and click **OK**.
Mari adds the module to your shader.

Tip *When you select a type of **Module**, a brief **Description** displays below — but for full descriptions and examples, see [Shader Module Types and Examples](#) at the end of this chapter.*

4. Set the details for the new module in the bottom part of the **Shaders** palette (these depend on the type of module you've selected). Some require inputs — for example, a masked diffuse module would require an input for the channel containing the mask.
5. Select the input by clicking on the button to display a menu with a list of all the available channels in the project, and double-click on the channel you want. For projects with lots of channels, you can use the **Filter** field to quickly search for a specific channel in the list.



Tip You can also **Invert** the input. This inverts the color values (0,0,0 [black] displays as 1,1,1 [white], 0.3,1,1 displays as 0.7,0,0, and so on). The **Invert** function is useful for painting masks. Masks need to be white on black, where the white areas are masked. If you prefer to paint black on white, you can use a shader that inverts the current channel. The shader makes the channel appear black on white, while actually painting in the correct format.

6. To remove a module, click .

Mari asks you to confirm the deletion, then removes the module.

Tip Shaders start with a single module by default (**Color (Diffuse)**). You can remove this module.

7. You can temporarily turn off individual modules within the shader by unchecking .

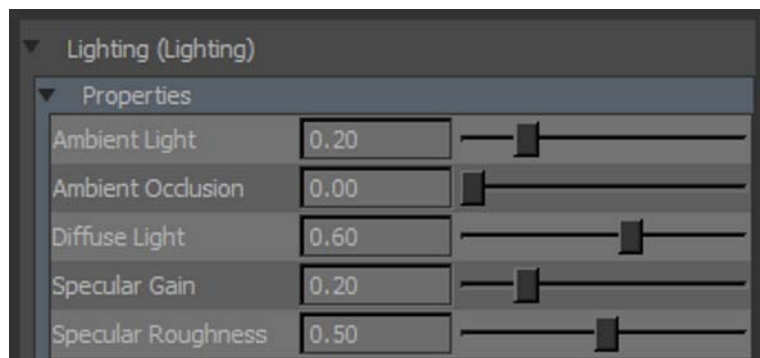
The model remains part of the shader, but Mari ignores it until you turn it back on.

8. You can move modules around within the shader by dragging them up and down the list.

Mari applies the shaders in order, starting at the bottom of the list and working up. Changing the order of the modules can change how the final shaded surface looks.

Tip *Put specular shaders towards the top of the list. Ideally, the specular module should be right under the lighting module (that is, applied right before the final lighting).*

9. In the **Default (Lighting)** module (at the top of the set of modules), set the default lighting details. These control the initial settings for lighting, which Mari supplements with the lights defined in your project. Set the **Default** values for the:
- **Ambient Light** — the 'directionless' light on the surfaces.
 - **Ambient Occlusion** — the amount of shadowing from features on the surface itself.
 - **Diffuse Light** — the directed light on the surface.
 - **Specular Gain** — how bright specular lights are.
 - **Specular Roughness** — how "rough" the surface is. Mari spreads the highlight more across rougher surfaces, and concentrates specular lights more on smoother surfaces (with lower values). A very smooth surface just reflects the specular light straight back.



Sandbox *Since the shaders don't affect the content of the channels, you can safely play around without worrying about overwriting anything. Try building the example below. Download the blacksmith sample project from the Mari web site, if you haven't already. Load all the channels. Try:*

- *Creating a new shader module.*
- *Setting the **Texture** for the initial **Base Diffuse** module to the **color** channel.*
- *Adding a **Masked Diffuse** module, using the **veinColor** channel as the **Base Texture** and the **veinMask** channel as the **Mask Texture**.*
- *Adding a **Displacement** or **Bump** module, using the **dispFine** channel as the **Texture**.*

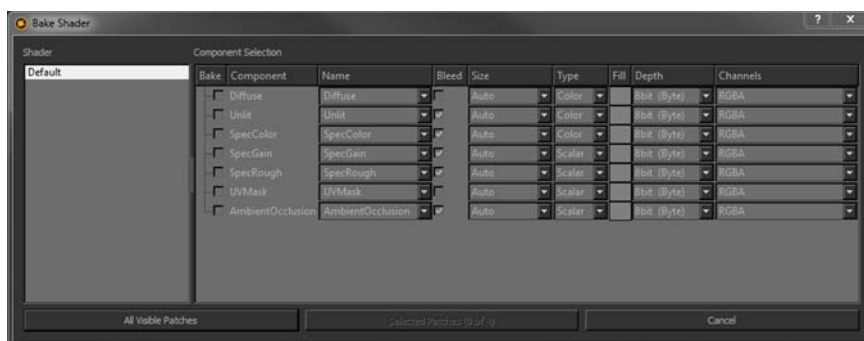
Experiment with the other shader modules, using the channels in the project

as inputs.

Bake a Shader Component onto a Channel

1. To bake a shader down onto a channel, from the **Channels** menu, select **Bake Shader**.

The *Bake Shader* dialog box opens:



2. Select the **Shader** for Mari to use to generate the surface of the new or selected channel(s).
3. From the **Component** list, check the boxes to select which component of the shader (which part of the data that the shader produces) you want Mari to bake:

- **Diffuse** — diffuse surface color.
- **Unlit** — unlit surface color.
- **Ambient** — ambient lighting degree.
- **SpecColor** — specular color.
- **SpecGain** — specular gain amount.
- **SpecRough** — specular roughness.
- **UVMask** — UV patch data.
- **AmbientOcclusion** — ambient occlusion data.

Mari creates a new channel for each selected component. For example, if you select **Diffuse**, Mari creates a new channel containing only the diffuse color information from the shader. Mari ignores any other information from the shader (such as specular details or displacement data).

4. Enter the name of the destination channel in the **Name** entry box. This is set to the name of the component by default.
5. Check the **Bleed** checkbox to set whether bleed is generated automatically for the baked shader.
6. Click the button to choose whether to bake **All Visible Patches** in the project or only the **Selected Patches**.

Mari extracts the selected components from the shader, and bakes them onto a new channel.

Note *When baking a shader’s ambient occlusion data to a new channel, the resolution of the ambient occlusion is determined by the size of the largest patch in the shader.*

Example: Shader
Component in a
Channel

Here is an example of baking a shader into a channel:

Shaded surface	Component	Resulting channel
	Diffuse	

Sandbox *In your sample project, try baking the **Diffuse** component of your sample shader. Open the **Channels** palette and have a look at it.*

Then try:

- *Baking the ambient occlusion for your project to a channel, then create a shader with:*
 - *the **Base Diffuse** set to the color channel*
 - *an **Ambient Occlusion** module set to the ambient occlusion channel.*
- *Edit your ambient occlusion channel, and note the effect on the shaded surface. Try:*
 - *painting black or white onto the surface*
 - *applying a **Contrast** filter to the channel - you can watch the shading update as you preview the filter effect.*

Create a Layered
Shader

To create a layered shader (assuming the channels in your project were imported from a layered .psd file):

1. Click on the **Create Layered Shader** button in the **Shaders** palette or click **Shading > Create Layered Shader**.

A new shader appears in the list of available shaders.

The new shader contains a **Base Diffuse** module and, depending on how many layers were in the original **.psd** file, one or more **Diffuse Blend** modules. It is assigned the name of the original **.psd** file, but you can change this if you like by double-clicking on the shader name and renaming it.

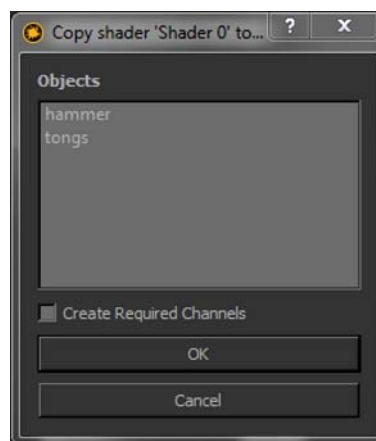
2. Every shader module is automatically assigned a channel (layer) as an input texture. To change this, click on the **Texture** dropdown and select another channel.
3. Adjust the **Blend Amount** and, for **Diffuse Blend** modules, the **Blend Mode**.

Note *When importing a layered **.psd** file, Mari retrieves the blend mode information from the file and automatically sets the **Blend Mode** for each layer in the shader. Mari supports most of the common blend modes available in other paint applications. In situations where the exact blend mode is not available, Mari matches it as closely as possible.*

Copy an Existing Shader to Another Object

1. To copy an existing shader to any number of objects within your project, right-click on a shader and select **Copy Shader**.

*The **Copy Shader** dialog box opens:*



2. Select the object(s) you want to copy the shader to.
3. Check the **Create Required Channels** checkbox to automatically create channels that exist in the selected shader but not on the destination object.
4. Click **OK** to copy the shader to the selected object(s).

Mari copies the shader to the selected object(s).

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. To set up a basic channel on objects that do not need flow maps, in the **Channels** palette, either:

- right-click on the palette and select **Add Channel** from the dropdown menu, or

- click on the  button.

*Mari displays a **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) with a Scalar **Color Space**. Create the channel for only the selected object. For this example, called Object A.

Tip *Create the channel with a 256x256 **Size** to avoid taking up unnecessary space.*

3. With Object A still selected, in the **Shaders** palette, create a **Vector Data** shader module, and set the **Texture** to the newly-created "gray" channel. Set the **Texture** for the **Base Diffuse** shader module to "diffuse", or any channel you want to see without the markers.

*The **Vector Data** module, when set up this way, blocks out the vector inspector markers on objects you don't want to create a flow map for.*

4. 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.
5. Select a second object for which you want to create a flow map. For this example, the second object is called Object B. Create a new channel, following the instructions in steps one and two above. Name the channel "flow" or similar.

This allows you to set up a specific flow map channel where the vector markers are present, without affecting other objects.

6. With Object B still selected, in the **Shaders** palette, create a **Vector Data** shader module, and set the **Texture** to the "flow" channel. Set the **Texture** channel for the **Base Diffuse** shader module to "diffuse" or any channel you want to see beneath the vector markers on this object.

*If you paint on the model now, paint is applied to all unlocked objects. However, only objects with the "flow" channel listed in the **Vector Data** module's **Texture** field show changing vector inspector markers.*

Save, Load, and Delete Shaders

1. To save a shader, in the **Shaders** palette, either:
 - right-click on an existing shader and select **Save Shader** from the dropdown menu, or
 - click on the  button.

*Mari displays a **Save As** dialog box, for you to choose a location on disk and set the shader's name. The file extension is **.msh** (Mari shader).*

2. To load a saved shader file, in the **Shaders** palette, either:
 - right-click and select **Load Shader** from the dropdown menu, or
 - click on the  button.

*Mari displays a **Load Shader** dialog box, for you to find the shader file. This is a **.msh** file saved previously.*

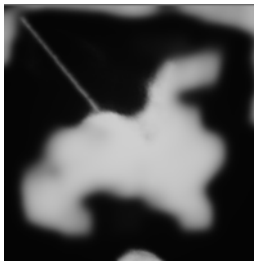
3. To delete a shader, either:
 - in the **Shaders** palette, right click on the shader and select **Remove SurfaceShader** from the dropdown menu, or
 - click on the  button.

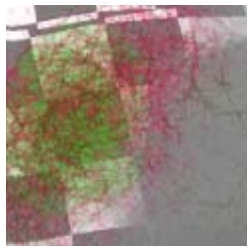
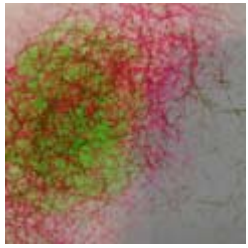
Mari removes the selected shader.

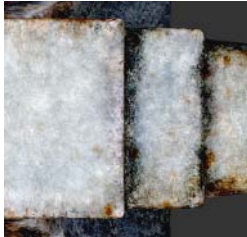
Shader Module Types and Examples

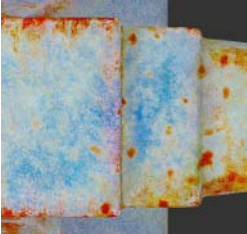
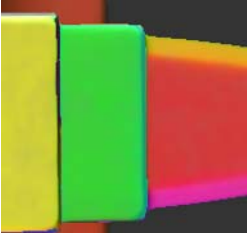
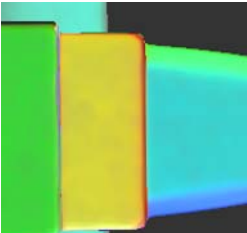
Below are descriptions and examples of the types of modules available for building a shader in Mari.

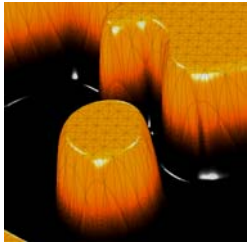
Type	Description	Settings	Example	
Ambient Occlusion	Applies an ambient occlusion shading to the surface. This overrides Mari's default ambient occlusion shader, allowing you to change the details. For example, you can edit the channel to use as an ambient occlusion map. If you want to save the default ambient occlusion data for the project as a channel, you can bake the ambient occlusion from the default shader. You can then edit this baked channel as you want. 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.	AO Texture — the channel containing the ambient occlusion data.	The original, unshaded surface:	AO Texture channel:
			The shaded result:	

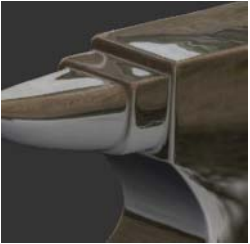
Type	Description	Settings	Example	
Bump	Provides a quick preview of a displacement, by using a simple bump map.	• Texture — the channel containing the displacement data. • Bump Weight — how much bump is applied. Lower values are smaller bumps, higher values are larger and more obvious displacements.	Displacement map: 	The result: 
Debug	Displays a graphic representation of the properties of the shaded surface. For example, the Spec Rough Amount setting displays the specular roughness on the current shaded surface as a color map. These settings are intended for debugging your shaders.	For each of the settings you can select the amount of blend between the displayed information and the shaded surface. Each is from 0 to 1; 0 is totally transparent, 1 is solid color. • Tangents Amount — displays the tangents on your model's surface. • Bitangents — displays the bitangents on your model's surface. • Spec Gain Amount — displays the level of specular gain on the shaded surface. • Spec Rough Tangents Amount — displays the level of specular roughness on the shaded surface. • Spec Color Tangents Amount — displays the specular color of the shaded surface. • Luminosity Amount — inverts the current luminosity (that is, higher values display as darker). • UVs Amount — displays a representation of the UV patches that make up the model. • Normal Tangents Amount — displays the normal values of the shaded surface.	With Tangents set to 1: 	With Spec Gain Amount set to 0.8: 

Type	Description	Settings	Example	
Debug Position	Displays the graphic representation of the surface positions (XYZ as RGB).	• Position Scale – displays the scale of the RGB position. From 0.001 to 100; the RGB scale becomes sharper as the slider is moved towards 100.	With Position Scale set to 1.000:	
Diffuse	Adds the selected input to the shaded surface.	• Texture — the channel to add to the shaded surface. • Blend Amount — how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color.	The initial surface:	Input Texture :
				
			The result, with a Blend Amount of 0.5:	
				

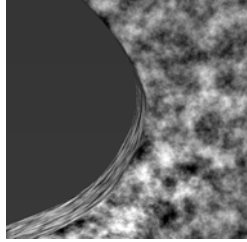
Type	Description	Settings	Example	
Diffuse Blend	Adds the selected input to the shaded surface, using the selected blend mode.	• Texture — the channel to add to the shaded surface. • Blend Amount — how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Mode — the blend mode to use when adding the texture to the shaded surface. The blend modes available are Mari's standard Paint Blending Modes. The default mode is to add the texture on top of the shaded surface, but other modes allow you to darken, multiply, burn, and so on.	The initial surface:	Input Texture :
			 Result with Blend Amount of 0.96 and Mode set to Multiply : 	 Result with Blend Amount of 0.75 and Mode set to Exclusion : 
Diffuse Brightness/Contrast	Applies a brightness and contrast adjustment to the current shader's diffuse.	• Adjust Brightness — adjusts the brightness of the current shader's diffuse. • Adjust Contrast — adjusts the contrast of the current shader's diffuse.	The initial surface:	With brightness and contrast adjusted:
				
Diffuse Color Curves	Adjusts the current shader's color correction using curve editors.	• Red — adjusts the red value in the current shader. • Green — adjusts the green value in the current shader. • Blue — adjusts the blue value in the current shader.	The initial surface:	With a color correction applied:
				

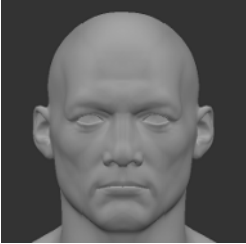
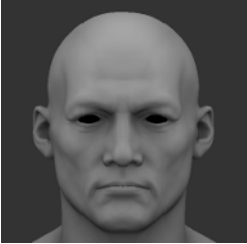
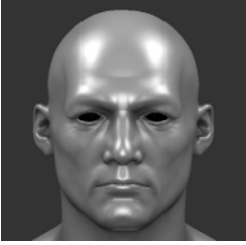
Type	Description	Settings	Example	
Diffuse HSV	Adjusts the hue, saturation, and value of the current shader's diffuse.	• Hue Shift — adjusts the hue shift of the current shader's diffuse. • Sat Scale — adjusts the saturation scale of the current shader's diffuse. • Value Scale — adjusts the value scale of the current shader's diffuse. • Calculation — determines whether the colorspace is set to use the Mari default Original [HSV], or Photoshop [HSL].	The initial surface: 	With the hue, saturation, and value scales adjusted: 
Diffuse Tangent to Screen	Converts the diffuse RGB, used as color-encoded vectors, from tangent space to screen space.	• Suppress Blue - suppresses the blue value from the color-encoded vectors in screen space.	The initial surface: 	Without the Suppress Blue option ticked: 
Diffuse Tangent to World	Converts the diffuse RGB, used as color-encoded vectors, from tangent space to world space.	None	The initial surface: 	Converted to World Space: 

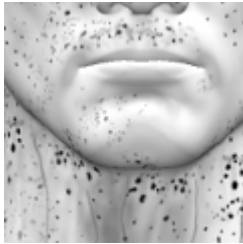
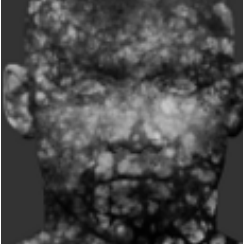
Type	Description	Settings	Example	
Displacement	Provides a more advanced preview of a displacement using dynamic tessellation, and a displacement map as the input. Note: For more precise displacement, the Virtual Texture Type, under Edit > Preferences > Gpu 0, should be set to at least Half when using the displacement shader module. In addition, the channel containing the displacement map should be set to at least 16-bit. Please note that using higher values here may reduce preview performance. This module is disabled for Ptex channels. Not all graphics cards support displacement preview. Please refer to the list of recommended graphics cards in the Getting Started Guide for more information.	• Texture — the channel containing the displacement data. • Invert — select to invert the values taken from the Texture. • View Tessellation — determines how many pixels the surface is tessellated to. • Texture Tessellation — determines how many texels the surface is tessellated to. • Texture Weight — how much displacement is applied. Lower values equal smaller displacement, higher values equal larger and more obvious displacement. • Texture Bias — the starting point at which white or black values are pushed or pulled from the surface of the object. The default value is 0.500, which is mid-gray.	Displacement map:	Color map:
			 Result, with a medium Texture Weight value: 	 Zoomed in, with the wireframe on: 
Emissive	Creates a shader that simulates surfaces that emit light or glow. This shader is lighting-independent.	• Texture - the channel containing the shaded surface. • Invert - select to invert the values taken from the Texture. • Multiply - the strength of the illumination on the Texture channel. • Blend Amount - how much the Emissive shader is blended into the Texture channel.		

Type	Description	Settings	Example	
Environment Cube	Simulates a shiny reflective surface using a cubic environment map as the input.	• Cubic Image – the location of the cube map. • Blend Amount – how strongly the cubic image displays on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Falloff Start – controls where the cubic image starts. • Falloff End – controls where the cubic image ends.	The initial surface before the cube map is applied: 	Result with the cube map and a Blend Amount of 0.5: 
Environment Sphere	Simulates a shiny reflective surface using a spherical environment map as the input.	• Sphere Map — the location of the environment map. • Blend Amount — how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Falloff Start – controls where the spherical image starts. • Falloff End – controls where the spherical image ends.	The initial surface before the environment map is applied: 	Result with the environment map and a Blend Amount of 0.5: 

Type	Description	Settings	Example
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, on the Flow shader, the Speed attribute is removed in that area. The vector inspector directional markers are still shown on the model. This has to do with how the Flow shader module handles the alpha channel.	• Texture - the channel containing vector data. • Invert - inverts the directions of the vector paint. • Blend Amount - how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode - the painting blend mode used to apply the tile image input to the shaded surface. • 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. • Tile Image - the image used on the model for flow. • Repeat - sets the repetition value of the tiled image.	In the example below, the Flow shader would animate the molten liquid so that it appeared to moving across the surface of the texture: 

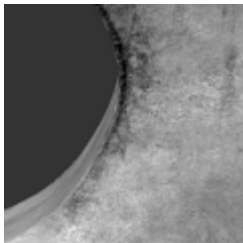
Type	Description	Settings	Example	
Fractal Noise	Blends seamless 3D noise with the shaded surface, using the specified Blend Mode .	• Blend Amount — how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Seed — changes the pattern of the fractal noise by moving the pattern start point through 3D space. • Color A — sets the first color of the fractal noise features. • Color B — sets the second color of the fractal noise features. • Size — determines the size of the fractal noise features. From 0 to 1; 0 is totally transparent, 1 is solid color. • Roughness — determines the roughness of the fractal noise features. From 0 to 1; 0 is totally transparent, 1 is solid color. • Contrast — controls the level of contrast of the fractal noise before it is blended with the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Curve — controls how the fractal noise relates to color A and B, as well as allowing fine-tuned control over levels of contrast for each. • Blend Mode — the blending mode used. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.	The initial surface: 	Fractal Noise: 

Type	Description	Settings	Example	
Height Map As Normal Map	Interprets the selected 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.	• Texture – selects which texture channel the shader should apply to. Default is Current Channel. • Invert – inverts the texture to be generated as RGB values. • Bump Weight – how much bump is applied. Lower values are smaller bumps; higher values are larger and more obvious bumps.	Texture in the diffuse channel: 	Bump Weight set at 10.000: 
Lighting	Sets the default lighting details. These control the initial settings for lighting, which Mari supplements with the lights defined in your project.	• Ambient Light — the amount of ambient light the shader uses to render the surface. • Ambient Occlusion — the amount of ambient occlusion the shader uses to render the surface. Note that in order for this to calculate correctly, you need to run the ambient occlusion (Objects menu, select Ambient Occlusion) on each object in your project. Otherwise, everything is shaded equally, instead of the occluded areas being shaded more. • Diffuse Light — the amount of diffuse light the shader uses to render the surface. • Specular Gain — the amount of light the shader uses when generating specular values. • Specular Roughness — how “rough” the surface is by default. Rougher surfaces have the highlight spread more across the surface. Lower values mean the surface is smoother (specular lights are more concentrated), higher values mean it is rougher (specular lights are spread out more).	The initial surface with default lighting applied: 	With the Ambient Occlusion set to 1:  With the Ambient Occlusion set to 1 and Specular Gain set to .75: 

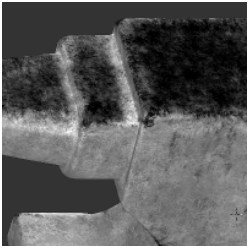
Type	Description	Settings	Example	
Mask to Alpha	Shows the areas of the shaded surface that correspond to the masked areas on the selected input.	• Mask Texture — the channel (or other input) containing the mask data. • Mask Amount — how much to blend the masked channel with the shaded surface. Lower values show more of the original surface, high values show only the masked areas. • Mask Component — the color channel on the Mask Texture to use as the mask data. Select a color channel. If the mask is black and white, any color channel works.	The initial surface 	Mask Texture 
			The result, with a Mask Amount of 1.0 and the Mask Component of R: 	
Normal	Shows the input texture as a normal map. Mari assumes that the map is normalized around 0.5.	• Texture — the channel containing the normal mapping data. • Invert — Select to invert the normal data input. • Invert R — Select to invert the red channel in the normal data input. • Invert G — Select to invert the green channel in the normal data input. • Invert B — Select to invert the blue channel in the normal data input.	The initial surface 	Texture: 
			The result: 	

Type	Description	Settings	Example
Projection Cube	Uses a cubic image (in a .dds file format) as a texture to project onto the model.	• Cubic Image - the filepath to the cubic image to use for projection. • Blend Amount - how strongly the image is projected on the model. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode - select the blend mode applied to the image projected on the model. This is similar to the painting mode menu found in the Paint toolbar. • 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. • Cull Backfaces - when the checkbox is ticked, it ensures that projection does not affect areas facing away from the camera. • 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 projection. • Distance Falloff Curve - allows you to adjust the otherwise linear shape of the falloff to your desired shape.	

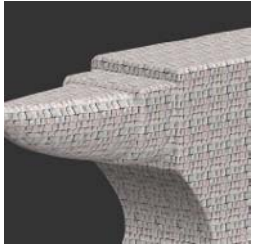
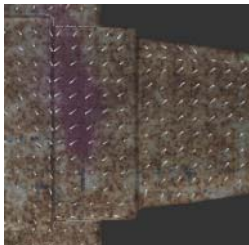
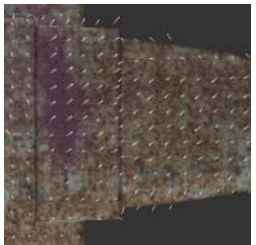
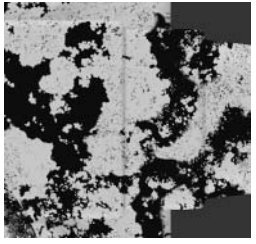
Type	Description	Settings	Example
Projection Sphere	Uses a spherical image of any file type as a texture to project onto the model.	• Spherical Image - the file-path to the spherical image to use for projection. • Blend Amount - how strongly the image is projected on the model. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode - select the blend mode applied to the image projected on the model. This is similar to the painting mode menu found in the Paint toolbar. • 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. • Cull Backfaces - when the checkbox is ticked, it ensures that projection does not affect areas facing away from the camera. • 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 projection. • Distance Falloff Curve - allows you to adjust the otherwise linear shape of the falloff to your desired shape.	

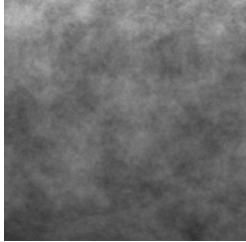
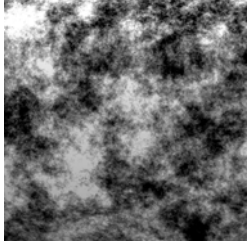
Type	Description	Settings	Example	
RGBFilter	Filters the existing shaded surface by controlling which of the Red, Green and Blue color channels display.	You can select which of the three color channels (and alpha channel, if applicable) to display.	The initial surface:	Displaying only Blue and Red channels:
				
			Rough Texture:	Color Texture:
				
			And the result, with all values at 1.0:	
				

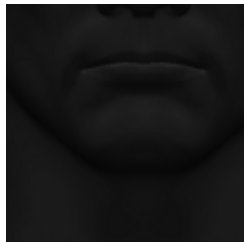
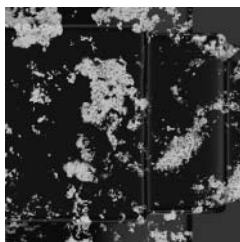
Type	Description	Settings	Example	
Specular Color	Sets the specular color. This is the color that the shiny areas on the surface reflect.	Color Texture — the color to render specular areas. Select the channel that holds the specular color details.	Initial surface:	Color Texture:
			Result:	
Specular Gain	Sets the specular gain. This controls how shiny the surface is.	Gain Texture — the channel to use as a specularity map.	Initial surface:	Gain Texture:
			Result:	

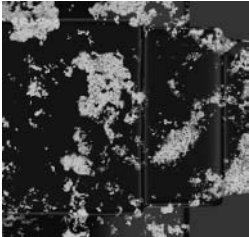
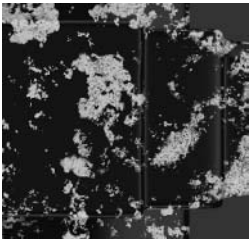
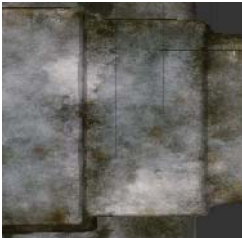
Type	Description	Settings	Example	
Specular Roughness	Sets the specular roughness. This controls how spread out the specular highlight is — rougher surfaces spread the highlights out more.	Rough Texture — the channel to use as a roughness map.	Initial surface:	Rough Texture:
				
			Result: 	
Surface Normals As Normal Map	Converts the present surface normals to a Diffuse RGB color. This is useful for baking out normal maps from bump shader modules.	There are no settings for this module.	Bump Displacement:	Result:
				

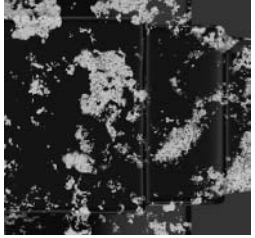
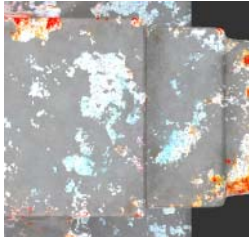
Type	Description	Settings	Example	
Tiled	Lets you select an image to tile across the surface of your model. This module is disabled for Ptex channels.	• Tile Image — the location of the image you want to tile across the surface of your model. • U Repeat — the frequency at which the image is repeated on the U axis. • V Repeat — the frequency at which the image is repeated on the V axis. • 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. • Mirror U — check to mirror the image on the U axis. • Mirror V — check to mirror the image on the V axis. • Blend Amount — how strongly the image appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode — the blending mode used. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.	The initial surface before the image is tiled across the surface: 	Result with the tiled image and a Blend Amount of 0.3: 

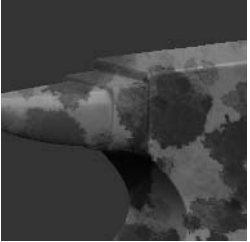
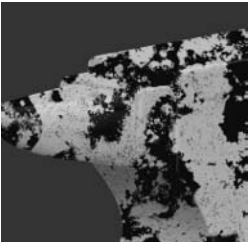
Type	Description	Settings	Example	
Triplanar	Lets you select three different images to project onto the surface of your model from three different directions (Front, Top, Side).	Refer to the Shaders Palette section (Chapter 22) of the Reference Guide for a breakdown of all the available settings in this shader module.	The initial surface:	A tileable texture projected onto the front, top, and side of the model.
				
Vector Data	Sets the vector field in the shaders that is used by the Vector Inspector.	• Texture - the channel to be used for the vector data. • Invert - inverts the vector information in the specified channel.	The initial surface with active Vector Inspector :	Vector Data inverted:
				
Masked Blend Constant Color	Adds a selected color to the surface, using a channel as a mask.	• Base Color — the color to add to the shaded surface. Click the swatch to open a color picker. • Mask Texture — the channel with the mask that controls where to add the color to the shaded surface. • Invert — select to invert the mask data input. • Blend Amount — how much of the input appears on the shaded surface. • Blend Mode — the blending mode used. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.	The initial surface	Mask Texture:
				
			The result, with a Blend Amount of 1.0 and the Blend Mode set to Normal :	
				

Type	Description	Settings	Example	
Masked Bump	Displays an input texture as a bump map and uses a mask texture to control the bump weight.	• Texture — the channel containing the displacement data. • Invert — select to invert the values taken from the Texture. • Mask Texture — the channel (or other input) with the mask that controls which of the inputs from the Texture are added to the shaded surface. • Invert — select to invert the values taken from the Mask Texture. • Bump Weight — how much bump is applied. Lower values are smaller bumps, higher values are larger and more obvious displacements.	Displacement map: 	Mask texture: 
Masked Constant Color	Adds a constant color input to the shaded surface, using a channel as a mask. Areas on the shaded surface that are not covered by the mask are left unchanged.	• Base Color — the color to add to the shaded surface. • Mask Texture — the channel containing the mask data. • Blend Amount — how strongly to add the color to the surface. From 0 to 1; at 1 you get the full value as set by the mask channel, at 0.5 you would get half the weight from the mask.	The initial surface 	Mask Color: 
			The result, with a Base Color of bright red: 	

Type	Description	Settings	Example	
Masked Diffuse	Adds the color from one input to the shaded surface, using a texture as a mask. By default, the diffuse effect is limited to the non-black areas of the mask.	• Base Texture — the channel to add to the shaded surface. • Mask Texture — the channel with the mask that controls where to apply the color from the Base Texture. • Blend Amount — how strongly the added texture appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color.	The initial surface: 	Base Texture: 
Masked Diffuse Blend	Blends the texture over the current diffuse color through a mask using the specified Blend Mode . This module combines the Masked Diffuse and Diffuse Blend modules.	• Texture — the channel to add to the shaded surface. • Invert — select to invert the values taken from the Texture. • Mask Texture — the channel with the mask that controls which of the inputs from the Texture are added to the shaded surface. • Invert — select to invert the values taken from the Mask Texture. • Blend Amount — how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode — the blending mode used. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other blending modes.	The initial surface: 	Base Texture: 
			Mask Texture: 	The result, with a Blend Amount of 0.880 and the mask inverted: 

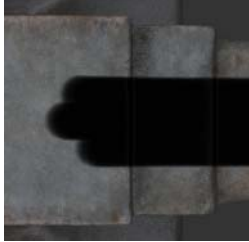
Type	Description	Settings	Example	
Masked Diffuse Brightness/Contrast	Applies a brightness and contrast adjustment to the current shader's diffuse, using a channel as a mask.	• Adjust Brightness — adjusts the brightness of the current shader's diffuse. • Adjust Contrast — adjusts the contrast of the current shader's diffuse. • Mask Texture — the channel with the mask that controls where to apply the brightness and contrast adjustment to the shaded surface. • Invert — select to invert the values taken from the Mask Texture.	The initial surface:	Mask Texture:
				
Masked Diffuse Color Curve	Adjusts the current shader's color correction using curve editors within a mask as a channel.	• Red - adjusts the red value in the current shader. • Green - adjusts the green value in the current shader. • Blue - adjusts the blue value in the current shader. • Mask Texture - applies the color corrections to the specified shader. • Invert - select to invert the colors taken from the Mask Texture.	The initial surface:	Mask Texture:
				
			The result, with the red, green, and blue values adjusted.	
				

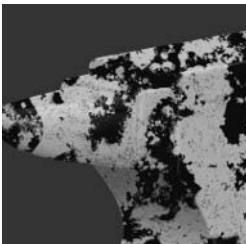
Type	Description	Settings	Example	
Masked Diffuse HSV	Adjusts the hue, saturation, and value of the current shader's diffuse, using a channel as a mask.	• Mask Texture — the channel with the mask that controls where to apply the hue, saturation, and value adjustment to the shaded surface. • Invert — select to invert the values taken from the Mask Texture. • Hue Shift — adjusts the hue shift of the current shader's diffuse. • Sat Scale — adjusts the saturation scale of the current shader's diffuse. • Value Scale — adjusts the value scale of the current shader's diffuse.	The initial surface: 	Mask Texture: 
			The result, with the hue, saturation, and value scales adjusted: 	
Masked Emissive	Creates a masked shader that simulates surfaces that emit light or glow. This shader is lighting-independent.	• Texture - a channel containing the colors that the surface should emit. • Invert - select to invert the values taken from the Texture. • Mask Texture - the channel containing the mask. • Invert - select to invert the values taken from the Mask Texture. • Multiply - the strength of the illumination on the Mask Texture channel. • Blend Amount - how much the Emissive shader is blended into the Texture channel.		

Type	Description	Settings	Example	
Masked Environment Cube	Simulates a shiny reflective surface using an environment map as the input and a channel as a mask.	• Cubic Image – the location of the cube map. • Blend Amount – how much of the input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Falloff Start – controls where the cube reflection starts on the model. • Falloff End – controls where the cube reflection ends on the model.	The initial surface: 	Base Texture: 
Masked Environment Sphere	Simulates a shiny reflective surface using an environment map as the input and a channel as a mask.	• Sphere Map — the location of the environment map. • Blend Amount — how much of the input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Mask Texture — the channel with the mask that controls where to apply the environment map to the shaded surface. • Invert — select to invert the values taken from the Mask Texture. • Falloff Start — controls where the environment reflection starts on the model. • Falloff End — controls where the environment reflection ends on the model.	Initial surface: 	Mask Texture: 
			Result with a environment map applied and a Blend Amount of 1.000: 	

Type	Description	Settings	Example
Masked Flow	Displays a masked image onto the model's surface and uses vector data to animate it flowing. Note: if painting in the Clear blending mode, on the Flow shader, the Speed attribute is removed in that area. The vector inspector directional markers are still shown on the model. This has to do with how the Flow shader module handles the alpha channel.	• Texture - the channel containing vector data. • Invert - inverts the directions of the vector paint. • Mask Texture - the channel containing the masked vector data. • Invert - inverts the directions of the masked vector paint. • Blend Amount - how strongly the added input appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode - the painting blend mode used to apply the tile image input to the shaded surface. • 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. • Tile Image - the image used on the model for flow. • Repeat - sets the repetition value of the tiled image.	In the example below, the Flow shader would animate the molten liquid so that it appeared to moving across the surface of the texture: 

Type	Description	Settings	Example
Masked Projection Cube	Uses a cubic image (in a .dds format) as a texture to project onto the model, with a mask option to mask out parts of the projection.	• Cubic image - the filepath to the cubic image to use for projection. • Mask Texture - allows you to mask out channels from the image. Used, for example, when you might need to overlap other scene projection. • Invert - inverts the masked texture to show what's in the mask. The rest of the settings for the Masked Projection Cube are the same as for the Projection Cube module.	
Masked Projection Sphere	Uses a spherical image of any file type as a texture to project onto the model, with a mask option to mask out projection.	• Spherical Image - the filepath to the spherical image to use for projection. • Mask Texture - allows you to mask out channels from the image. Used, for example, when you might need to overlap other scene projection. • Invert - inverts the masked texture to show what's in the mask. The rest of the settings for the Masked Projection Sphere are the same as for the Projection Sphere module.	

Type	Description	Settings	Example	
Masked Tiled	Adds a tiled image texture to the shaded surface, using a channel as a mask. Areas on the shaded surface that are not covered by the mask are left unchanged. This module is disabled for Ptex channels.	• Tile Image — the location of the image you want to tile across the model and add to the shaded surface. • 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. • 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. • Mirror U — check to mirror the image on the U axis. • Mirror V — check to mirror the image on the V axis. • Alpha Mask — the channel with the mask that controls which of the inputs from the Tile Image are added to the shaded surface. • Invert — select to invert the values taken from the Alpha Mask. • Blend Amount — how strongly the image appears on the shaded surface. From 0 to 1; 0 is totally transparent, 1 is solid color. • Blend Mode — the blending mode used. The default is Normal (paint in the buffer overwrites the surface), but Mari supports a number of other Paint Blending Modes.	The initial surface:	Base Texture:
				
			Mask Texture:	Result:
				

Type	Description	Settings	Example	
Masked Triplanar	Lets you select a number of images to tile across the surface of your model using a channel as a mask.	Refer to the Shaders Palette section (Chapter 22) of the Reference Guide for a breakdown of all the available settings in this shader module.	The initial surface:	Mask Texture:
				
			The result:	
				

23 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:

- the entire contents of a channel,
- only the current unbaked paint (that is, the contents of the paint buffer), or
- an image within the **Image Manager**.

Filters apply a transformation to the painting on the model. For example, Mari ships with filters to:

- Apply a variety of blurs to the painting.
- Invert colors.
- Change the hue, saturation, or value of the painting.
- Remove specific color channels from the painting.
- Apply color correction.
- Add 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 use the **Marquee Select** tool, the **Current Channel**, or alternatively, 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 channels.

As well as using Mari's predefined filters to edit the paint on a channel or in the paint buffer, you can use them as part of a custom color profile. These filters only affect how the colors display onscreen (rather than actually changing the painting itself). See [Managing Colors in Mari](#) 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.*

Steps for Using Filters

To use filters in Mari, follow this step:

1. [Select and Apply Your Filter](#)

This tutorial also includes a reference table of [Available Filters](#).

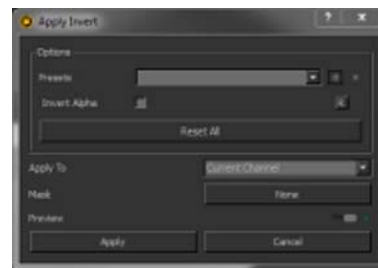
Select and Apply Your Filter

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 active in the **Channels** palette
 - an image, select it in the **Image Manager**.
4. If you plan to mask an area of the **Current Painting** 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 Painting**.*

5. 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**.*

6. 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 .
7. Select whether the filter should be applied to:
 - everything on the **Current Channel** (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**).
8. Select whether to use a **Mask**:
 - If you're applying the filter to the **Current Channel**, you can use the **Current Channel** or any other channel in the project as a mask.
 - If you're applying the filter to the **Current Painting**, you can use the **Marquee Select** 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](#).*

9. Toggle the **Preview** switch to see the filter's effects before applying them.

Tip *Double-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**.*

10. Click **Apply**.

Sandbox *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**.*

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:



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.	

Filter	Description	Options	Example
Luminosity	Outputs just the luminosity value of each pixel in the selection. That is, it outputs a greyscale image based on the brightness of each of the pixels in the original image.	None.	
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	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 lines on the boundaries between patches. This is an artifact of the way that Mari handles the edges of patches.	

Filter	Description	Options	Example
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.	
sRGB To Linear	Applies an sRGB to linear color-space conversion.	Checking Invert applies a linear to sRGB color-space conversion.	

Filter	Description	Options	Example
Hue	Changes the hue, saturation, and value of the colors on the model.	Set the Calculation before you begin, to determine whether to use the Original (HSV) colorspace used as default by Mari, or the Photoshop (HSL) colorspace. 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 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 pre-set values. You can edit the Value, 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.	

Filter	Description	Options	Example
Contrast	Changes the contrast of the paint.	You can select the amount of contrast using the slider.	
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 entry boxes. Select the individual Color Component to clamp in the dropdown menu.	

Filter	Description	Options	Example
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 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 • right-clicking to select from a set of pre-set values.	

Filter	Description	Options	Example
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 greyscale 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 pre-multiplied alpha onto one with it, use this filter to perform the pre-multiplication, so the images match and you avoid lines around the outside of the patch. (Post-multiply works the same, but in reverse (removes pre-multiplication to match images that do not have pre-multiplied alpha).	If the Postmultiply box is checked, this acts in reverse (that is, removes pre-multiplied alpha).	
Edge Detect	Finds "edges" (transitions between colors) in the image and heightens them.	None.	
Emboss	Applies an "emboss" effect, so that textures look "raised" on the surface.	None.	

Filter	Description	Options	Example
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.	
Color Correction	Applies a colorspace correction to the current channel or painting.	You have the option to use Mari's default Configuration File or a custom one. 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.	



24 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 unproject. This lets you unproject multiple channels at once.

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.

Steps for Using Projectors

To use projectors in Mari, follow these steps:

1. [Set the View](#)
2. [Create the Projector](#)
3. [Unproject from Your Projector](#)
4. [Edit the File\(s\) Externally](#)
5. [Project the File\(s\) Back onto the Model](#)
6. [Unproject Multiple Channels](#)
7. [Use the Quick Project Functions](#)
8. [Use the Layered Painting Project Functions](#)
9. [Project onto a Model Using .fbx Files](#)
10. [Generate a Turntable from a Projector](#)
11. [Save, Load, and Delete Projectors](#)

Set the View

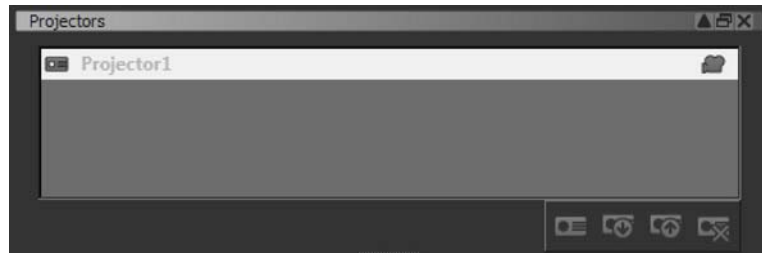
1. Start Mari and open your project.
2. Adjust the view how you'd like it. The projector takes snapshots of the contents of the paint buffer, exactly as they appear onscreen, so:
 - switch to your preferred camera view (UV, orthographic, or perspective)
 - rotate, spin, and zoom the model so you can 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.



Create the Projector

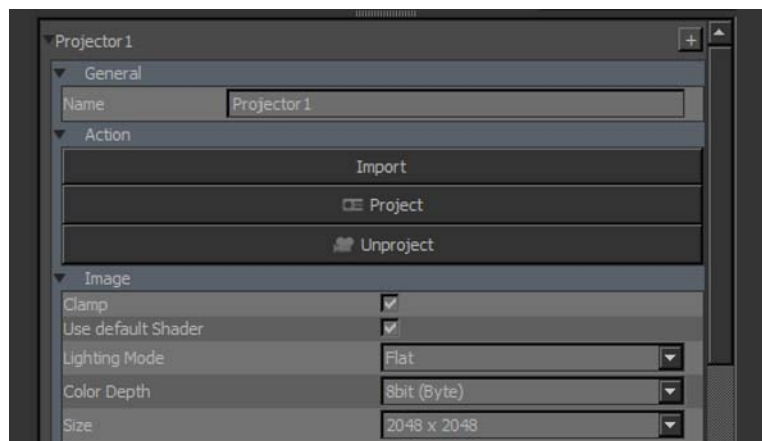
1. In the **Projectors** palette, either:
 - right-click and select **Create Projector** from the dropdown menu, or
 - click the  button at the bottom of the projectors list.

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 **Image** 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.
 - If **Use default shader** is checked, the projector takes a shot using Mari's default shader. If this is unchecked, the projector uses the shader in use when you take the shot.
 - 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.

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.*

*You can also include **\$CHANNEL** in the input file path.*

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 **Projection** 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 **Projection** palette.*

Unproject 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.

Edit the File(s) 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.



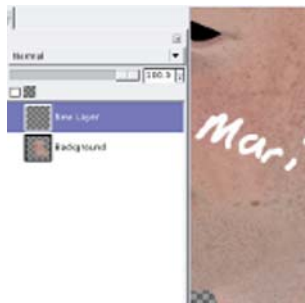
We edit the whole snapshot, even though we only want to change one small section.



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.

Project the File(s) 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.*

Sandbox *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.*

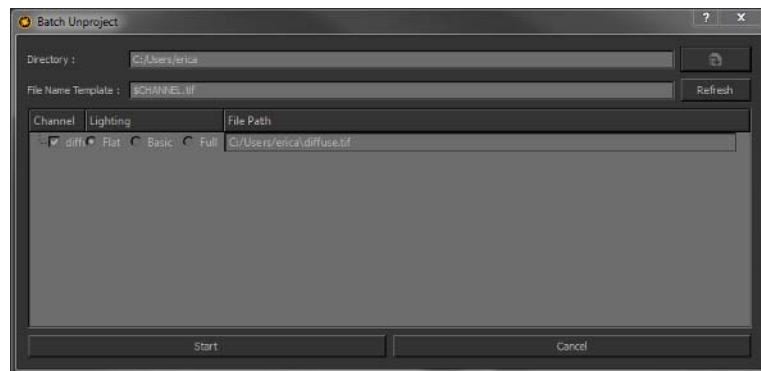
Unproject Multiple Channels

Mari has a batch mode to unproject from multiple channels at once.

To unproject multiple channels at once:

1. Right-click on the projector in the **Projectors** palette and select **Batch Unproject** from the dropdown menu.

*The **Batch Unproject** dialog box displays:*



2. Set the details for the batch unproject:

- Select the **Directory** to store the unprojected files.
- If you want to change the filenames from the default (each file named after the channel it contains), set the **File Name Template**. You can use any standard Mari variables in the template.
- Select the channels to unproject from the list.

*Mari exports each checked channel, using the selected shader and lighting mode, to the listed **File Path**. You can change the details of each channel. You can also right-click on the list to get options to select all channels, select no channels, or set all channels to use a particular lighting mode.*

3. Click **Start**.

Mari unprojects all the selected channels. Each channel in the unprojection is a file on disk.

Tip *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.

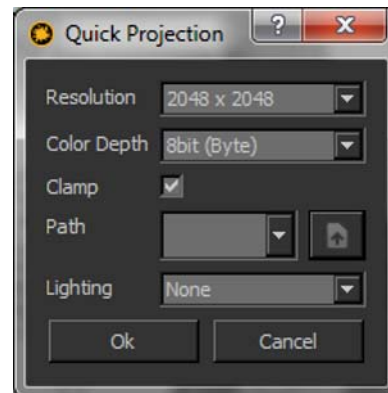
Use the Quick Project Functions

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 [Edit the File\(s\) 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

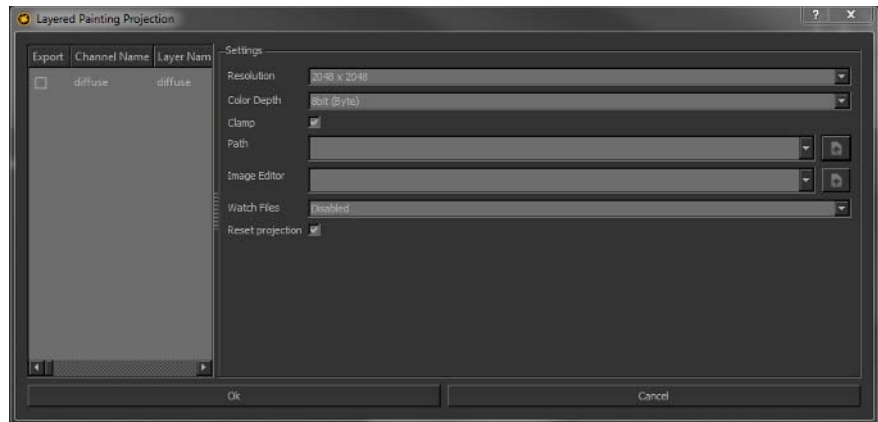
*Mari loads the updated image from the location set in the **Quick Projection** dialog box and projects it back onto the model.*

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**.

Use the Layered Painting Project Functions

The Layered Painting Project functions provide a quick and easy way to project and unproject layered **.psd** files. They let you project and unproject from the current view without having to create a projector first. Setting this up is easy:

1. Click **Camera > Layered Painting Projection Settings**.



The Layered Painting Projection settings dialog displays.

2. In this dialog, set:
 - the channels to unproject
 - the **Resolution** of the layered **.psd** file
 - the **Path** to where the **.psd** file is saved
 - the location of the external **Image Editor** that you want to use to edit the **.psd** file
 - if the **.psd** file is watched (whether changes to the **.psd** file in your external image editor are automatically applied in Mari)
 - whether the original positions of the camera and paint buffer are restored before the painting is projected back onto the model (**Reset Projection**).
3. Click **OK** to apply.

To unproject the current view of the model:

1. Click **Camera > Layered Painting Unproject**. This unprojects the current view to a **.psd** file as specified in the **Layered Painting Projection** settings dialog. If you have set up an external image editor, it automatically opens with the layered **.psd** file loaded.
2. Paint on or edit the image in your image editor.

Note *For layered .psd files you must paint on the layers labeled [channelname]_Paint, otherwise you cannot see the changes when the image is projected back onto your model in Mari.*

In addition, the layers for the unprojected channels are locked so any modification to them is ignored.

3. Save your painting.
4. If you chose to watch the .psd file, Mari displays a dialog asking what you would like to do with the image. You can:
 - **Project Image**—Project the image back onto the model, where modifications to the layers [channelname]_Paint are projected back onto channels called [channelname].
 - **Ignore**—Do nothing.
 - **Stop Watching File**—Do nothing and stop watching the .psd file.

If you chose to not watch the .psd file, nothing happens. If you like, you can manually project the image back onto the model by clicking on **Camera > Layered Painting Project**.

Project 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 .

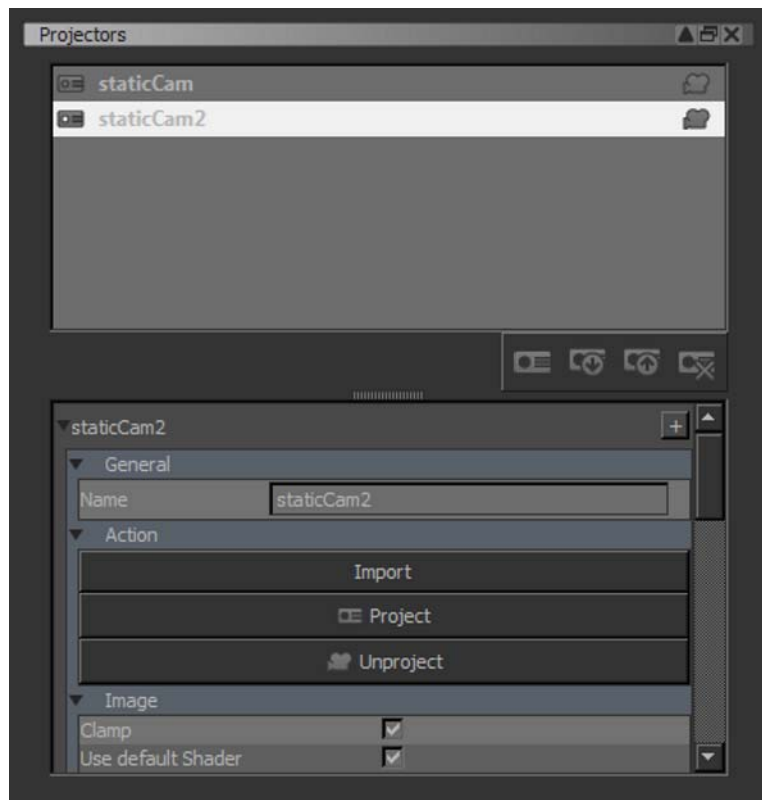
*The **Load Projector** dialog box displays.*

3. Locate the **.fbx** file associated with your model and click **Open**.



*The **Select Entity** dialog box displays.*

4. Click **Select All** to highlight all the cameras, then click **Import**.
*Mari imports the camera data from the **.fbx** file and creates a corresponding number of projectors in the **Projectors** palette.*



5. 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.
6. Select the required file and click **Open**.



7. 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.*

Mari projects the texture onto the model as specified.

- Repeat steps 5 and 6 for all the imported projectors to complete the process.
- Once you're happy with the results, you can export your work as described in [Export Textures](#), [Generate a Turntable from a Projector](#), or [Save, Load, and Delete 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.*

Sandbox *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?*

Generate 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.

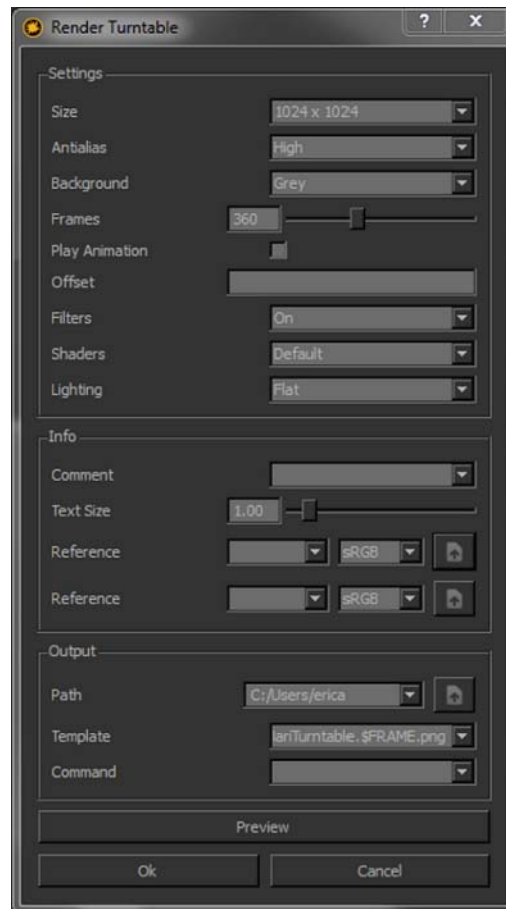
Tip *Turntables always rotate around the center of your current view. That is, the axis for a turntable is a line straight up the middle of the screen in the current view. This is in the center of the visible portion of the model.*

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 drop-down menu.

*The **Render Turntable** dialog box displays.*

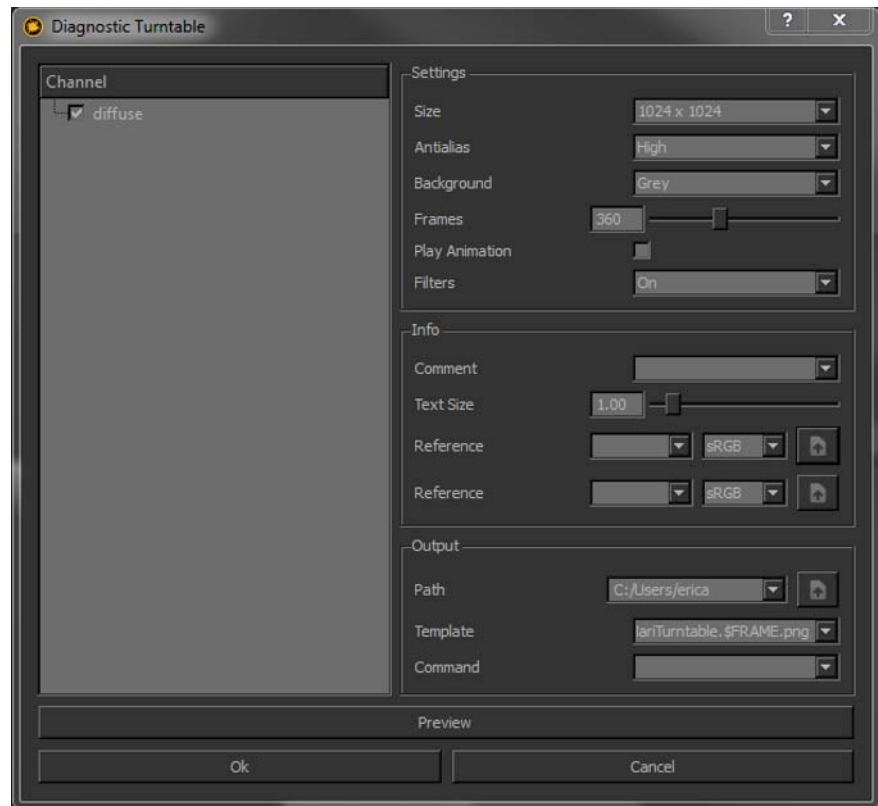


2. Set the basic image details:
 - **size** of the generated images
 - the amount to **Antialias** the images.
 - **background** for the images — either grey, transparent or the current background in Mari
 - the number of **Frames** to generate — that is, how many images should be in the sequence.
3. 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** palette.

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 **Shaders** and **Lighting** for the turntable images:
 - You can use any shader defined in your project, or **Current** as a short-cut to use your current shader.
 - 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 choose 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 Controls** palette.
4. Set the other details for the turntable(s) to generate. The properties for diagnostic turntables are the same as for [render turntables](#), 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 (%CHANNEL 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:



Sandbox *Try generating some turntables.*

Save, Load, and Delete 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 choose 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 choose 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.

25 MULTIPLE OBJECTS

Mari allows you to load multiple 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, built 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.

Channels and Shaders

Each object in the project has its own set of channels. When you add or remove 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.

You can also match channels between objects. This takes one object as a model and creates a matching set of channels on the destination object.

Similarly, each object has its own shaders. As with channels, you must set the shaders up separately for each object.

Versioning

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.

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.

Object Shadows

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.

Shadows can be toggled on or off in the display by selecting the **Shadows** button in the **Lighting** toolbar.

If **Depth Projections** have been enabled in the **Preferences**, there are two related options that can be set to customize the shadow settings. **Depth Projection Resolution** and **Depth Projection texture Array Size** 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.

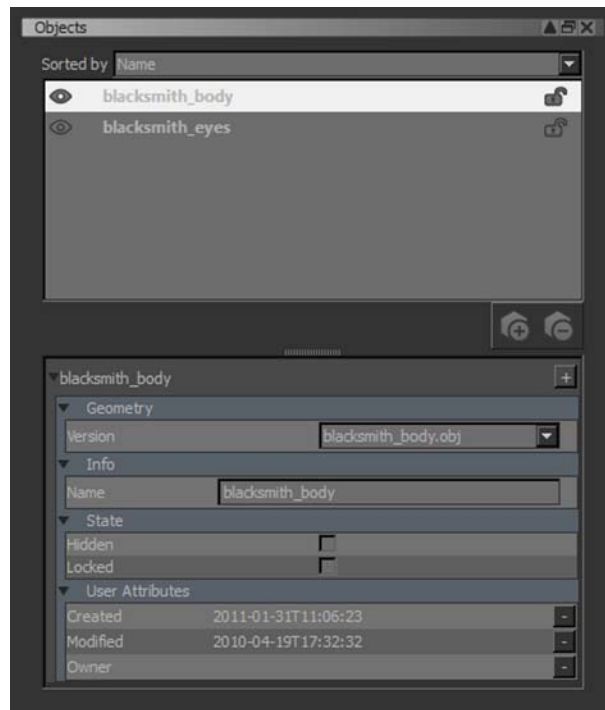
Steps for Working with Multiple Objects:

Working with multiple objects involves:

1. [Adding and Removing Objects](#)
2. [Moving, Viewing, and Locking Objects](#)
3. [Matching Channels Between Objects](#)
4. [Adding and Removing Object Versions](#)
5. [Calculating Ambient Occlusion](#)
6. [Shadow Settings](#)

Adding and Removing Objects

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 object, click the  button.

*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**.

*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 [Create and Delete Individual Channels](#)).*

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. To move it, see [Moving, Viewing, and Locking Objects](#).

5. To remove an existing object, select it in the list and click the  button.

Mari asks you to confirm the deletion.

6. 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

You can move objects around on your Mari canvas. For example, you might have added an object which has appeared on top of an existing object (where the objects appear depends on the configuration of the model file).

1. To move an object, select the **Pan Object**  tool.
2. Click and drag to move the object. It's that simple.



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, or press the apostrophe key (') on the keyboard to reset all objects.

Tip You can also use the **Pan Object** tool to move lights around (see [Configure the Lighting](#)).

Matching Channels Between Objects

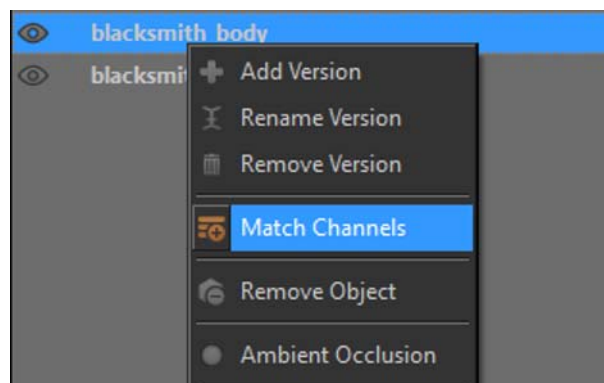
To speed up your workflow when working with multiple objects, Mari lets you match the channels from an object. This creates equivalent channels – with the same names, resolutions, color depths, and so on – on the other objects in the project. You can either match channels object-by-object, or all at once.

So you can set up the channels on one object, and then match them across

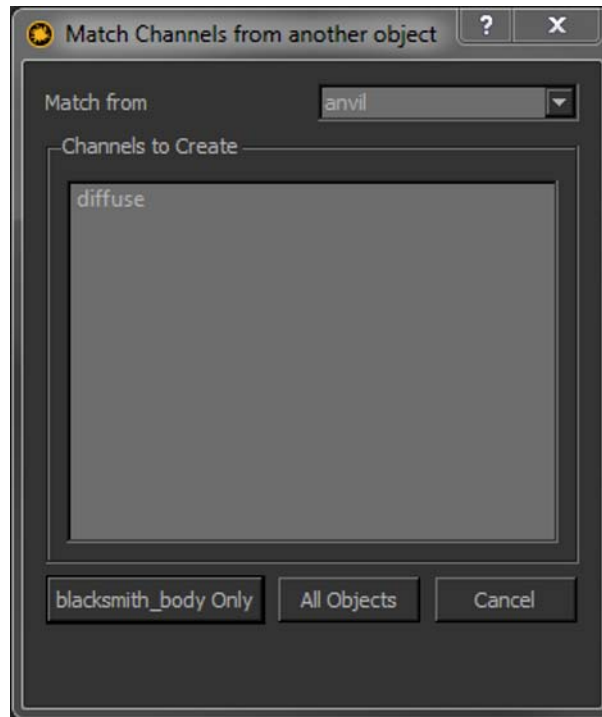
to the other objects in the project.

To match the channels from one object to another:

1. If you only want to match channels onto a **single object**, in the **Objects** palette, right-click on the target object, or
If you want to match channels onto **all objects**, in the **Objects** palette, right-click anywhere except on the object from which you want to match channels.
2. Select **Match Channels** from the dropdown menu.



*The **Match Channels from Another Object** dialog box displays.*



3. Select the source object in the **Match From** dropdown. The list contains all the objects except the “target” object (that is, the object you right-clicked on).
*The **Channels to Create** area shows the channels on the selected object. This is the set of channels that Mari creates on the target object(s).*
4. Click the buttons at the bottom to either match only to the selected object or to all objects in the project.
Mari updates the object(s) with the new channels.

Adding and Removing Object Versions

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** 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.*

Calculating Ambient Occlusion

To calculate the ambient occlusion for an object:

1. Select the object in the **Objects** palette.
2. Either:
 - from the **Objects** menu, select **Ambient Occlusion**, or
 - right-click on the object in the **Objects** palette and select **Ambient Occlusion** from the dropdown menu.

Mari calculates the ambient occlusion for the object(s) in the current project. Depending on the size and complexity of your model, this may be a good time to get a cup of tea.

Shadow Settings

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. If you want to turn off shadows, toggle the **Shadows** button in the **Lighting** toolbar.

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 **Variance Maths** checkbox under **Preferences > Data > Graphics Card**.*



26 LAYERS IN MARI

Layers are a common feature of digital painting programs such as Photoshop® or Gimp. Layers let you separate out various elements of the composition, paint them separately, and then merge them together to create the final result.

Mari does not have a specific layer function. Mari has channels—you can use **transparent channels** and **custom shaders** to get the same result. You can paint on several transparent layers, use a shader to change how the layers interact, how the layers are ordered, and so on. Finally, when you have finished your painting, you can flatten the channels into a single channel for export.

Base Channels

When using layers, you may find it useful to have a base channel. This is the “lowest” layer in the drawing — usually the diffuse color channel.

Flattening Layers

When you have the look you want, you can flatten the layers down to create a single layer. There are two ways to do this:

- bake the shader, or
- merge the channels.

Merging channels is faster, but only works on two channels (optionally using a third channel as a mask). Baking the shader gives you more flexibility — it outputs the visible result of the shader to a channel (existing or new). This lets you build up a complex shader with multiple channel (layer) inputs and then bake it to a single channel. However, baking a shader takes longer than a simple channel merge.

Steps for Working with Layers

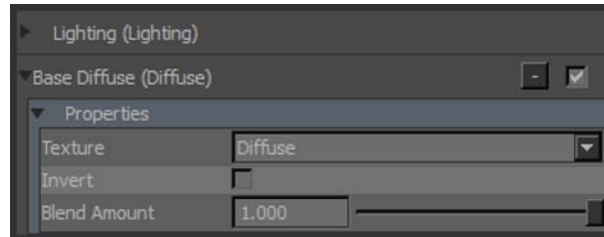
To use layers in your Mari project, follow these steps:

1. [Create the Shader and Channels](#)
2. [Paint on Your Layers](#)
3. [Flatten Layers](#)

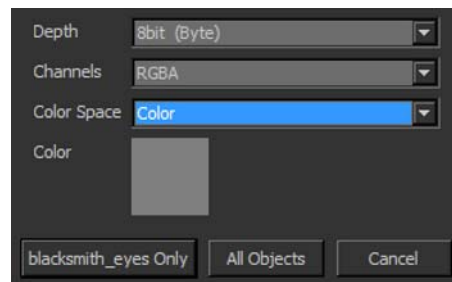
Create the Shader and Channels

First, you need to create a shader to display your layers correctly.

1. Start Mari and open your project.
2. In the **Shaders** palette, create a new shader.
3. Set the default **Base Diffuse (Diffuse)** module of your new shader to take the **Color Texture** from your base shader channel (if you have one).



4. In the **Channels** palette, create a new channel for each layer you want to use. Use the following settings:
 - **Channels** — RGBA (not RGB — you need the Alpha channel).
 - **Color** — 0% alpha (that is, totally transparent).

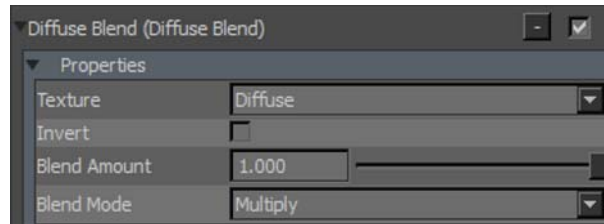


5. Back in the **Shaders** palette, add a module for each of your layers to your shader. Set the modules up as follows:
 - Module type should be **Diffuse Blend**.
 - Set the **Mode** to the blend mode to use for your channel.

Tip *This corresponds to the layer blend modes in Photoshop®. For example, you can set this to **darken**, **multiply**, **burn**, **hard light**, and so on.*

- Set the **Texture** to your new layer channel.

For example:



Tip Remember that the order of channels in the shader affects how the channels “stack” on top of each other in the display.

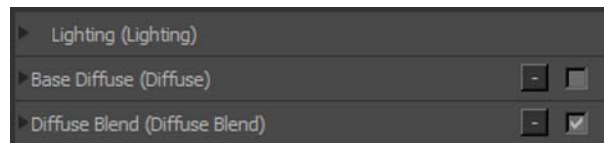
If the channels in your project were imported from a layered **.psd** file, there is a really quick way to create a layered shader. See the section on [Creating Layered Shaders](#) in the *Shaders* chapter for more information.

Paint on Your Layers

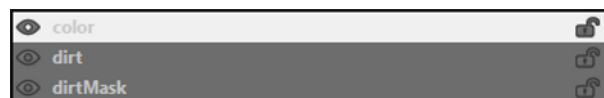
1. In the **Channels** palette, select a channel to paint on.

You may need to turn off the visibility of one or more channels. Do this by either:

- unchecking the display box for the channel's module in your layer shader, or



- in the **Channels** palette, clicking on the eye icon next to the channel.



2. Paint normally, using the standard Mari tools.

Tip When using channels as layers, you may find it particularly helpful to use the **Mask to Alpha** and **Alpha to Mask** functions — to speed up creating and using masks.

See [Channels](#) for details on the Channel functions.

Flatten Layers

You can flatten layers (transparent channels) in Mari by merging channels or baking a shader. See [Merge Channels](#) and [Bake a Shader Component onto a Channel](#).



27 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 textures to show the chipped paint and the displacement from the impact. You can adjust and export textures 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:

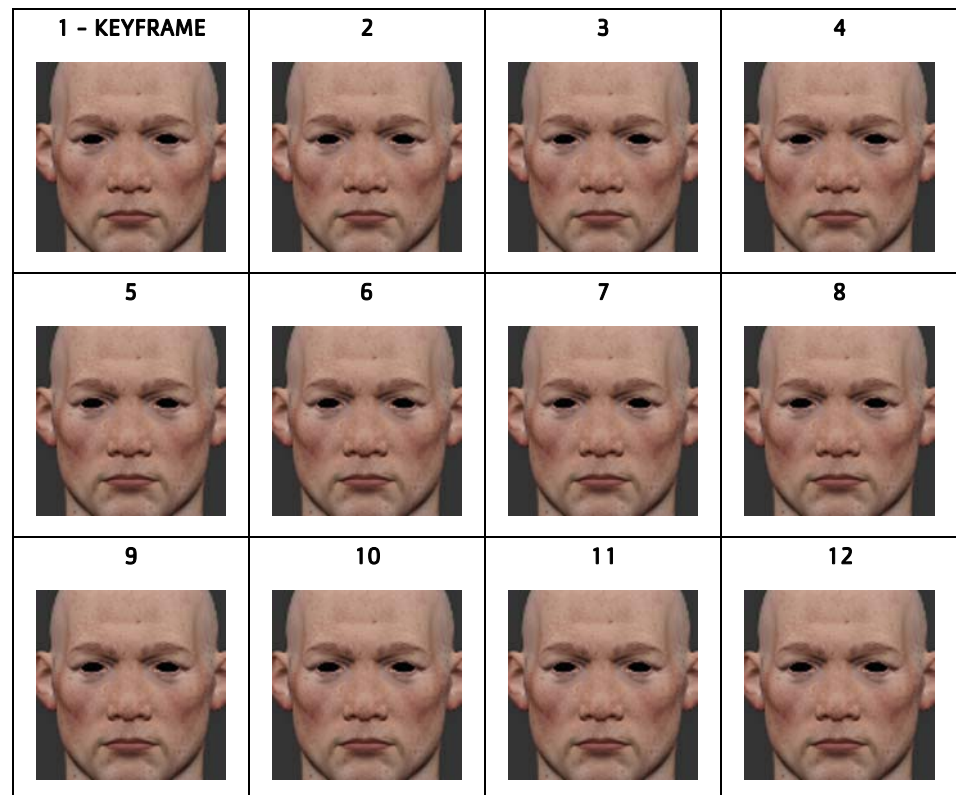
- Any textures you add earlier in the sequence disappear when Mari reaches the keyframe.
- Any textures you add after the keyframe appear once Mari reaches the keyframe.
- When you export your textures, you can export a separate version of each patch for each keyframe.

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.

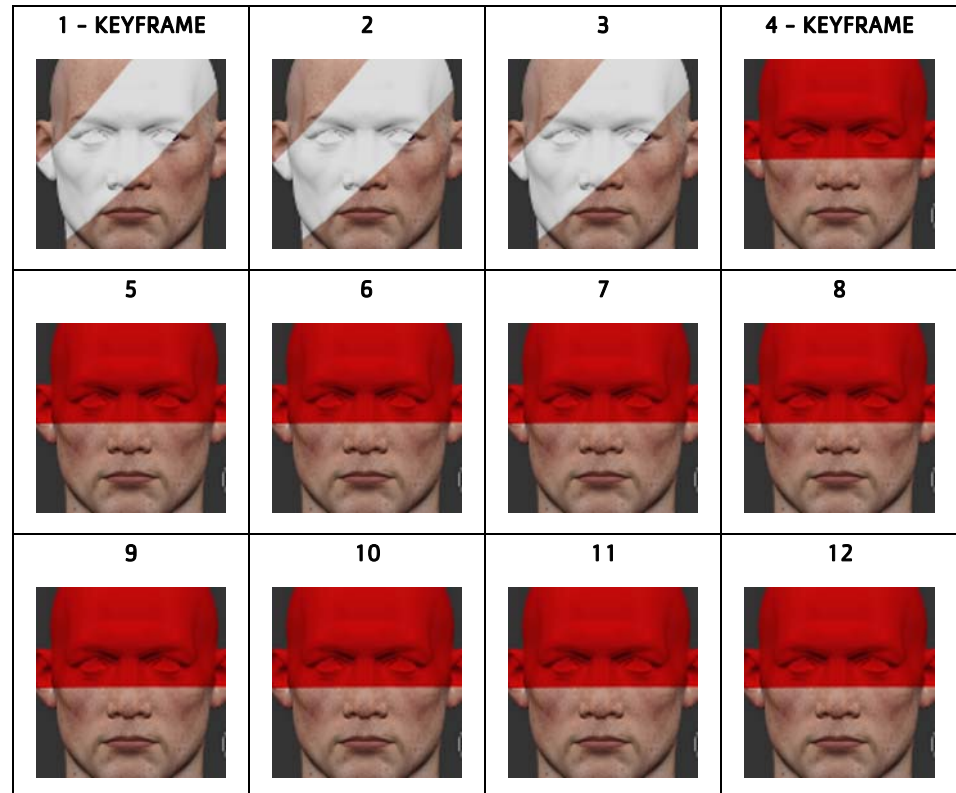
Example

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:



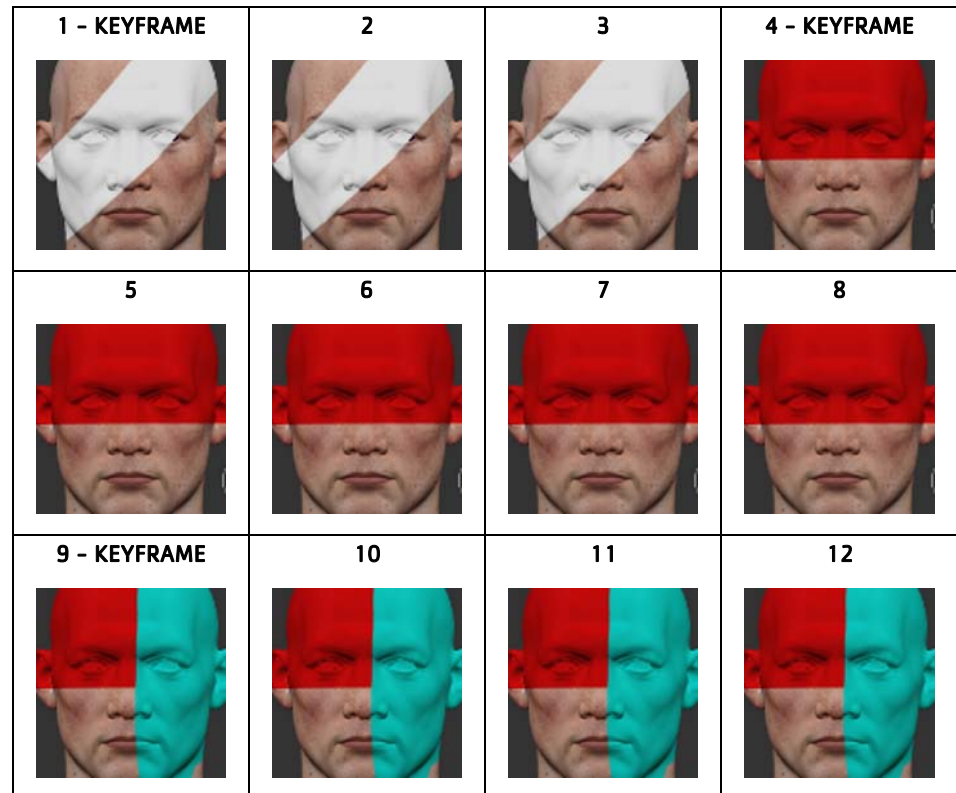
Now we set a keyframe at frame 4, and paint on frame 3 and frame 6. In this case:

- The paint on frame 3 appears from frames 1-3.
- The paint on frame 6 appears from frames 4-12.

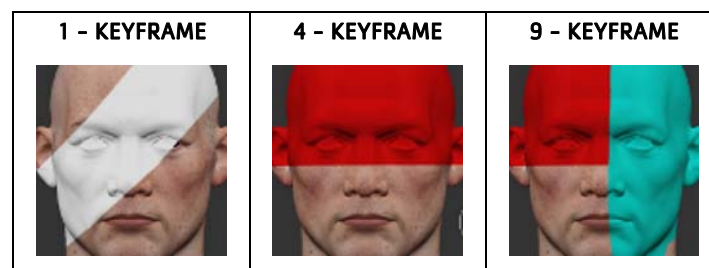


If we add another keyframe at frame 9 and paint on frame 10:

- 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:



When we export the channel, each patch has three versions — one per keyframe.



Steps for Working with Animated Objects

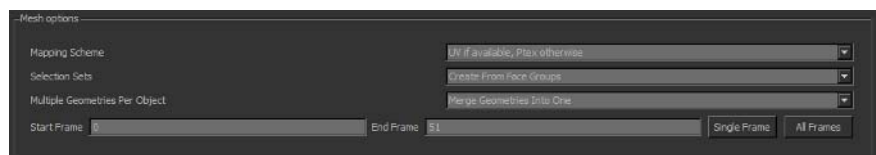
To use animated objects in your Mari project, follow these steps:

- [Loading Animated Objects](#)
- [Playing Animation](#)
- [Painting Animated Textures](#)
- [Exporting Animated Textures](#)

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** palette.



- Set the desired frame range in the boxes at the top of the palette. This lets you play only a subset of the animation.



- To skip to a particular frame, type the frame in the frame box.



- Set the playback mode in the dropdown.
This can be:
 - **Loop** — keep looping through the animation (start to finish).
 - **Bounce** — keep looping back and forth through the animation (start to finish, then backwards back to the start).
 - **Stop** — play the animation once and stop.
- Use the play controls to play through the animation:

Button	Effect	Shortcut key
	Move back one frame.	Ctrl+Left
	Move back to the first frame.	Ctrl+Down
No button	Move to the last frame.	Ctrl+Up
	Play forward.	None.
	Stop playing.	None.
	Move forward one frame	Ctrl+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+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 an 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:

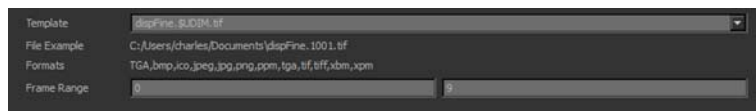
- Any textures you paint within one section disappear when you reach the next keyframe.
- You can paint, then add another keyframe, to "build up" an animated texture (see the [Example](#) above for an explanation).

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** dialog box as normal (from the **Channels** menu, select **Export**).
2. Set the export details as normal, 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 channel called **color**, with keyframes at 9 and 6, then using an export **Template** of **\$FRAME_\$CHANNEL.\$UDIM.tif** gives the following files for patch 0001: 00001_color.0001.tif, 00009_color.0001.tif, and 00016_color.0001.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 Channel** dialog box.



3. Click **All** or **Modified Only** as usual to start the export.
Mari exports the files, creating a file for each patch/keyframe combination.



28 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 **Add Multiple Channels** 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 **Add Multiple Channels** dialog box. These control the filenames that Mari looks for when importing textures into new channels.

You can also configure Mari using [environment variables](#).

About Custom Attributes

You can attach custom attributes to most objects in Mari (such as channels, models, projectors and lights). Rather than information that Mari needs, this is information that can be useful for yourself or other users of the project to know. For example, you can add the assigned owner of the model as a custom attribute attached to the model.

You can choose to add custom attributes for the current session only or save them as part of the project so that they persist when you quit and relaunch Mari.

About Custom Shortcuts

Three types of keyboard shortcuts are available in Mari:

- The **default set** of shortcuts — that come with the application, as listed under **Help > Shortcuts**.
- A **custom set** of shortcuts — defined in a shortcut configuration file. This is a text file, called **Shortcuts.conf**, containing a list of one or more Mari actions and their shortcuts. Mari looks for this file in the following subdirectory of your home directory:
 - **.config/TheFoundry** (on Linux), or
 - **.mari\TheFoundry** (on Windows).

Note *If there is no custom shortcut file, Mari uses the defaults. Configuration file shortcuts override default shortcuts.*

- **GUI shortcut** — defined within the GUI, in the **Manage Keyboard Shortcuts** dialog box, for specific actions or project items. You do this by selecting **Edit > Keyboard Shortcuts**, or right clicking on a project item (channels, patches, shaders, lights, projectors, and things in shelves), or sets of those items, by selecting **Assign Shortcut** from the dropdown menu. *GUI shortcuts override default and configuration file shortcuts.*

So Mari accepts the default keyboard shortcuts for an action... unless that action is also specified in a shortcut configuration file, in which case Mari uses your custom shortcuts... unless you've overridden any of them in the GUI with individual custom shortcuts.

Tip *If you are distributing a custom shortcuts file to multiple users, be aware that any individual custom shortcuts they apply override your shortcuts file. The **Shortcuts.conf** file does not have to list all possible Mari actions. You can set up a shortcut for any Mari action, but you can only assign shortcuts to project items (channels, patches, shaders, lights, projectors, and things in shelves), or sets of those items, from within the Mari GUI (not from within a customization file).*

About Custom Channel Presets

You can add channels in bulk to a Mari project, using the **Channels > Channel Presets** option. This opens a dialog box, **Add Multiple Channels**, where you can select to create a set of channels in a project and optionally to also import existing textures. In the **Add Multiple Channels** dialog box, you can select a **Category**, which represents a preset bundle of channels. For example, if you select the “Vehicles” category, you get a list of standard channels for painting a vehicle.

Mari comes with several default categories, stored in a channel preset file. If you want, you can replace the defaults with your own customized channel preset categories. Mari custom channel presets files are **XML** files with an **.mtd** extension. Mari looks for them by default in the following subdirectory of your home directory:

- **.config/TheFoundry** (on Linux), or
- **.mari.TheFoundry** (on Windows).

You can select a different channel presets directory on the **Channels** tab of the **Mari Preferences** dialog box.

Tip *If you use a customized channel preset file, Mari ignores its default channel presets. Only the channels in your custom preset file are available.*

About Custom Channel Templates

When you're adding channels in bulk through the **Add Multiple Channels** dialog box, you can select to import existing textures into the channels as Mari creates them. Mari looks for existing textures on disk. If it finds these textures, you can import them as the channels are created. The channel template controls the format of the filenames that Mari looks for. For example, one channel template is `$PATH/$SHOT/color$SHOT_color.$UDIM.tif`.

Mari comes with several default channel templates. If you want, you can add your own channel template files. Channel template files are straight text files, with an `.mtt` extension. Mari looks for them in the same directories as channel preset files.

About Environment Variables

Environment variables are named variables used to store a value, such as a specific file path. They are used to dynamically configure Mari's behavior when Mari is launched. For example, Mari uses the information stored in them to define where to place certain files.

You can:

- Set environment variables that Mari recognizes.
- Create your own environment variables that Mari looks for when importing images.
- Use environment variables to override any entry in the **Mari.conf** file. This file contains all user settings that are not project specific.

Steps for Using Custom Attributes

To use custom attributes in Mari, follow these steps:

1. [Add Custom Attributes to Objects](#)
2. [Remove Custom Attributes from Objects](#)

Steps for Customizing Mari

To customize your installation of Mari, follow these steps:

1. [Review the Custom Shortcut File Format](#)
2. [Define Custom Shortcuts](#)
3. [Review the Custom Channel-Preset File Format](#)
4. [Define Custom Channel Presets](#)
5. [Review the Custom Channel Template File Format](#)
6. [Define Custom Channel Templates](#)

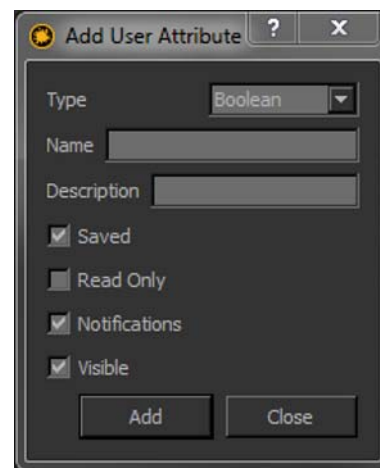
Steps for Setting Environment Variables

You can set all the environment variables mentioned in this chapter in the same manner. To do so, follow these steps:

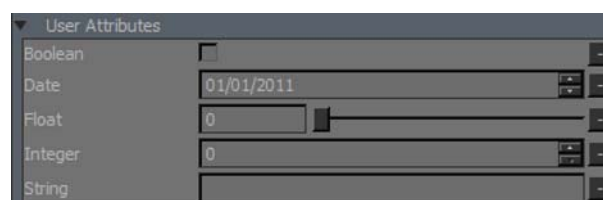
1. [Choose an Environment Variable to Set](#)
2. [Set the Environment Variable](#)
3. [Check That the Variable Now Exists](#)

Add 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.

Remove 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**.

Review the Custom Shortcut File Format

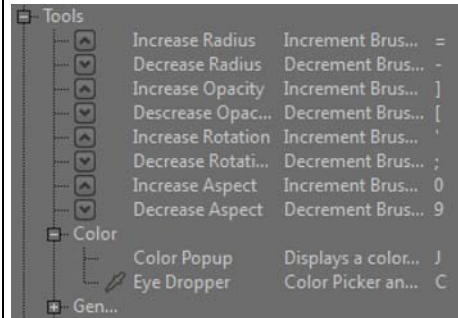
1. Read through the following information about the required format of a Mari custom shortcuts file.
2. Study the example default shortcuts file.

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** 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
	<pre>[Shortcuts] ... Mari\Tools\Increase%20Radius=" " Mari\Tools\Decrease%20Radius=- Mari\Tools\Increase%20Opacity=] Mari\Tools\Decrease%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</pre>

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 **Transform Paint** 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** 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	Ctrl	No
Shift	Shift	No
Alt (not Alt Gr)	Alt	No
Meta (or Windows key)	Meta	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
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 subdirectory of the Mari application directory:

- **Media/Settings** (on Linux), or
- **Bundle\Media\Settings** (on Windows).

We recommend you [create your custom shortcuts](#) by copying this file and editing it, rather than starting from scratch.

```
[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\Default%20Brushes\Paint=P
Mari\Tools\Default%20Brushes\Paint%20Through=U
Mari\Tools\Default%20Brushes\Eraser=E
Mari\Tools\Image\Marquee=M
Mari\Canvas\Bake=B
Mari\Tools\Camera\Rotate=Alt+
Mari\Tools\Camera\Zoom=Alt+Ctrl+
Mari\Tools\Camera\Pan=Alt+Shift+
Mari\Tools\Camera\Spin=Ctrl+R
Mari\Canvas\Camera\Focus=F
Mari\Tools\Color\Eye%20Dropper=C
Mari\Tools\Color\Color%20Popup=j
Mari\Tools\General\Shelf%20Popup=k
Mari\Tools\General\Image%20Popup=l
Mari\Tools\General\Pie%20Selector=F9
Mari\Animation\Start%20/%20Stop%20Animation=Space
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\Canvas\Bake=B
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\Undo=Ctrl+Z
Mari\Actions\Redo=Ctrl+Y
Mari\Actions\Previous%20View="Ctrl+, "
Mari\Actions\Next%20View=Ctrl+.
Mari\Actions\Hide%20Palettes=Home
Mari\Project\Open=Ctrl+O
Mari\Project\Save%20As=Ctrl+Shift+S
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\Palettes\Colors=F6
Mari\Palettes\Tools=F7
Mari\Palettes\Brushes=F8
Mari\Canvas\Camera\View All=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\Project\Project%20Explorer\Select Invert=Ctrl+I
Mari\Project\Project%20Explorer\Select All=Ctrl+A
Mari\Project\Project%20Explorer\Select None=Ctrl+E
```

```
Mari\Project\Project%20Explorer\Toggle%20Hidden=Ctrl+H
Mari\Project\Project%20Explorer\Hide%20Selected=H
Mari\Project\Project%20Explorer\Hide%20Unselected=Shift+H
Mari\Project\Project%20Explorer\Show%20All=Ctrl+Shift+H
Mari\Painter\Paint%20Mask=Q
Mari\Painter\Toggle%20Edge%20Mask=G
Mari\Painter\Toggle%20Show%20Painting=.
Mari\Painter\Show%20Paintable=
Mari\Projection\Toggle%20Mask%20Preview%20Enabled=", "
Mari\Painter\Invert=Shift+I
Mari\Channels\Toggle%20Channels=T
Mari\Channels\Next%20Channel=PgDown
Mari\Channels\Previous%20Channel=PgUp
Mari\Tools\Toggle%20Brushes=N
Mari\Tools\Decrease%20Aspect=9
Mari\Tools\Decrease%20Radius=-
Mari\Tools\Decrease%20Rotation="; "
Mari\Tools\Decrease%20Opacity=[
Mari\Tools\Increase%20Aspect=0
Mari\Tools\Increase%20Opacity=]
Mari\Tools\Increase%20Radius="="
Mari\Tools\Increase%20Rotation='
Mari\Project\Project%20Explorer\Shader%201=Ctrl+1
Mari\Project\Project%20Explorer\Shader%202=Ctrl+2
Mari\Project\Project%20Explorer\Shader%203=Ctrl+3
Mari\Project\Project%20Explorer\Shader%204=Ctrl+4
Mari\Painter\Set%20Colors%20to%20Black%20And%20White=D
```

Define Custom Shortcuts

To create a Mari shortcut customization file:

1. Copy the Mari default shortcuts file, **DefaultShortcuts.conf**, from the following subdirectory of the Mari application directory:
 - **Media/Settings** (on Linux), or

- **Bundle\Media\Settings** (on Windows).
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 subdirectory of your home directory:
 - **.config/TheFoundry** (on Linux), or
 - **.mari\TheFoundry** (on Windows).

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=Alt+Paint** (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.

Review 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.

XML File: Header, Details, Closing Blocks

Channel preset files are **XML** files with a **.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 the user's specific channel presets and one on a network drive for the group's 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
channelName	Name for the channel to create.	My Custom Channel1
scalar	Whether the channel is color or scalar. This is a Boolean; true means a scalar channel, false means color.	false

Setting	Meaning	Example value
width	Width of patches in the channel. Must hatch the heightType.	4096
height	Height of patches in the channel. Must match the widthType.	4096
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 user selects this category in the Channel Presets dialog, this channel appears in the list.	Custom

Tip *If the width and height are greater than a project’s maximum resolution, Mari creates the channel at the project’s maximum resolution.*

Closing block

This must be the last thing in the file.

```
</m_Channels>
</MriSerializableObject>
</Mari>
```

Example: The default channel presets file is included in Mari in the Media/Settings directory (**DefaultChannels.mtd**). We recommend you [create your custom](#)

DefaultChannels.mtd

[channels](#) file by copying this and editing it, rather than starting from scratch.

```
<!DOCTYPE MariML>
<Mari Version="2" >
  <MariSerializableObject Type="MariSerializableObject" >
    <objectName Type="QString" ></objectName>
    <selected Type="bool" >false</selected>
    <m_Channels Size="23" Type="MariChannelInfo"
ContainerType="QVector" >

      <MariChannelInfo Type="MariChannelInfo" >
        <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" >RGB</channels>
        <depth Type="QString" >Byte</depth>
        <color Type="MariColor" >(0.5,0.5,0.5,1)</color>
        <category Type="QString" ></category>
      </MariChannelInfo>

      <MariChannelInfo Type="MariChannelInfo" >
        <objectName Type="QString" ></objectName>
        <selected Type="bool" >false</selected>
        <channelName Type="QString" >specRough</channelName>
        <fileFormat Type="QString" ></fileFormat>
        <path Type="QString" ></path>
        <scalar Type="bool" >true</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" ></category>
</MriChannelInfo>

<MriChannelInfo Type="MriChannelInfo" >
  <objectName Type="QString" ></objectName>
  <selected Type="bool" >>false</selected>
  <channelName Type="QString" >dispFine</channelName>
  <fileFormat Type="QString" ></fileFormat>
  <path Type="QString" ></path>
  <scalar Type="bool" >>true</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" ></category>
</MriChannelInfo>

<MriChannelInfo Type="MriChannelInfo" >
  <objectName Type="QString" ></objectName>
  <selected Type="bool" >>false</selected>
  <channelName Type="QString" >dispBroad</channelName>
  <fileFormat Type="QString" ></fileFormat>
  <path Type="QString" ></path>
  <scalar Type="bool" >>true</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" ></category>
</MriChannelInfo>

<MriChannelInfo Type="MriChannelInfo" >
  <objectName Type="QString" ></objectName>
  <selected Type="bool" >>false</selected>
  <channelName Type="QString" >dirt</channelName>
  <fileFormat Type="QString" ></fileFormat>
  <path Type="QString" ></path>
  <scalar Type="bool" >>false</scalar>
  <width Type="int" >2048</width>
  <height Type="int" >2048</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" ></category>
</MriChannelInfo>

<MriChannelInfo Type="MriChannelInfo" >
  <objectName Type="QString" ></objectName>
  <selected Type="bool" >>false</selected>
  <channelName Type="QString" >dirtMask</channelName>
  <fileFormat Type="QString" ></fileFormat>
  <path Type="QString" ></path>
  <scalar Type="bool" >>true</scalar>
  <width Type="int" >4096</width>
  <height Type="int" >4096</height>
  <channels Type="QString" >RGB</channels>
  <depth Type="QString" >Byte</depth>
```

```
<color Type="MriColor" >(1,1,1,1)</color>
<category Type="QString" ></category>
</MriChannelInfo>

</m_Channels>
</MriSerializableObject>
</Mari>
```

Define Custom Channel Presets

To create a Mari custom channel presets file:

1. Copy the Mari default channel presets file, **DefaultChannels.mtd**, from the following subdirectory of the Mari application directory:
 - **Media/Settings** (on Linux), or
 - **Bundle\Media\Settings** (on Windows).
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 subdirectory of your home directory:
 - **.config/TheFoundry** (on Linux), or
 - **.mari\TheFoundry** (on Windows).

This is also the directory specified on the **Channels** tab of the **Mari Preferences** dialog box.

Review 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 Add Multiple Channels 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 Add Multiple Channels 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"
```

Define Custom Channel Templates

To create a Mari custom channel templates file:

1. Copy the Mari default channel templates file, **DefaultTemplates.mtt**, from the following subdirectory of the Mari application directory:
 - **Media/Settings** (on Linux), or
 - **Bundle\Media\Settings** (on Windows).
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 subdirectory of your home directory:
 - **.config/TheFoundry** (on Linux), or
 - **.mari\TheFoundry** (on Windows).

This is also the directory specified on the **Channels** tab of the **Mari Preferences** dialog box.

Choose an Environment Variable to Set

1. Read through the following information on Mari environment variables and how you can use them.
2. Choose an environment variable you want to set.

Environment Variables That Mari Recognizes

This table lists the environment variables Mari recognizes.

Environment variable	Description	Example value
MARI_CACHE	The location(s) of the Mari cache files. This can be just one directory or several (to make caching faster). You should separate multiple path entries by a : (colon) on Linux, or a ; (semi-colon) on Windows. Cache 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 cache 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 subdirectory of your home directory on Linux, or the .mari\TheFoundry subdirectory on Windows). be cautious of changing this. The set of cache 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]/Mari-Cache On Windows: C:\[MyPath]\MariCache
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, or • MariLog.txt in the C:\Users\[Username]\Documents\Mari\Logs directory on Windows.	On Linux: /user/Mari/MyLog.txt On Windows: C:\TEMP\MyLog.txt

Environment variable	Description	Example value
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. 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.	30
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, or • MariLog.txt in the C:\Users\[Username]\Documents\Mari\Logs directory on Windows.
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: /SharedDisk/Mari
MARI_DEFAULT_IMAGE_PATH	The default path to load and save reference images.	
MARI_DEFAULT_GEOMETRY_PATH	The default path to load geometry from.	
MARI_DEFAULT_ARCHIVE_PATH	The default path to load and save project archives.	
MARI_DEFAULT_SHELF_PATH	The default path to load and save shelf files.	
MARI_DEFAULT_EXPORT_PATH	The default path to export textures to.	

Environment variable	Description	Example value
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_CAMERA_PATH	The default path to load and save cameras and projectors.	
MARI_USER_PATH	Root of the default user path.	~/Mari on Linux, or C:\Users\username\Documents\Mari on Windows.
MARI_SCRIPT_PATH	A list of paths to run scripts from.	~/Mari/Scripts on Linux, or C:\Users\username\Documents\Mari\Scripts on Windows

Image Importing

You can set environment variables that Mari looks for 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, or **%** on Windows. Mari replaces that part of the path with the value of the environment variable. For example:

On Linux, 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 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 **Mari.conf** (on Linux) or **Mari.ini** (on Windows). This file contains all user settings that are not project specific. By default, you can find it in the following subdirectory of your home directory:

- **.config/TheFoundry** (on Linux), or
- **.mari\TheFoundry** (on Windows).

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.

Set the Environment Variable

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**.

Check That the Variable Now Exists

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.

29 APPENDIX A: SHORTCUTS

Keyboard Shortcuts

Keyboard Shortcuts			
Painting		Selection	
Bake	B	Copy Patch	Ctrl+C
Bake and Clear	Ctrl+Shift+B	Paste Patch	Ctrl+V
Undo	Ctrl+Z	Select All	Ctrl+A
Redo	Ctrl+Y	Select None	Ctrl+E
Hide/Show Painting	.	Hide	H
Hide/Show Edge mask	,	Hide Unselected	Shift+H
Edge Mask On/Off	G	Show All	Ctrl+Shift+H
Invert Edge Mask	Shift+I	Invert	Ctrl+I
Disable/Enable Masking	Ctrl+M	Toggle Hidden	Ctrl+H
Color Picker Menu	J	Reset Object	'
Shelf Menu	K	Clear Marquee Selection	Ctrl+D
Image Manager Menu	L		
Selection 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		

Keyboard Shortcuts			
Tools		Brush	
Select	S	Adjust Radius	R
Transform	M	Rotation	W

Keyboard Shortcuts			
Gestures	\	Squish	Q
Hide/Show Widgets	/	Opacity	O
Paint	P	Increase/Decrease Radius	= / -
Paint Through	U	Increase/Decrease Aspect	O / 9
Eraser	E	Toggle Last Brush	N
Color Picker	C		
Vector Inspector	V		
Toggle Vector Inspector	Shift+I		

Keyboard Shortcuts			
Camera		View	
Rotate	Alt	Go to Next Tab	Ctrl+Tab
Zoom	Alt+Ctrl+Left button	Go to Previous Tab	Ctrl+Shift+Back Tab
Pan	Alt+Shift	Full Screen	Ctrl+Shift+F
Spin	Ctrl+R	Increase/Decrease Opacity	] / [
Focus	F	Increase / Decrease Rotation	' / ;
Camera View Presets	1,2,3,4,5,6	Previous View	Ctrl+ ,
Focus on Selection	A	Next View	Ctrl+ .
View All	Shift+A	Show Console	Ctrl+Alt+P
Load Camera	F10		
Ortho Camera	F11		
UV Camera	F12		

Keyboard Shortcuts			
Animation		Project	
Move Forward One Frame	Ctrl+Right	New	Ctrl+N
Move Backward One frame	Ctrl+Left	Save	Ctrl+S
Fast Forward	Ctrl+Up	Close	Ctrl+W
Rewind	Ctrl+Down		

Keyboard Shortcuts			
Add keyframe	Ctrl+K		
Scrub Animation	Shift+?		

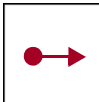
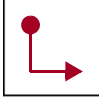
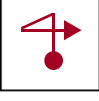
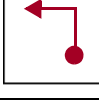
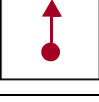
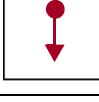
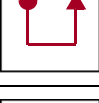
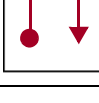
Keyboard Shortcuts			
Channels		General	
Next	PgDown	Quit	Ctrl+Q
Previous	PgUp	Hide/Show Palettes	Home
Toggle	T	Hide/Show Tools Palette	F7
Halve Size	Ctrl+-	Hide/Show Brush Editor Palette	F8
Double Size	Ctrl+ +	Toggle Wireframe View	Shift+W
Channel Menu	I	Take Screenshot	Ctrl+Shift+Print
Bake on New Channel	Ctrl+B		

Keyboard Shortcuts			
Color		Lighting	
Swap	X	Flat	F1
Set to Black & White	D	Basic	F2
Increase Pick Size	]	Full	F3
Decrease Pick Size	[		

Keyboard Shortcuts			
UV View presets		Navigation	
Reset Model to Default	1	Orbit	Alt+Left button
Rotate 90 degrees counter-clockwise	2	Pan	Alt+Shift+Left button or Alt+Middle button

Keyboard Shortcuts			
Rotate 180 degrees counter-clockwise	3	Roll	Ctrl+R+Left button
Rotate 270 degrees counter-clockwise	4		

Mouse Gestures

Mouse gestures	
Hold down \ and draw...	
 Undo	 Redo
 Clear Painting	 Bake
 Previous Channel	 Next Channel
 Toggle Last 2 Channels	 Paint Tool
 Paint Through Tool	 Selection Tool
 Erase Tool	 Show Palettes
 Increase Brush Radius	 Decrease Brush Radius
 Hide Unselected	 Hide Selected
 Show All	 View All
 Toggle Brush Tip	



30 APPENDIX B: FREQUENTLY ASKED QUESTIONS

Frequently Asked Questions

This appendix lists some of the most common questions about Mari. If your question isn't answered here, review the documentation and the release notes or contact The Foundry 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 **Projection** palette.
- A mask is preventing paint from appearing. Check if any of the masks are enabled in the **Projection** palette.
- **Project On** has been set to **SelectedOnly** in the **Projection** 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** or **Unlock All**.
- The paint blending mode (**Painting Mode** in the **Projection** 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 **Brush Editor** palette.
- **Brush Flow** has been set to 0 in the **Brush Editor** 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 **Projection** 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 Virtual Texture Size can affect Mari's performance.

Increasing the **Buffer Size** or the **Virtual Texture Size**, 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 **Virtual Texture Size** in the **Mari Preferences** dialog under **Data > Graphics Card**.

Displacement preview performance seems slow.

When using the **Displacement** shader module, the **Virtual Texture Type** under **Edit > Preferences > Data > Graphics Card**, 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 **.** 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, Alt, or meta (Windows) shortcut key doesn't do what it should. Instead, it does something else, like moves the entire Mari window.

This is usually not a problem in Mari but caused 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 at least choose 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 be the **meta** key instead. This way, the Mari shortcuts that don't work are limited to those that require the **meta** key.

To block the global shortcut on a per-application basis (that is, for Mari only) on KDE 4, for example, 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 choose which key is used as the global shortcut, again 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 **meta** key).
This allows you to use the other two keys (in this case, **Ctrl** 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.

1 APPENDIX C: THIRD PARTY LICENSES

Third Party Licenses

This appendix lists third party libraries used in Mari, along with their licenses.

Library	Description	Licence
argparse	Parser for command-line options, arguments and sub-commands	Copyright 2008 Larry Gritz and the other authors and contributors. All Rights Reserved. Based on BSD-licensed software Copyright 2004 NVIDIA Corp. 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 software's owners 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. (This is the Modified BSD License)
Autodesk FBX	File Format Support	This software contains Autodesk® FBX® code developed by Autodesk, Inc. Copyright 2008 Autodesk, Inc. All rights, reserved. Such code is provided "as is" 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	Description	Licence
Curl 7.21.1	URL transfer library	Copyright (c) 1996 - 2010, Daniel Stenberg, <daniel@haxx.se>. 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.
EdenCGL	Object file loader	Copyright (c) 2001-2008, Philip Lamb, phil@eden.net.nz. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Expat	XML parser	Copyright © 1998, 1999, 2000 Thai Open Source Software Center Ltd and Clark Cooper Copyright © 2001, 2002, 2003, 2004, 2005, 2006 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	Description	Licence
FBO	Frame buffer object library	Copyright (c) 2005, Aaron Leohn (leohn@cs.ucdavis.edu) Adam Moerschell (atmoerschell@ucdavis.edu) All rights reserved. This software is licensed under the BSD open-source license. See http://www.open-source.org/licenses/bsd-license.php for more detail. 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 University of California, Davis nor the names of the 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.
FreeType	Font support	Portions of this software are copyright © 2008 The FreeType Project (www.freetype.org). All rights reserved.

Library	Description	Licence
GLEW	OpenGL support	The OpenGL Extension Wrangler Library Copyright © 2002-2008, Milan Ikits <milan.ikits@ieee.org> Copyright © 2002-2008, Marcelo E. Magallon <mmagallo@debian.org> Copyright © 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.
GLib 2.24.0	Core application building blocks for applications written in C; used under LGPL v2	GObject - GLib Type, Object, Parameter and Signal Library Copyright (C) 2000-2001 Red Hat, Inc. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA. To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.

Library	Description	Licence
GLSL Cellular Noise	2D and 3D Cellular Noise	Copyright (c) 2011 by Stefan Gustavson <stefan.gustavson@liu.se> 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.
GLSL Perlin Noise	Perlin noise library	2D, 3D and 4D Perlin noise, classic and simplex, in a GLSL fragment shader. Classic noise is implemented by the functions: <pre>float noise(vec2 P) float noise(vec3 P) float noise(vec4 P)</pre> Simplex noise is implemented by the functions: <pre>float snoise(vec2 P) float snoise(vec3 P) float snoise(vec4 P)</pre> Author: Stefan Gustavson ITN-LiTH (stegu@itn.liu.se) 2004-12-05 You may use, modify and redistribute this code free of charge, provided that my name and this notice appears intact.
GTS	Triangulated surface library	This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.
IJG JPEG	File format support	This software is based in part on the work of the Independent JPEG Group.

Library	Description	Licence
libnoise	Perlin noise library	Copyright (C) 2003, 2004 Jason Bevins This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License (COPYING.txt) for more details. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA The developer's email is jlbezigvins@gmzigail.com (for great email, take off every 'zig'.) To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.
Little CMS	Color Management System	Copyright (c) 1998-2010 Marti Maria Saguer 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	Description	Licence
MD5	MD5 Algorithm	Copyright (C) 1999, 2002 Aladdin Enterprises. All rights reserved. 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: 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. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software. This notice may not be removed or altered from any source distribution. L. Peter Deutsch ghost@aladdin.com
Mouse Gesture Package	Mouse Gesture Recognition Package	Copyright (C) 2006 Johan Thelin <e8johan@gmail.com> 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 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	Description	Licence
MSIntTypes r20	C++ Standards compliant stdint.h and inttypes.h for Microsoft Visual Studio	ISO C9x compliant stdint.h for Microsoft Visual Studio Based on ISO/IEC 9899:TC2 Committee draft (May 6, 2005) WG14/N1124 Copyright (c) 2006 Alexander Chemeris 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 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 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 (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	Description	Licence
OpenColorIO	OpenColorIO library	Copyright (c) 2003-2010 Sony Pictures Imageworks Inc., et al. 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 Sony Pictures Imageworks 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	Description	Licence
OpenCV 2.0.0	Open Source Computer Vision library	IMPORTANT: READ BEFORE DOWNLOADING, COPYING, INSTALLING OR USING. By downloading, copying, installing or using the software you agree to this license. If you do not agree to this license, do not download, install, copy or use the software. License Agreement For Open Source Computer Vision Library Copyright (C) 2000-2008, Intel Corporation, all rights reserved. Copyright (C) 2008-2009, Willow Garage Inc., all rights reserved. Third party copyrights are property of their respective owners. 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 copyright holders may not 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 Intel Corporation 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	Description	Licence
OpenEXR	File format support	Copyright © 2002, Industrial Light & Magic, a division of Lucas Digital Ltd. LLC 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 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	Description	Licence
OpenImageIO 0.9	A library for reading and writing images	OpenImageIO and all code, documentation, and other materials contained therein are: Copyright 2008 Larry Gritz and the other authors and contributors. 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 the software's owners 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. (This is the Modified BSD License.)

Library	Description	Licence
OpenSSL 1.0.0a	A toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library	The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org. OpenSSL License Copyright (c) 1998-2008 The OpenSSL 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 also provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)" THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED 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 OpenSSL PROJECT OR ITS 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. This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Library	Description	Licence
		Original SSLeay License Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. 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 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. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related. 4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)" THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "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 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	Description	Licence
		The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]
PSD	PSD file support. Based on Disney software.	PSD file support includes Disney technology licensed from Walt Disney Animation Studios.
Ptex	Ptex library	Copyright 2009 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 OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
Python 2.6.5	Source code language	Copyright (c) 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010 Python Software Foundation; All Rights Reserved

Library	Description	Licence
PythonQt	Python/Qt bindings	Copyright (C) 2010 MeVis Medical Solutions AG All Rights Reserved. This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. Further, this software is distributed without any warranty that it is free of the rightful claim of any third person regarding infringement or the like. Any license provided herein, whether implied or otherwise, applies only to this software file. Patent licenses, if any, provided herein do not apply to combinations of this program with other software, or any other product whatsoever. You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Contact information: MeVis Medical Solutions AG, Universitaetsallee 29, 28359 Bremen, Germany or: http://www.mewis.de To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.
Pystring	Pystring library	Copyright (c) 2008-2010, 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 the organization Sony Pictures Imageworks 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	Description	Licence
Qt	Application and UI framework	The Qt GUI Toolkit is Copyright (C) 2009 Nokia Corporation and/or its subsidiary(-ies). Contact: Qt Software Information (qt-info@nokia.com) Qt is available under the LGPL. Nokia Qt LGPL Exception version 1.0: As a special exception to the GNU Lesser General Public License version 2.1, the object code form of a "work that uses the Library" may incorporate material from a header file that is part of the Library. You may distribute such object code under terms of your choice, provided that the incorporated material (i) does not exceed more than 5% of the total size of the Library; and (ii) is limited to numerical parameters, data structure layouts, accessors, macros, inline functions and templates. To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.

Library	Description	Licence
QtSingleApplica- tion	A QtSolutions archive component which provides support for applications that can be only started once per user; used under LGPL	Copyright (c) 2009 Nokia Corporation and/or its subsidiary(-ies). All rights reserved. Contact: Nokia Corporation (qt-info@nokia.com) This file is part of a Qt Solutions component. Commercial Usage Licensees holding valid Qt Commercial licenses may use this file in accordance with the Qt Solutions Commercial License Agreement provided with the Software or, alternatively, in accordance with the terms contained in a written agreement between you and Nokia. GNU Lesser General Public License Usage Alternatively, this file may be used under the terms of the GNU Lesser General Public License version 2.1 as published by the Free Software Foundation and appearing in the file LICENSE.LGPL included in the packaging of this file. Please review the following information to ensure the GNU Lesser General Public License version 2.1 requirements will be met: http://www.gnu.org/licenses/old-licenses/lgpl-2.1.html. In addition, as a special exception, Nokia gives you certain additional rights. These rights are described in the Nokia Qt LGPL Exception version 1.1, included in the file LGPL_EXCEPTION.txt in this package. GNU General Public License Usage Alternatively, this file may be used under the terms of the GNU General Public License version 3.0 as published by the Free Software Foundation and appearing in the file LICENSE.GPL included in the packaging of this file. Please review the following information to ensure the GNU General Public License version 3.0 requirements will be met: http://www.gnu.org/copyleft/gpl.html. Please note Third Party Software included with Qt Solutions may impose additional restrictions and it is the user's responsibility to ensure that they have met the licensing requirements of the GPL, LGPL, or Qt Solutions Commercial license and the relevant license of the Third Party Software they are using. If you are unsure which license is appropriate for your use, please contact Nokia at qt-info@nokia.com. To see the GNU Lesser General Public License, go to http://www.gnu.org/licenses/ or /Media/Help/CopyrightNotices in the Mari application directory.
QuaZip	Qt zip wrapper	Copyright (C) 2005 Sergey A. Tachenov Released under the GNU Lesser General Public License.
Qwt	Qt widgets for technical applications	Mari is based in part on the work of the Qwt project (http://qwt.sf.net).

Library	Description	Licence
SGI virtual trackball	Virtual trackball for interacting with 3D objects	(c) Copyright 1993, 1994, Silicon Graphics, Inc. ALL RIGHTS RESERVED Permission to use, copy, modify, and distribute this software for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice appear in supporting documentation, and that the name of Silicon Graphics, Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. THE MATERIAL EMBODIED ON THIS SOFTWARE IS PROVIDED TO YOU "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 SILICON GRAPHICS, INC. BE LIABLE TO YOU OR ANYONE ELSE FOR ANY DIRECT, SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER, INCLUDING WITHOUT LIMITATION, LOSS OF PROFIT, LOSS OF USE, SAVINGS OR REVENUE, OR THE CLAIMS OF THIRD PARTIES, WHETHER OR NOT SILICON GRAPHICS, INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE POSSESSION, USE OR PERFORMANCE OF THIS SOFTWARE. US Government Users Restricted Rights Use, duplication, or disclosure by the Government is subject to restrictions set forth in FAR 52.227.19(c)(2) or subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and/or in similar or successor clauses in the FAR or the DOD or NASA FAR Supplement. Unpublished-- rights reserved under the copyright laws of the United States. Contractor/manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94039-7311. OpenGL(TM) is a trademark of Silicon Graphics, Inc. ===== Trackball code: Implementation of a virtual trackball. Implemented by Gavin Bell, lots of ideas from Thant Tessman and the August '88 issue of Siggraph's "Computer Graphics," pp. 121-129. Vector manip code: Original code from: David M. Ciemiewicz, Mark Grossman, Henry Moreton, and Paul Haeberli Much mucking with by: Gavin Bell

Library	Description	Licence
Skein-Hash	Hashing library	Source code author: Doug Whiting, 2008. This algorithm and source code is released to the public domain. Copyright (c) 2003, Dr Brian Gladman, Worcester, UK. All rights reserved. LICENSE TERMS The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that: 1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer; 2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials; 3. the copyright holder's name is not used to endorse products built using this software without specific written permission. ALTERNATIVELY, provided that this notice is retained in full, this product may be distributed under the terms of the GNU General Public License (GPL), in which case the provisions of the GPL apply INSTEAD OF those given above. DISCLAIMER This software is provided 'as is' with no explicit or implied warranties in respect of its properties, including, but not limited to, correctness and/or fitness for purpose.
TinyXML	TinyXML library	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.

Library	Description	Licence
Tri Stripper	Geometry optimisation library	Copyright (C) 2004 Tanguy Fautrv@. 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. Tanguy Fautrv@ softdev@telenet.be
yaml-cpp	YAML parser and emitter for C++	Copyright (c) 2008 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.

2 APPENDIX D: END USER LICENSE AGREEMENT

End User Licencing Agreement (EULA)

IMPORTANT: BY INSTALLING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT DO NOT INSTALL, COPY OR USE THE SOFTWARE.

This END USER LICENSE AGREEMENT (this "Agreement") is made by and between The Foundry Visionmongers Ltd., a company registered in England and Wales, ("The Foundry"), and you, as either an individual or a single entity ("Licensee").

In consideration of the mutual covenants contained herein and for other good and valuable consideration (the receipt and sufficiency of which is acknowledged by each party hereto) the parties agree as follows:

SECTION 1. GRANT OF LICENSE.

Subject to the limitations of Section 2, The Foundry hereby grants to Licensee a limited, non-transferable and non-exclusive license to install and use a machine readable, object code version of this software program (the "Software") and accompanying user guide and other documentation (collectively, the "Documentation") solely for Licensee's own internal business purposes (collectively, the "License"); provided, however, Licensee's right to install and use the Software and the Documentation is limited to those rights expressly set out in this Agreement.

SECTION 2. RESTRICTIONS ON USE.

Licensee is authorized to use the Software in machine readable, object code form only, and Licensee shall not: (a) assign, sublicense, sell, distribute, transfer, pledge, lease, rent, share or export the Software, the Documentation or Licensee's rights hereunder; (b) alter or circumvent the copy protection mechanisms in the Software or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software; (c) modify, adapt, translate or create derivative works based on the Software or Documentation; (d) use, or allow the use of, the Software or Documentation on any project other than a project produced by Licensee (an "Authorized Project"); (e) 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 or Documentation; (f) copy or install the Software or Documentation other than as expressly provided for herein; or (g) 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 The Foundry or any third party with intellectual property rights in the Software (each, a "Third Party Licensor"). Furthermore, for purposes of this Section 2, the term "Software" shall include any derivatives of the Software.

Licensee shall install and use only a single copy of the Software on one computer, unless the Software is installed in a "floating license" environment, in which case Licensee may install the Software on more than

one computer; provided, however, Licensee shall not at any one time use more copies of the Software than the total number of valid Software licenses purchased by Licensee.

Please note that in order to guard against unlicensed use of the Software a licence key is required to access and enable the Software. The issuing of replacement or substituted licence keys if the Software is moved from one computer to another is subject to and strictly in accordance with The Foundry's Licence Transfer Policy, which is available on The Foundry's website and which requires a fee to be paid in certain circumstances. The Foundry may from time to time and at its sole discretion vary the terms and conditions of the Licence Transfer Policy.

Furthermore, if the Software can be licensed on an "interactive" or "non-interactive" basis, licensee shall be authorized to use a non-interactive version of the Software for rendering purposes only (i.e., on a CPU, without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Licensee shall be authorized to use an interactive version of the Software for both interactive and non-interactive rendering purposes, if available.

If Licensee has purchased the Software on the discount terms offered by The Foundry's Educational Policy published on its website ("the Educational Policy"), Licensee warrants and represents to The Foundry as a condition of this Agreement that: (a) (if Licensee is an individual) he or she is a part-time or full-time student at the time of purchase and will not use the Software for commercial, professional or for-profit purposes; (b) (if the Licensee is not an individual) it is an organisation that will use it only for the purpose of training and instruction, and for no other purpose (c) Licensee will at all times comply with the Educational Policy (as such policy may be amended from time to time).

Finally, if the Software is a "Personal Learning Edition," ("PLE") Licensee may use it only for the purpose of personal or internal training and instruction, and for no other purpose. PLE versions of the Software may not be used for commercial, professional or for-profit purposes including, for the avoidance of doubt, the purpose of providing training or instruction to third parties.

SECTION 3. SOURCE CODE.

Notwithstanding that Section 1 defines "Software" as an object code version and that Section 2 provides that Licensee may use the Software in object code form only, The Foundry may also agree to license to Licensee (including by way of upgrades, updates or enhancements) source code or elements of the source code of the Software the intellectual property rights in which belong either to The Foundry or to a Third Party Licensor ("Source Code"). If The Foundry does so Licensee shall be licensed to use the Source Code as Software on the terms of this Agreement and: (a) notwithstanding Section 2 (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 its own internal business purposes and in all respects in accordance with this Agreement; (b) Licensee shall in respect of the Source Code comply strictly with all other restrictions applying to its use of the Software under this Agreement as well as any other restriction or instruction that is communicated to it by The Foundry at any time during this Agreement (whether imposed or requested by The Foundry or by any Third Party Licensor); (c) notwithstanding any other term of this Agreement The Foundry gives no warranty whatsoever in respect of the Source Code, which is licensed on an "as is" basis, or in respect of any modification of the Source Code made by Licensee ("Modification"); (d)

notwithstanding any other term of this Agreement The Foundry shall have no obligation to provide support, maintenance, upgrades or updates of or in respect of the Source Code or of any Modification; and (e) Licensee shall indemnify The Foundry against all liabilities and expenses (including reasonable legal costs) incurred by The Foundry in relation to any claim asserting that any Modification infringes the intellectual property rights of any third party.

SECTION 4. BACK-UP COPY.

Notwithstanding Section 2, Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location 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 Agreement.

SECTION 5. OWNERSHIP.

Licensee acknowledges that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation and all intellectual property rights and other proprietary rights relating thereto are and shall remain the sole property of The 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 The Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this Agreement, 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 The Foundry hereunder. Nothing in this 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 The Foundry or any Third Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of The Foundry or any Third Party Licensor in Licensee's promotion or publicity without The Foundry's express written approval.

SECTION 6. LICENSE FEE.

Licensee understands that the benefits granted to Licensee hereunder are contingent upon Licensee's payment in full of the license fee payable in connection herewith (the "License Fee").

SECTION 7. UPGRADES/ENHANCEMENTS.

The Licensee's access to support, upgrades and updates is subject to the terms and conditions of the "Annual Upgrade and Support Programme" available on The Foundry's website. The Foundry may from time to time and at its sole discretion vary the terms and conditions of the Annual Upgrade and Support Programme.

SECTION 8. TAXES AND DUTIES.

Licensee agrees to pay, and indemnify The 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 this Agreement.

SECTION 9. LIMITED WARRANTY.

The Foundry warrants that, for a period of ninety (90) days after delivery of the Software: (a) the machine readable electronic files constituting the Software and Documentation shall be free from errors that may arise from the electronic file transfer from The Foundry and/or its authorized reseller to Licensee; and (b) to the best of The Foundry's knowledge, Licensee's use of the Software in accordance with the Documentation will not, in and of itself, infringe any third party's copyright, patent or other intellectual property rights. Except as warranted, the Software and Documentation is being provided "as is." THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND The Foundry DISCLAIMS ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, REGARDLESS OF WHETHER The Foundry KNOWS OR HAS REASON TO KNOW OF LICENSEE'S PARTICULAR NEEDS. The 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. No employee or agent of The Foundry is authorized to modify this limited warranty, nor to make additional warranties. No action for any breach of the above limited warranty may be commenced more than one (1) year after Licensee's initial receipt of the Software. To the extent any implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO NINETY (90) DAYS AFTER DELIVERY OF THE SOFTWARE TO LICENSEE.

SECTION 10. LIMITED REMEDY.

The exclusive remedy available to the Licensee in the event of a breach of the foregoing limited warranty, TO THE EXCLUSION OF ALL OTHER REMEDIES, is for Licensee to destroy all copies of the Software, send The Foundry a written certification of such destruction and, upon The Foundry's receipt of such certification, The Foundry will make a replacement copy of the Software available to Licensee.

SECTION 11. INDEMNIFICATION.

Licensee agrees to indemnify, hold harmless and defend The Foundry, the Third Party Licensors and The Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives (collectively, the "Released Parties") from all claims, defense costs (including, but not limited to, attorneys' fees), judgments, settlements and other expenses arising from or connected with the operation of Licensee's business or Licensee's possession or use of the Software or Documentation.

SECTION 12. LIMITED LIABILITY.

In no event shall the Released Parties' cumulative liability to Licensee or any other party for any loss or damages resulting from any claims, demands or actions arising out of or relating to this Agreement (or the Software or Documentation contemplated herein) exceed the License Fee paid to The Foundry or its authorized reseller for use of the Software. Furthermore, IN NO EVENT SHALL THE RELEASED PARTIES BE LIABLE TO LICENSEE UNDER ANY THEORY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, EXEMPLARY OR CONSEQUENTIAL DAMAGES (INCLUDING DAMAGES FOR LOSS OF BUSINESS OR LOSS OF PROFITS) OR THE COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, REGARDLESS OF WHETHER THE RELEASED PARTIES KNOW OR HAVE REASON TO KNOW OF THE POSSIBILITY OF SUCH DAMAGES AND REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL

PURPOSE. No action arising out of or related to this Agreement, regardless of form, may be brought by Licensee more than one (1) year after Licensee's initial receipt of the Software; provided, however, to the extent such one (1) year limit may not be valid under applicable law, then such period shall be limited to the shortest period allowed by law.

SECTION 13. TERM; TERMINATION.

This Agreement is effective upon Licensee's acceptance of the terms hereof and Licensee's payment of the License Fee, and the Agreement will remain in effect until termination. If Licensee breaches this Agreement, The Foundry may terminate the License granted hereunder by notice to Licensee. In the event the License is terminated, Licensee will either return to The Foundry all copies of the Software and Documentation in Licensee's possession or, if The Foundry directs in writing, destroy all such copies. In the later case, if requested by The Foundry, Licensee shall provide The Foundry with a certificate signed by an officer of Licensee confirming that the foregoing destruction has been completed.

SECTION 14. CONFIDENTIALITY.

Licensee agrees that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation are proprietary and confidential information of The Foundry or, as the case may be, the Third Party Licensors, and that all such information and any communications relating thereto (collectively, "Confidential Information") are confidential and a fundamental and important trade secret of The Foundry or the Third Party Licensors. Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" of such Confidential Information, and shall advise any recipients of Confidential Information that it is to be used only as authorized in this Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information available to any other of the Licensee's employees or to any third parties without the express written consent of The 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 such 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. The 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. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that The Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or equitable relief as may be deemed proper by a court of competent jurisdiction.

SECTION 15. INSPECTION.

Licensee shall advise The Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit The Foundry or its authorized agents to inspect all such locations during normal business hours and on reasonable advance notice.

SECTION 16. NONSOLICITATION.

Licensee agrees not to solicit for employment or retention any of The Foundry's current or future employees who were or are involved in the development and/or creation of the Software.

SECTION 17. U.S. GOVERNMENT LICENSE RIGHTS.

The Software, Documentation and/or data delivered hereunder are subject to the terms of this Agreement and in no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication or disclosure by the U.S. Government is subject to the applicable restrictions of: (i) FAR §52.227-14 ALTS I, II and III (June 1987); (ii) FAR §52.227-19 (June 1987); (iii) FAR §12.211 and 12.212; and/or (iv) DFARS §227.7202-1(a) and DFARS §227.7202-3.

The Software is the subject of the following notices:

- Copyright © 2012 The Foundry Visionmongers, Ltd.. All Rights Reserved.
- Unpublished-rights reserved under the Copyright Laws of the United Kingdom.

SECTION 18. SURVIVAL.

Sections 2, 3, 5, 6, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 and 20 shall survive any termination or expiration of this Agreement.

SECTION 19. IMPORT/EXPORT CONTROLS.

To the extent that any Software made available hereunder is subject to restrictions upon export and/or reexport from the United States, Licensee agrees to comply with, and not act or fail to act in any way that would violate, the applicable international, national, state, regional and local laws and regulations, including, without limitation, the United States Foreign Corrupt Practices Act, the Export Administration Act and the Export Administration Regulations, as amended or otherwise modified from time to time, and neither The Foundry nor Licensee shall be required under this Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations.

SECTION 20. MISCELLANEOUS.

This Agreement is the exclusive agreement between the parties concerning the subject matter hereof and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject. This Agreement may be modified only by a written instrument signed by both parties. If any action is brought by either party to this Agreement against the other party regarding the subject matter hereof, the prevailing party shall be entitled to recover, in addition to any other relief granted, reasonable attorneys' fees and expenses of litigation. Should any term of this Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms of this Agreement. The failure of either party to enforce any rights granted hereunder or to take action against the other party in the event of any breach hereunder shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches. This Agreement shall be governed by, and construed in accordance with English Law.

The 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 The Foundry under this Agreement that is capable of application to any proprietary or other right of that Third Party Licensor in or in relation to the Software. The Foundry and Licensee reserve the right under section 2(3)(a) of the Act to rescind, terminate or vary this Agreement without the consent of any Third Party Licensor.

Copyright © 2012 The Foundry Visionmongers Ltd. All Rights Reserved. Do not duplicate.