



GUIDE TO FLIX

Version 8.0.0

Flix™ Flix Guide. Copyright © 2025 The Foundry Visionmongers Ltd. All Rights Reserved. Use of this Flix Guide and the Flix software is subject to an End User License Agreement (the "EULA"), the terms of which are incorporated herein by reference. This Flix Guide and the Flix software may be used or copied only in accordance with the terms of the EULA. This Flix Guide, the Flix 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 this Flix Flix Guide, [End User License Agreement \(EULA\)](#).

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

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

Flix™ software © 2025 The Foundry Visionmongers Ltd. All Rights Reserved. Flix™ is a trademark of The Foundry Visionmongers Ltd.

Sony Pictures Imageworks is a trademark of Sony Pictures Imageworks. Sony Pictures Animation is a trademark of Sony Pictures Animation.

Linux ® is a registered trademark of Linus Torvalds.

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

Windows ® is the registered trademark of Microsoft Corporation.

© 2012 Google Inc. All rights reserved. Chrome™ browser is a trademark of Google Inc.

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

Adobe ® and Photoshop ®, and Premiere ® 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.

The Foundry
5 Golden Square
London
W1F 9HT

Rev: Wednesday, October 22, 2025

Contents

Flix Install Guide	18
Installing Flix Server	18
Flix Server System Requirements	18
Recommended Requirements	18
Minimum Requirements	19
Installing and Running MySQL	20
Download Flix Server	20
Configuring Flix Server	21
Licensing Flix Server	23
Single Server	23
Multiple servers	23
Setting a Custom Assets Directory	24
Store Show Assets on the Server in Subdirectories	24
Setting up Shared Storage	25
OAuth	26
Setting Up Email Notifications	27
Setting Up HTTPS	28
How to Set Up a Reverse Proxy Server	29
Running Flix Server	31
Testing the Connection to Flix Server	32

Managing Your Flix Deployment	33
Upgrading Flix	33
Rolling back to an earlier version of Flix	38
Archive, Restore or Delete a Show	38
Adding Servers	40
Server Management	41
Migrating Assets When Switching to a New Assets Directory	42
Command Line (CLI) Arguments	43
Installing and Launching Flix Client	43
Flix Client System Requirements	43
Installing and Launching Flix Client	44
Setting Up the Client for HTTPS Connection to the Server	48
Running Flix in a Web Browser (Beta)	48
Setting Up User Authentication	49
Control User Access with Group and Role Permissions	50
Creating a User Account	50
Editing and Deleting User Accounts	51
Creating Groups and Assigning Roles to Manage Permissions	53
Add Custom Roles	57
Single Sign-On	58
IdP Settings	59
Linking Flix to Your IdP	60

LDAP (Lightweight Directory Access Protocol)	61
User Search	62
Group Search	63
Auto-assign Group-Role Pairs	63
Example LDAP Flix Config File	64
OAuth	65
Managing Extensions & Plugins	66
Installing Extensions	67
Checking Versions	67
Viewing Extension Documentation	68
Uninstalling Extensions	68
Automated Server Polling for Extension Updates	68
Manually Adding Extensions to the Server	69
Plugins	69
Plugin Deployment	69
Managing Plugins	70
Safe Mode	70
Flix User Guide	71
Getting Started	71
Navigating Through Flix	71
Loading an Existing Sequence	72

Creating a Clean Version of an Edit	73
Creating a Show	73
In the video:	73
Details	74
Configuration	74
Permissions	75
Editing a Show	76
Hiding a Show	76
Starring a Show	77
Creating an Episode	78
Editing an Episode	79
Hiding an episode	80
Creating Additional Seasons	80
Creating a Sequence	81
Editing a Sequence	82
Color Tagging a Sequence	83
Hiding a Sequence	84
Keeping Edits Organized	86
Color Tagging a Sequence Revision	87
Hiding Sequence Revisions	88
Managing Autosaves	88
Settings & Preferences	89
Studio Settings	89
Show Level Settings	89

User Level Settings	90
Setting Naming Conventions	90
Using the Panel Browser	92
In the video:	92
Quickly Locate Panels Using the Panel ID	93
Highlighting Panels	93
Hiding Panels	94
Using Shots in Flix	95
Editing Shots	96
Shots Properties Panel	97
Custom Workflows	98
Naming Shots	98
Setting Shot Number Format	99
Renaming Shots	99
Exporting Shots During Publishing	101
Shots and Storyboard Pro	103
Giving and Managing Feedback	103
Adding Comments to Panels	104
Annotating Panels	106
Customizing Contact Sheet Templates	108
Assigning Templates to Shows	110
Editing an Existing Template	111
Creating a New Contact Sheet Template	112

Locating Panels and Source Files	113
Locating Panels in the Library	113
Browsing Source Files in the Source File Library	116
Filtering the Source Files	117
Flix for Story	118
Importing Panels into Flix	118
Panel Versioning	119
See Import Progress at a Glance in the Transfer Queue	122
Keep Working During File Transfers	126
Actions That Stop File Transfers	129
Preview Artwork and Movies from Panels in a Sequence	129
Making Changes to the Edit	131
Editing Actions:	132
Editing Panels Already in Flix	133
Adding or Editing Dialogue	133
Method 1	133
Method 2	136
Recording a Pitch	137
Flix for Editorial	139
Setting up Flix with Adobe Premiere	139
Setting up Flix with Avid Media Composer	142

Preserve Sequence Timings from Flix	145
Publishing a Sequence to Editorial	146
Filtering a Sequence by Published/Unpublished Panels	149
Importing Your Sequence to Adobe Premiere	150
Importing Your Sequence to Avid Media Composer	151
Importing Your Dialogue File as Captions in Avid Media Composer	152
Sending a Sequence Revision back to Flix from Editorial	153
Exporting Your Sequence Revision from Adobe Premiere	154
Exporting Your Sequence Revision from Avid Media Composer	155
Importing Your Sequence in Flix	158
Viewing Import Summary	161
Opening New Revisions	161
Import Logs	162
View Related Panels from Editorial in Flix	163
Access Editorial Publish Information from Your Inbox	164
Photoshop Extension Workflows	166
Setting Up Flix to Work with Photoshop	166
Install the Flix Extension for Photoshop	166
Customizing the Flix Plugin	167
Photoshop Workflow Options	168
About Source Files	168
Setting the Aspect Ratio	169
Pushing Panels to Flix	169

Sending New Panels and Thumbnails To Flix	170
Annotating a Panel Using Photoshop	171
Working with Photoshop's Layer Comps	171
Working with Photoshop's Frame Timeline	173
Working with Photoshop's Layers/Groups	174
Photoshop Actions	176
Photoshop Plugin Overview	176
Send vs Replace Actions	178
Sending the Source File	178
Create Actions	178
Image Actions	179
Annotation Actions	179
Layer Comp Actions	180
Frame Actions	180
Layer Actions	181
Crop Actions	181
Maya Extension Workflows	182
Setting up Flix to Work With Maya	182
Maya Flix Plugin Overview	183
Sending New Panels and Panel Revisions from Maya to Flix	183

Sending Scene Data to Flix	184
Sending Panels to Maya	185
Using Panels in Maya	186
Storyboard Pro & Flix	188
Setting Up Flix for Storyboard Pro	188
Installing the Flix Extension for Storyboard Pro	189
Importing Storyboard Pro Projects into Flix	189
Smart Panel Relinking Between Flix and Storyboard Pro	196
Exporting Flix Sequences to Storyboard Pro	197
Re-Conforming Your Project in Storyboard Pro	200
Storyboard Pro Extension Workflows	203
Access Editorial Publish Information from Your Inbox	205
Flix Reference Guide	207
Flix Server Options	207
Required Options	207
MySQL	208
Asset Storage	208
Licensing	208
Authentication	209
LDAP (Lightweight Directory Access Protocol)	209

User Search	210
Group Search	210
Auto-assign Group-Role Pairs	211
Example LDAP Flix Config File	211
Email	212
HTTPS	212
Proxy	213
Security	213
Logs	213
Other	214
Flix Preferences	214
General	214
Third-Party Apps	215
Sketching Tools	215
Editorial Tools	217
Adobe Premiere	217
Publish and Export	223
Publish	223
Export	223
QuickTime Export	224
Audio	224
Panel Browser	225

Notifications	226
Advanced	227
Flix Logs	227
Server Logs	228
Accessing Server Logs in Flix Client	228
Client Logs	229
Transfer Logs	229
Download Logs	230
Keyboard Shortcuts	230
Project	230
General Edit Actions	231
Player	231
Panels and Shots	231
Window	233
Flix Resources	234
Flix Top Five Things to Onboard New Story Artists	234
In the video:	234
1. Importing Artwork into Flix	235
2. Panel ID vs Panel Index	238
3. Editing Existing Panels	239
4. Adding Dialogue Text	240
5. Adding Audio to a Pitch	242

Flix Knowledge Base Articles	243
Flix Example Scripts	243
Flix Server Technical Overview	244
Architecture	244
Server Requirements	245
Hardware Specifications	246
Horizontal Scaling	246
Vertical Scaling	247
Storage	247
Database	248
API and Websockets	249
Security	249
Licensing	250
Remote Access	250
Minimum Bandwidth Requirements for Flix	251
Flix Server Technical Overview	252
Architecture	252
Server Requirements	253
Hardware Specifications	254
Horizontal Scaling	254
Vertical Scaling	255

Storage	255
Database	256
API and Websockets	257
Security	258
Licensing	258
Remote Access	259
Minimum Bandwidth Requirements for Flix	259
Flix Server and Client API Reference	260
Supported File Formats	262
Panel Formats	262
Stills	262
Audio Formats	262
Flix and Storyboard Pro 20 Container Formats	262
Flix and Adobe Premiere Pro Container Formats	263
Flix and Avid Media Composer Container Formats	264
Third-Party Application Support	265
Flix 8.0 Series	265
Applications	265
Flix	266
Flix 7.1 Series	266
Applications	266

Flix	267
Flix 7.0 Series	267
Applications	267
Flix	268
Flix 6.6 Series	268
Applications	268
Flix	269
Flix 6.5 Series	269
Applications	269
Flix	270
Flix 6.4 Series	270
Applications	270
Flix	271
Flix 6.3.7	271
Applications	271
Flix	272
Flix 6.3.6	272
Applications	272
Flix	273
Flix 6.3.5	273
Applications	273
Flix	274
Flix 6.3.4	274
Applications	274

Flix	275
Flix 6.3.3	275
Flix	275
Flix 6.3.2	276
Applications	276
Flix	276
Third-Party Software Notices	277
Flix Server	277
Transfer Utility	296
Python	306
Flix Client	306
Photoshop	356
Maya	391
Third-Party Licenses	393
End User License Agreement (EULA)	394

Flix Install Guide

Installing Flix Server

Copyright © 2025 The Foundry Visionmongers Ltd.

The following instructions are a guide to get your Flix Server up and running quickly.



Note: If you're new to Flix Server installation, please read [Flix Server Technical Overview](#) to familiarize yourself with server requirements and architecture before getting started. The following System Requirements are recommendations only. Actual server specifications vary greatly depending on factors such as the number of users and the server usage required for different productions.

Flix Server System Requirements



Warning: For security reasons, the date and time for the machine on which Flix Server is installed needs to be set correctly. This also applies to virtual machines. For more information, please refer to this [Knowledge Base article](#).

Recommended Requirements



Note: Flix Server is built and tested exclusively on Intel processors. Running the Flix Server on AMD processors may lead to unexpected runtime errors.

Linux:

Operating System	
	CentOS 7 64-bit (this will be deprecated in future releases)

	• Rocky 9
RAM	At least 16 GB  Note: The volume where the <code>/tmp</code> directory resides should have at least twice as much free space as the largest AAF imported into Flix. We recommend at least 16 GB, which is enough for most studios.
CPU	A 64-bit Intel processor @ 3.2 GHz, or higher 8 cores, or more

Minimum Requirements

Linux



Note: Flix Server is built and tested exclusively on Intel processors. Running the Flix Server on AMD processors may lead to unexpected runtime errors.

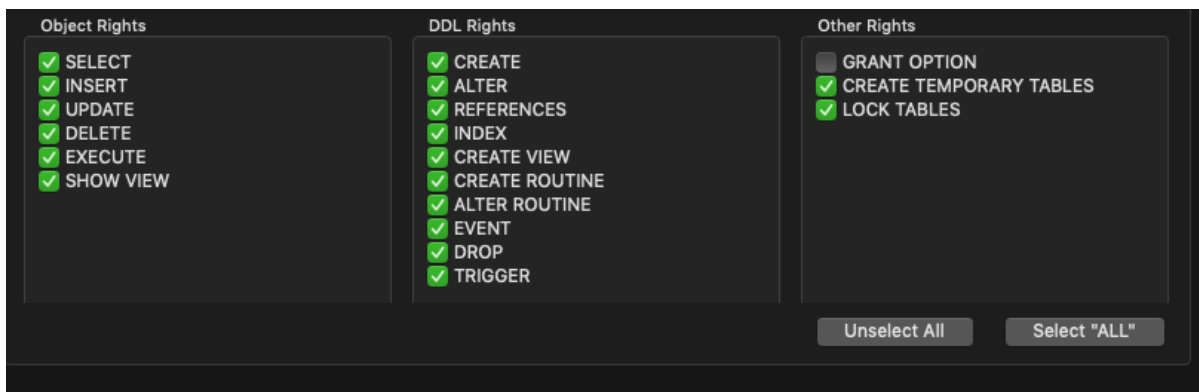
Item	Requirements
Operating System	• CentOS 7 64-bit (this will be deprecated in future releases) • Rocky 9
RAM	8 GB  Note: The volume where the <code>/tmp</code> directory resides should have at least twice as much free space as the largest AAF imported into Flix. We recommend at least 8 GB, which is the minimum storage requirement.
CPU	A 64-bit Intel processor @ 2 GHz, or higher

Installing and Running MySQL

Flix Server is the server application for Flix. Installation of MySQL 8 or 5.7 is required for Flix Server to run.

The MySQL database doesn't need to be running on the same machine as the Flix Server. However, it must accept external connections over a network to allow for communication with your Flix Server(s). Refer to this [Knowledge Base article](#) for more information.

1. Install MySQL and check that your MySQL server is running. Instructions for this can be found in this [Knowledge Base article](#) or online.
2. Ensure the username that Flix uses to connect to the MySQL database has the following permissions:



Object Rights	DDL Rights	Other Rights
☒ SELECT	☒ CREATE	☐ GRANT OPTION
☒ INSERT	☒ ALTER	☒ CREATE TEMPORARY TABLES
☒ UPDATE	☒ REFERENCES	☒ LOCK TABLES
☒ DELETE	☒ INDEX	
☒ EXECUTE	☒ CREATE VIEW	
☒ SHOW VIEW	☒ CREATE ROUTINE	
	☒ ALTER ROUTINE	
	☒ EVENT	
	☒ DROP	
	☒ TRIGGER	

Unselect All Select "ALL"

3. Set the `innodb_autoinc_lock_mode` global variable to 1, consecutive mode, on the MySQL server. If MySQL is configured to use interleaved mode, the server will not start.



Article: For guidance on upgrading from MySQL 5.7 to MySQL 8, please see the following [Knowledge Base article](#).

Download Flix Server

1. Download Flix Server at <https://www.foundry.com/products/flix/download>.

For Linux, a **.tar.gz** file is downloaded.



Note: You must be logged in to your Foundry account to download Flix Server.

2. Open/untar the downloaded archive.

The folder contains the following:

flix_server - a binary file which you execute to start server Flix Server.

flix-server-utility - a helper binary which Flix Server starts upon startup.

thirdparty - a folder containing thirdparty libraries Flix Server relies on.

Example of extracted archive:

```
flix_server_6.3.1_20
├── flix_server
├── flix-server-utility
└── thirdparty
```

3. Place these files wherever you want Flix Server to be installed.

Configuring Flix Server

You need to set up a **config.yml** file before you can run Flix Server. By default, Flix Server reads the config file from the same directory as its binary file (**flix_server**). In order to make upgrading to future versions of Flix easier, we strongly recommend storing the config file in a different location and pointing to it when running Flix Server using the **--config-file** CLI argument.

Example of **config.yml** stored alongside the Flix Server directory:

```
[[flix.user@flix002 foundry]$ tree -L 2
.
├── config.yml
└── flix_server_6.3.1_20
    ├── flix_server
    ├── flix-server-utility
    └── thirdparty
```

See [Running Flix Server](#) for more information.

Here's an example **config.yml**. You can download and edit a sample [here](#).

```
hostname: flix001.mycompany.com
http_port: 8080
mysql_hostname: db1.mycompany.com
mysql_username: root
mysql_password: password
```



Note: Ensure that the **hostname** option is set to a publicly available hostname or IP address. We recommend a fully qualified domain name and unique hostname for each server.



Note: The **mysql_username** and **mysql_password** are for the user specified in [Installing and Running MySQL](#) with those permissions.



Tip: You may need to add a firewall port exception to allow access on the ports through which Flix communicates with clients. See your OS documentation for more information on firewalls.

The Flix Server Address end users require is in the following format: **http://[hostname]:[port]**. Using the example **config.yml** above, the Flix Server Address is: **http://flix001.mycompany.com:8080**.



Warning: macOS users: If you're installing the server on Mac OS, hostnames contain **.local** and aren't supported. Multicast DNS (mDNS) domains are not supported. As a workaround, set the public IP and hostname in your **/etc/hosts** file, for example: 172.168.3.42 flix.local



Tip: See [Flix Server Options](#) for a full list of configuration options.



Article: For a best practice guide on adding new servers and upgrading Flix, check out the following [Knowledge Base Article](#).

Licensing Flix Server

Single Server

If you plan on using only a single Flix Server then all you need is a node-locked license for the Flix Server machine.



Note: Please refer to our [Licensing Documentation](#) for instructions on installing a node-locked license.

For more information on setting up a Flix 6 license, see the following Knowledge Base article: [Flix 6 Licensing Setup and Troubleshooting](#).

Multiple servers

If you plan on using more than one Flix server, we recommend using a floating license, either hosted on one of the Flix Servers or from another dedicated license server.



Note: Please refer to our [Licensing Documentation](#) for instructions on installing a floating license.

You will then need to point all of your Flix Servers to use the license floating from your Foundry License Server by setting the **floating_license_hostname** and **floating_license_port** config options.

For example, if your Foundry License Server is running on a machine called "my_license_server" and using the default port 4101 then you would set the following:

```
floating_license_hostname: my_license_server  
floating_license_port: 4101
```



Note: If you're unsure about the hostname and/or port to point your Flix Server to, you can refer to [this section in our Licensing Documentation](#).

You have now completed the minimum steps for setting up Flix Server. The following instructions guide you through more advanced set up options. If you want to return to these custom options later, you can jump to [Running Flix Server](#).

Setting a Custom Assets Directory

By default, your Flix assets are stored in an 'assets' directory, which is located in your install folder alongside the `flix_server` binary. We strongly recommend that you change this to a custom location, to make the upgrading process easier for all future releases of Flix. To do this, set the **asset_directory** option in your **config.yml** file. For example:

```
asset_directory: /mnt/flix-assets
```



Note: If you were previously using Flix without specifying the asset directory, or changed the value of the `asset_directory` option, see [Migrating Assets When Switching to a New Assets Directory](#) to import assets from the original directory.



Note: From Flix 6.3.0 onwards, assets are stored in sub-directories per show. See [Release Notes for Flix 6.3](#) for more information.

Store Show Assets on the Server in Subdirectories

In large productions, asset numbers can reach the hundreds of thousands, so it's a good idea to store assets across multiple subdirectories rather than a single directory, as this can provide performance benefits when working at such a large scale.

To set new files to be stored in subdirectories, update the Flix Server config file to include the following:

```
sub_directory:  
  disabled: false  
  dir_name_length: 3
```

The **disabled** setting decides whether the files will be stored all in one place or in subdirectories.

The `dir_name_length` setting defines how the files will be split. The number stands for the number of characters to group files by. All Flix files are stored with a number prefix, so the higher the number chosen for `dir_name_length`, the more subdirectories with less files in them. The lower the number, the more files will be split in less subdirectories.

Migrate Existing Assets into Subdirectories

To migrate preexisting assets to a new file structure with subdirectories:

- Change the config to reflect the desired subdivision. See above example.
- Run Flix Server with the `--migrate-files` flag, followed by the `--show-id` flag to better manage the migration.



Note: The server will launch once the process is finished, but this can be interrupted and restarted at a more convenient time.

Setting up Shared Storage

Flix can be configured to store assets on shared storage, accessible by all servers. This way assets aren't siloed across multiple servers. To switch to shared storage, add the `shared_storage` setting into the `config.yml` file.

By default, under the assets directory, each server will have its own directory specified by the server identification number (a long string of numbers/characters) where it stores its own assets. If you want all the servers to store assets in a shared directory, set the `shared_storage` option to `true`. Every Flix Server stores the assets in the same directory specified by the `asset_directory` folder.



Note: If you were previously using Flix without shared storage and want to switch, see [Migrating Assets When Switching to a New Assets Directory](#) to import assets from each server's asset directory to the shared one.

OAuth

The OAuth configuration allows users to log in to Flix using their credentials provided by an OAuth service. Using the example below, configure your Flix server config file to use OAuth. Once the configuration is enabled, restart your Flix Server, and users should now be able to log in using the **Sign In with Google** button.

use_oauth - This determines whether to use OAuth authentication or not.

username - This determines whether artists are authenticated by 'username' or 'email' address and is used to create their Flix username. Setting the **username** parameter to **email** forces Flix to create unique usernames.



Note: If the username parameter is blank, misspelled, or omitted, authentication defaults to 'username'.

domain - This specifies the domain name for your OAuth provided credentials.

providers - This specifies the OAuth provider.

Example of OAuth:

```
oauth:
  use_oauth: true
  username: email
  domain: my_domain.com
  providers:
    - Google
```



Warning: Please be certain not to have LDAP and OAuth both enabled.

OAuth doesn't currently obtain the user groups from Google like it does for LDAP, so group management for project access needs to be done using Flix's built-in Group Management.



Warning: OAuth is not supported when using HTTPS. Due to the design of OAuth, it requires making a HTTP callback to the Flix Server. With HTTPS enabled, this callback would also need to be via HTTPS, however without external access to your certificate authority, this request would fail. Currently this limitation prevents OAuth and HTTPS from being used at the same time.



Note: Currently the only OAuth provider supported is Google.

Setting Up Email Notifications

Flix can be configured to send email notifications to members of the production when publishing a sequence to and from editorial.

To do this, Flix uses your SMTP server. To set up email notifications, set the following options in your server's config.yml file:

smtp_hostname - Hostname of the SMTP server to use.

smtp_port - Port number of the SMTP server to use.

smtp_username (optional) - Username of the account to authenticate with the SMTP server.

smtp_password (optional) - Password of the account to authenticate with the SMTP server.

smtp_send_from (optional) - Sets the email address Flix uses for notifications.

Example of Email Notifications config:

```
smtp:
  smtp_hostname: smtp.mystudio.com
  smtp_port: 465
  smtp_username: example@mystudio.com
  smtp_password: MyP@ssword
  smtp_send_from: flix_publishes@mystudio.com
```



Note: If the **smtp_username** and **smtp_password** config options are not set, Flix Server attempts to connect to the smtp server without authenticating when sending notification emails.



Note: If the option for **smtp_send_from** is not set, Flix sends email notifications from the email address of the user publishing to and from editorial. If the user doesn't have an email address, no email notifications are sent.

Setting Up HTTPS

By default, Flix's security relies on every request between Client and Server being signed. For added security, you have the option to run Flix over HTTPS, so that all communication between Flix Client and Flix Server is encrypted. This would be preferable if, for example, running Flix on a publicly available server.

To set up HTTPS, you will need to set the following options in your server's config.yml file:

ca_file - Add this option and the path to a CA certificate file if using self-signed certificates.

cert - Add this option and the path to a TLS certificate file (public key).

key - Add this option and the path to the TLS key file (private key).



Note: Flix supports TLS 1.0 and TLS 1.1

Example of HTTPS config:

```
tls:
  ca_file: /Foundry/cert/ca.crt
  cert: /Foundry/cert/cert.crt
  key: /Foundry/cert/server.key
```



Note: The **ca_file** option should only be set if using self-signed certificates, otherwise only the cert (public key) and the key (private key) need to be set.



Note: When using self-signed certificates, make sure these have been installed on your end users' machines to enable communication with Flix Server. For more information, go to [Setting Up the Client for HTTPS Connection to the Server](#).



Warning: OAuth is not supported when using HTTPS. Due to the design of OAuth, it requires making a HTTP callback to the Flix Server. With HTTPS enabled, this callback would also need to be via HTTPS, however without external access to your certificate authority, this request would fail. Currently this limitation prevents OAuth and HTTPS from being used at the same time.

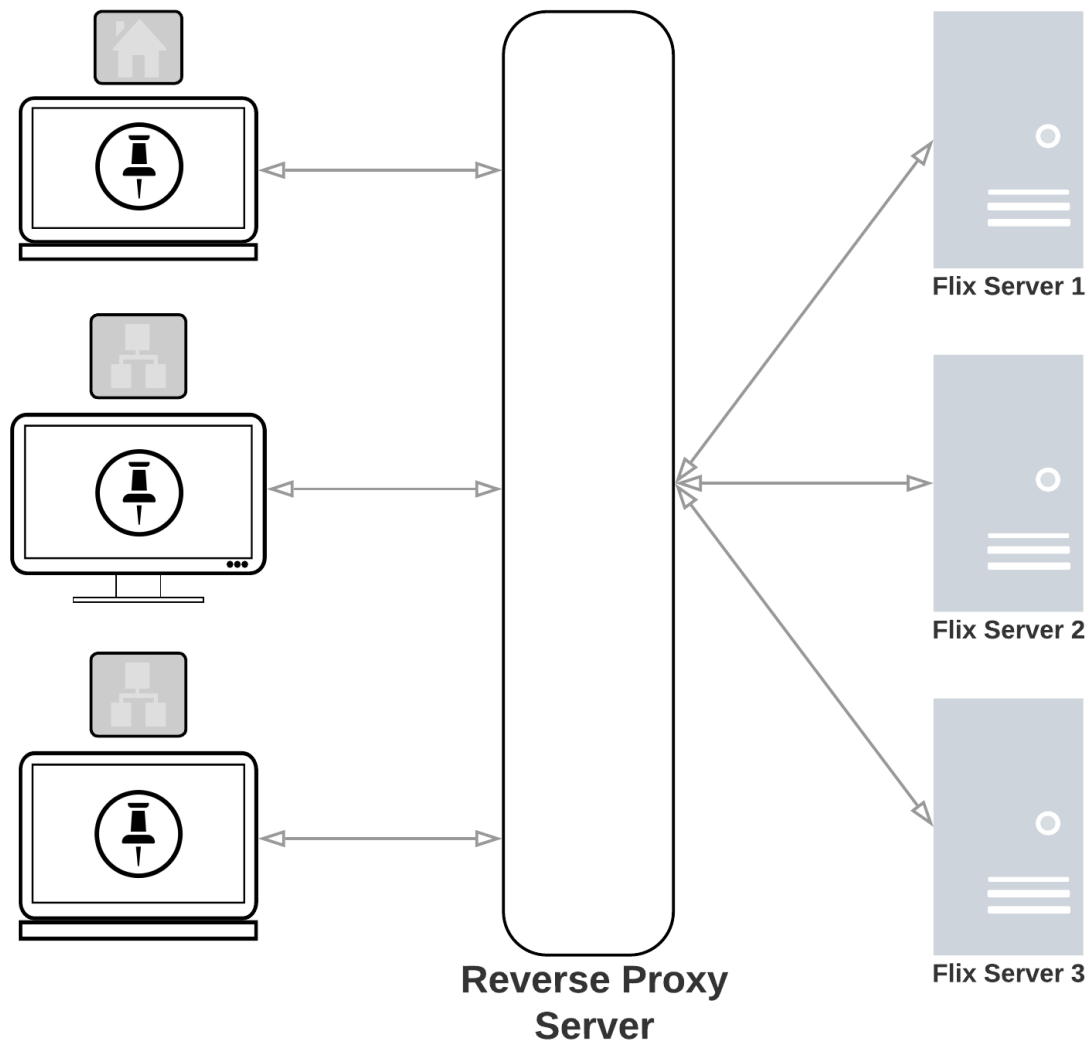
How to Set Up a Reverse Proxy Server

Making use of integrated cloud platform features such as auto-scaling and load balancing can be challenging, because Flix Clients require direct access to all Flix Servers to run effectively. Flix supports reverse proxy servers to simplify this process, which are commonly used intermediary servers that manage requests for resources across the internet. Using a reverse proxy server allows you to customize the connections between Flix Clients and Flix Servers to fit your needs, including:

Load Balancing - distribute Flix Client requests across the available Flix Servers to maximize efficiency and avoid any one server being overloaded. Without load balancing, all Flix Client asset requests in your studio might be directed to a single Flix Server despite there being more servers available to handle requests.

Web Acceleration - compress inbound and outbound data, as well as cache commonly requested content, both of which speed up the flow of traffic between clients and servers.

Security and Anonymity - hide the identities of Flix Servers, acting as an additional defense against security attacks, by masking the IP address in a similar way to a VPN.



The reverse proxy server handles server load bearing and hides your Flix Servers for added security.

Flix includes a health check to determine if a server is "healthy" and ready to handle remote procedural calls (RPCs) and then the reverse proxy server can act accordingly. For example, if one Flix Server reports as unhealthy, the remote proxy server can redirect Flix Client RPC requests to another available server.



Tip: The gRPC health check protocol in use by Flix is available from GitHub:

<https://github.com/grpc/grpc/blob/master/doc/health-checking.md>

Here's an example code snippet from the Flix [config.yml](#) showing you how to set up the proxy and security options:

```

proxy:
  host: flix.company.com
  port: 8989
  transfer_port: 8881

security:
  disable_hostname_check: true
  skip_transfer_tls: true

```

- **host** - The hostname of the reverse proxy server. This is the URL that Flix Client uses to connect to the server HTTP API.
- **port** - The HTTP(s) port the reverse proxy server is listening on.
- **transfer_port** - The grpcs port the reverse proxy server is listening on for file transfers.
- **disable_hostname_check** boolean [default: false] - If set to true, Flix Server does not check that the host header requested by the Flix Client matches the hostname of the server. This can be used for reverse proxy setups where the reverse proxy's public URL does not match the Flix Server's hostname.
- **skip_transfer_tls** boolean [default: false] - If set to true, Flix Server listens for insecure (non-TLS) RPC connections. This can be used for reverse proxy setups where the reverse proxy terminates the TLS connection (grpcs://) from the Flix Client and uses an insecure connection upstream to the Flix Server.



Note: Flix Clients always communicate using gRPCs, never using insecure (non-TLS) RPC connections.

Running Flix Server

Your operating system may not give run permission to the **flix_server** binary by default. To ensure you can run it, enter the following command:

```
chmod +x flix_server
```

If your server config file is located in the same directory as your Flix Server binary, you can run the server as follows:

```
./flix_server --verbose
```



Note: The **--verbose** flag is optional, but is useful as it displays a more detailed log output in the console.

You should see a readout like the screen shot below once Flix Server is successfully running. Using the **--verbose** flag would display more information than shown here.

```

(flix.user@flix002 flix_server_6.3.1_20)$ sudo ./flix_server --config-file /usr/local/foundry/config.yml
[sudo] password for flix.user:
Flix Server
INFO[0000] licensing.writeLicFile: Creating RLM floating license file Version=6.3.1_20 Path=/usr/local/foundry/RLM/flix_foundry.lic
INFO[0000] main.getApp: server version found Version=6.3.1_20
INFO[0000] entitlements.(*Entitlement).init: Waiting rpc server... Rpc server="http://localhost:8383"
INFO[0000] licensing.(*ServerLicense).lic: Licensed: Expires on 05-Feb-2025 Max concurrent users=5
INFO[0000] database.verifySchema: Versions matched! Version=25 Desired Version=25
INFO[0000] asset.CheckAssetsDirectory: Assetter: Using assets directory /usr/local/foundry/assets/7510075e-ab89-40d7-906f-bd4c137c17eb
INFO[0000] main.getApp: publishes will fail unless you have the font installed error="cannot find font 'arial.ttf' in user or system directories"
INFO[0000] main.getApp: Server Endpoint: http://flix002.thefoundry.co.uk:8080
INFO[0000] main.getApp: App created with configuration AssetDirectory=/usr/local/foundry/assets/7510075e-ab89-40d7-906f-bd4c137c17eb
DBInstall=false First run=false HTTPPort=8080 Hostname=flix002.thefoundry.co.uk Ident=7510075e-ab89-40d7-906f-bd4c137c17eb MySQLDbName=flix MySQLHo
stname=localhost Region=default RepPort=9876 ServerName=flix Shared Storage=false
INFO[0000] main.getApp: OAuth config Domain=foundry.com Providers="[Google]" Address=":9876"
INFO[0000] rpc.(*Server).Run: opening RPC listener
WARN[0000] masternode.MasterNode.CurrentMaster: master not set
INFO[0000] masternode.GetMaster: could not collect current master, performing election
INFO[0000] masternode.setMaster: new master node elected Server ID=7510075e-ab89-40d7-906f-bd4c137c17eb
INFO[0000] pycomms.PyComms.Log: [rpc.py:184] <module>: initialised Python logging Go server IP=flix002.thefoundry.co.uk Go server port=9876 Python
server port=9991
INFO[0000] pycomms.PyComms.Log: [rpc.py:149] serve: python RPC server has started Port=9991
  
```

Note the Server Endpoint info as you'll need to provide end users this to log on to Flix.

Non-sensitive information read from the config file is displayed, which helps to ensure the configuration has been applied as expected.

As mentioned above, we strongly recommend storing the server config file outside the Flix Server directory. To point Flix Server to the location of the server config file, use the **--config-file** CLI flag. For example:

```
./flix_server --verbose --config-file /mnt/flix/flix_config_prod.yml
```



Warning: If Arial font is not installed or cannot be located on the operating system running Flix Server, publishes will fail with the following error: **"cannot find font 'arial.ttf' in user or system directories"**. We recommend that you install the Arial font into your system's default font directory, or specify a custom font directory using the **font_directory** option. For example: **font_directory: /mnt/flix-fonts**.



Note: The first time the Flix Server is run, it automatically creates an admin user with the username and password both set to **admin**, which you can use to log in to Flix for the first time. It is recommended that you change the default password after the first log in.



Tip: You can set up Flix Server as a service, so that it starts automatically along with the server on which it's installed. Instructions on how to do so can be found in this [Knowledge Base Article](#).

Testing the Connection to Flix Server

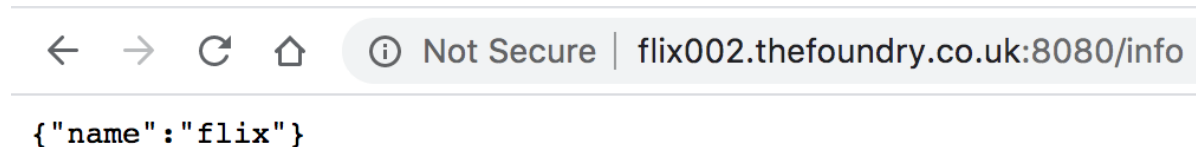
Now that Flix Server is running, it's a good idea to ensure it's accessible by other computers, as end users only access it from another computer at the studio (on the same network), or remotely (over VPN).

To test the connection:

1. Log in to another computer that can communicate with Flix Server, either on the same network or over VPN.
2. Open a Web Browser.
3. In the URL field, type in the Server Endpoint followed by **`/info`**. Using the example from [Running Flix Server](#), the URL would be:

`http://flix002.thefoundry.co.uk:8080/info`

You should see something like this:



If you receive a connection error in the web browser, try the following:

- Ensure Flix Server is running on the server
 - Ensure the computer you're on can resolve the hostname/IP of the server (flix002.thefoundry.co.uk in this example)
 - Ensure the port is open through any firewall that might be running on the server
4. Once Flix Server is running and accessible by other computers on the network, you can install the Flix Client app and connect it to Flix Server. See [Installing and Launching Flix Client](#) for more information.



Tip: Depending on the size of your Flix user base, a single server may not be enough to provide satisfactory performance and speed. Having multiple servers in your Flix Deployment helps ensure better performance across all users, with HTTP(S) requests being spread across all the available resources. For more information on adding another server to your Flix Deployment, see [Adding Servers](#).

Managing Your Flix Deployment

Upgrading Flix

1. Click [here](#) to download the latest version of Flix Server.
2. Extract the files from the download package into the same install directory where Flix Server is currently installed.

A new directory named after the new Flix version is created.

For example, if versions 6.0.5, 6.1.2, and 6.2.2 are installed and you want to upgrade to Flix 6.3.1, the install directory is arranged as follows:

```
[[flrx.user@flrx002 foundry]$ pwd
/usr/local/foundry
[[flrx.user@flrx002 foundry]$ tree -L 2
.
├── assets
│   └── 7510075e-ab89-40d7-906f-bd4c137c17eb
├── backups
│   ├── Mon Jul 14 17:10:19 2019.sql
│   ├── Mon Mar 9 12:14:18 2020.sql
│   └── Wed Jan 9 14:18:24 2019.sql
├── config.yml
├── flrx_server_6.0.5
│   ├── flrx_server
│   ├── flrx_server.log
│   ├── rpc-linux
│   └── thirdparty
├── flrx_server_6.1.2_23
│   ├── flrx_server
│   ├── flrx_server.log
│   ├── flrx-server-utility
│   └── thirdparty
├── flrx_server_6.2.2_30
│   ├── flrx_server
│   ├── flrx_server.log
│   ├── flrx-server-utility
│   └── thirdparty
└── flrx_server_6.3.1_20
    ├── flrx_server
    ├── flrx-server-utility
    └── thirdparty
```



Note: You need to make sure all the **config.yml** settings match your previous version of Flix 6 and the **asset_directory** location is set and pointing to the same location as the assets for your previous version.

3. You can now start the new version of Flix Server, pointing it to the same config file as previously used. Following the previous example, the command would be:

```
./flix_server_6.3.1_20/flix_server --config-file config.yml
```

4. Flix Server prompts you to upgrade the database if a mismatch between server and database versions is detected.

If you want to upgrade your database later, you can use the **--db-upgrade** mode to mutate the database schema to the latest version:

```
./flix_server --db-upgrade --verbose
```



Note: We recommend using the **--verbose** flag to have better visibility over the upgrade progress.

5. Flix Server asks if you want to back up the existing database. Press **Y** to start the backup.

Example of database backup:

```
Do you want to backup the Flix database? [y/N] y
What directory would you like to backup to? (Press enter to use '/Users/brice.banel/Documents/flixProjects/db_backups')
DEBU[0006]database.dumpDB: Backing up db to sql file Path=/Users/brice.banel/Documents/flixProjects/db_backups
DEBU[0006]mysqldump.(*Dumper).Dump: starting dump
DEBU[0006]mysqldump.getTable: getting tables
DEBU[0006]mysqldump.createTable: creating table Table name=access_groups
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=access_groups
DEBU[0006]mysqldump.createTableValues: creating values Table name=access_groups
DEBU[0006]mysqldump.createTable: creating table Table name=access_key
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=access_key
DEBU[0006]mysqldump.createTableValues: creating values Table name=access_key
DEBU[0006]mysqldump.createTable: creating table Table name=asset
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=asset
DEBU[0006]mysqldump.createTableValues: creating values Table name=asset
DEBU[0006]mysqldump.createTable: creating table Table name=asset_media_object
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=asset_media_object
DEBU[0006]mysqldump.createTableValues: creating values Table name=asset_media_object
DEBU[0006]mysqldump.createTable: creating table Table name=asset_media_object_to_media_object
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=asset_media_object_to_media_object
DEBU[0006]mysqldump.createTableValues: creating values Table name=asset_media_object_to_media_object
DEBU[0006]mysqldump.createTable: creating table Table name=dialogue
DEBU[0006]mysqldump.createTableSQL: running table SQL Table name=dialogue
DEBU[0006]mysqldump.createTableValues: creating values Table name=dialogue
```




Note: Backing up your existing database is strongly recommended when running a database upgrade. This is to ensure you can restore that backup in case of issues during the upgrade process. See this [Knowledge Base Article](#) for more information on restoring a Flix Database backup in MySQL.

- Once the backup is complete, Flix Server prompts you to start the database upgrade. Press **Y** to start the upgrade.

Example of upgrading from Flix 6.2 to 6.3:

```
DEBU[0007]mysqldump. (*Dumper).Dump: dumped successfully          Path="/Users/brice.banel/Documents/flixProjects/db_backups/Fri Mar
20 13:15:54 2020.sql"
INFO[0007]database.dumpDB: Created backup file: /Users/brice.banel/Documents/flixProjects/db_backups/Fri Mar 20 13:15:54 2020.sql
WARNING: Making changes to your database. Ensure you have backed up before continuing.
Please ensure you have backed up your Flix database before continuing.
Do you want to continue? [y/N] y
Upgrading DB from v20 to v25
Upgrading DB...
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=0 Version=21
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=1 Version=21
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=2 Version=21
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=0 Version=22
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=0 Version=23
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=0 Version=24
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=0 Version=25
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=1 Version=25
DEBU[0016]database.applyUpDefinitions: adding SQL statement          Statement=2 Version=25
Done.
```

Once the database has been upgraded, you can start the server normally.



Note: You need to upgrade the Flix Client to the same version as your server. You can download the Flix Client from [here](#), or use the auto-update feature when a new version becomes available.

Flix Server Version	Required Database Version
7.1.0 - 7.1.1	90
7.0.0 - 7.0.2	74
6.6.0 - 6.6.2	58
6.5.0 - 6.5.1	56
6.4.0 - 6.4.1	42
6.3.7	29



Note: Contact Flix support if you require the database version for older versions of Flix.



Note: You are presented with the option to copy pre-existing assets into new sub-directories for each show when you upgrade Flix Server. The next time you start the server, the option appears again to migrate the assets. To automatically skip the prompt regarding the 6.3 asset migration, you can use the **--skip-migration** flag when starting Flix Server. We recommend performing the asset migration eventually, as future minor (for example 6.5, 6.6) and major (for example 7.0, 8.0) releases may not support assets stored outside of show sub-directories.

Rolling back to an earlier version of Flix

Your production may at some stage want to roll back to an earlier version of Flix. You can do so with the following command:

```
./flix_server --db-downgrade
```

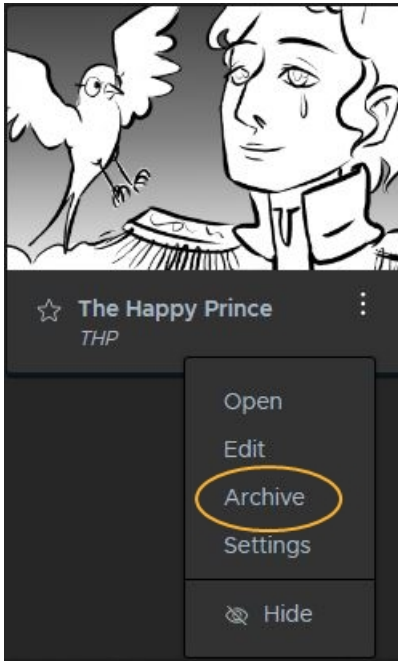
1. Flix asks if you want to backup the database. Type **Y** (yes) or **N** (no).
2. Enter the desired database version to roll back to. Please refer to the table above for reference.
3. Type **Y** (yes) to confirm the database version or **N** (no) to enter a different version.



Note: The latest version of Flix Server needs to be used to run the Downgrade. For example, if downgrading Flix from 6.1.0 to 6.0.5, use Flix Server 6.1.0 to run the downgrade from database version 12 to database version 5.

Archive, Restore or Delete a Show

Admin users can **archive**, **restore** or **delete** shows. Archiving older or inactive projects can help optimize performance and reduce costs, particularly for cloud-hosted instances.



To **archive** a show:

- Click on the show kebab menu and select **Archive**.
- Enter the show tracking code and click **confirm**.

Once the archive process is complete, the show is locked and the archive automatically saved on the server, in the location specified in your config for database backups. You can find the config option under [MySQL](#).

- A locked show is viewable by anyone that had previous access, but it is read-only and cannot be edited. Only admin users can unlock a show.

To **restore** an archived show:

- Click **Restore Archive** on the **Shows** page. A list appears of available show archives in the location specified by the backups config option. You can identify the show you want to restore from the tracking code recorded against the archive.

To **delete** an archived show:

- Click on the show kebab menu and select **Delete**. Enter the show tracking code and click **confirm**.
- A show can only be deleted if it has already been archived and locked. If you unlock a show you will not be able to delete it.



Note: To make the archive bundle as light as possible, the archive process excludes scaled, fullres, and thumbnail media objects. These are regenerated during a show restoration. Opening Sequence Revisions from a recently restored show may take a little longer than usual as Flix transcodes the new files.

Adding Servers

Having multiple servers in your Flix Deployment helps ensure better performance across all users, with HTTP(S) requests spread across all the available resources.

Flix handles its own load balancing, dispatching jobs from different users to all available servers. We strongly advise against setting up your own load balancer in your Flix Deployment, as it is unnecessary and could introduce communication issues between Flix Client and Flix Server.

If moving from a single-server Flix deployment to a multi-server deployment, we strongly recommend using Shared Storage, for all assets from all servers to be stored in a centralized location. We recommend setting up Flix to use Shared Storage for your single-server deployment, and migrating existing assets across to the new shared storage location prior to adding any more servers. See [Setting up Shared Storage](#) for more information.



Tip: If moving from a single-server Flix deployment to a multi-server deployment, licensing is much easier to manage with a floating license, as opposed to a node-locked one. See [Licensing Flix Server](#) for more information, and contact your Sales representative or our Support Team to change your node-locked license for a floating license.

Follow these instructions to add an extra server to an existing Flix deployment. You can download Flix Server [here](#).



Tip: Adding a new server is easiest if all your Virtual Machines access the Flix Server binary from a network location. Otherwise you need to install the Flix Server binary on any new server/VM and ensure all your Flix Servers are using the same version.

1. Ensure your new Flix Server can access the **config.yml** file your other Flix Server(s) are using.
2. Ensure the **hostname** option isn't specified in the **config.yml** file.



Note: If the **hostname** option was specified previously, make sure to run your old Flix Server by specifying its hostname using the **-hostname** CLI flag, as mentioned further below.

3. Ensure your new Flix Server is licensed. See [Licensing Flix Server](#) for more information.
4. Run all Flix Servers with the following 2 CLI flags:

-config-file pointing to your **config.yml** file, for example: `-config-file /mnt/flix/config.yml`

-hostname specifying the server's hostname, for example: `-hostname flix002.mystudio.com`

Example command: `./flix_server -hostname flix002.mystudio.com -config-file /mnt/flix/config.yml -verbose`

You can repeat these steps to add additional servers.



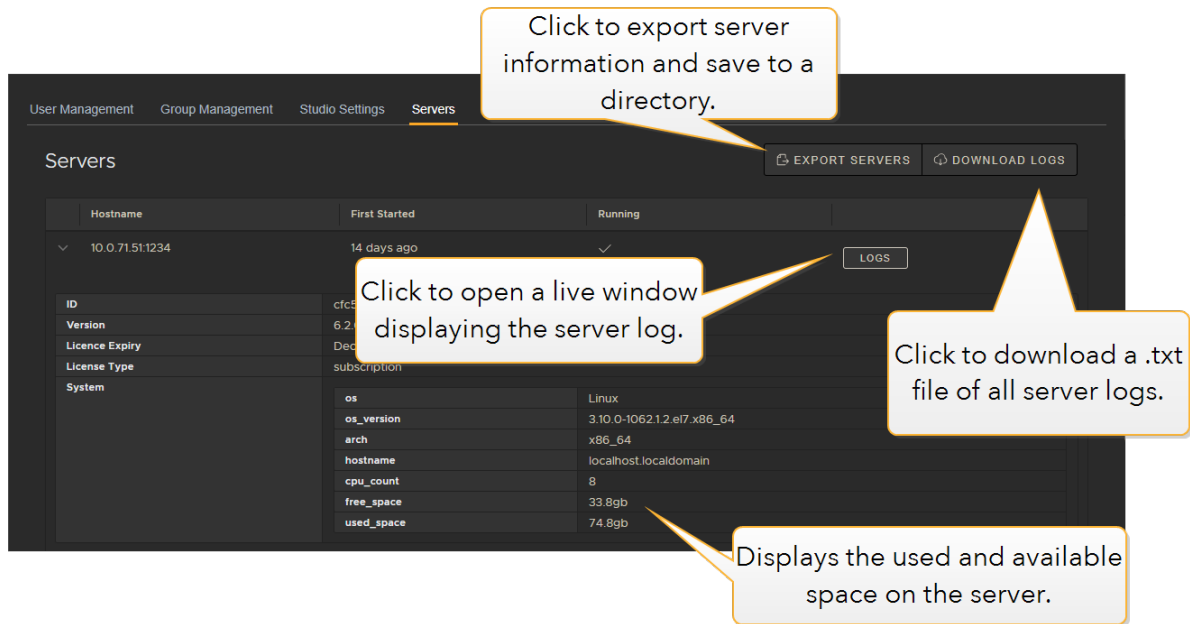
Note: Adding a server to your Flix deployment doesn't affect how end users log in. They can keep connecting to the original Flix Server, or any other in your deployment. Since Flix handles the load balancing, all end users can connect to the same Flix server. If a server is used this way, Flix will share the jobs with all servers in your deployment.



Tip: You can check if a server has been added successfully by going to **Flix > Management Console > Servers** in your Flix client. See [Managing Your Flix Deployment](#) for more information.

Server Management

You can check the list of servers running Flix by going to **File > Management Console > Servers** in your Flix Client. Here you can see the server ID, which version of Flix each server is running and download logs for Foundry support if needed.



Note: Flix stores all temporary files in a directory named 'Flix' followed by a 9 digit random number. This directory is created inside the system temporary directory. You can change the environment setting for TMP, TEMP, TMPDIR (depending on your OS) before starting the Flix server, if you want Flix to use a different location for temporary files. Flix server regularly cleans up files stored in the temp directory.

Migrating Assets When Switching to a New Assets Directory

After changing the asset directory option in the config.yml file (see [Setting a Custom Assets Directory](#)), you will need to migrate the data from the original assets directory to the new one.

Assets directories are named with their server identification, for example '8c17bef2-2fd9-439b-a5cf-8a1b082ee9d3'. To migrate your assets from their previous assets directory, run the server using the **--import-from** flag, pointing Flix Server to the old assets directory to import data from. Flix Server imports everything from the specified old assets directory to the new one now specified in the config.yml file by the **asset_directory** option.

For example, if the previous asset directory was '/mycompany/assets/directory' and your server ident was '8c17bef2-2fd9-439b-a5cf-8a1b082ee9d3', you would use the following command:

```
./flix_server --import-from /mycompany/assets/directory/8c17bef2-2fd9-439b-a5cf-8a1b082ee9d3
```

Your assets are copied to the new asset directory. If files already exist in that directory, they are not copied to avoid duplication.

Command Line (CLI) Arguments

For a complete list of Command Line Arguments, simply run Flix Server with the **--help** flag. For example:

```
./flix_server --help
```

Installing and Launching Flix Client

Flix Client System Requirements



Note: Flix Client is currently not available on Linux. Other operating systems may work with Flix Client, but have not been fully tested.

MacOS

Item	Requirements
Operating System	• 13.x (Ventura) • 14.x (Sonoma) • 15.x (Sequoia)
RAM	4 GB
Processor	ARM-based Apple Silicon processor or Intel processor with 64-bit support 2 GHz or faster processor with SSE 4.2, or later



Article: For more information on Foundry products and supported macOS versions, see Foundry Knowledge Base article [Q100592](#).

Windows

Item	Requirements
Operating System	• Windows 10 • Windows 11
RAM	4 GB
Processor	A 64-bit processor @ 2 GHz, or higher

Installing and Launching Flix Client

To download and install Flix Client:

Windows:

1. Download and unzip the **.zip** file located [here](#).
2. In the unzipped folder, double-click or open **Flix.exe**.

Mac:

1. Download the **.dmg** file located [here](#).
2. Double-click the **.dmg** file to open its content.
3. Drag the Flix application into your **/Applications** directory to install.



Note: You can also download the Flix client for your operating system directly from your Flix server using the following URL:

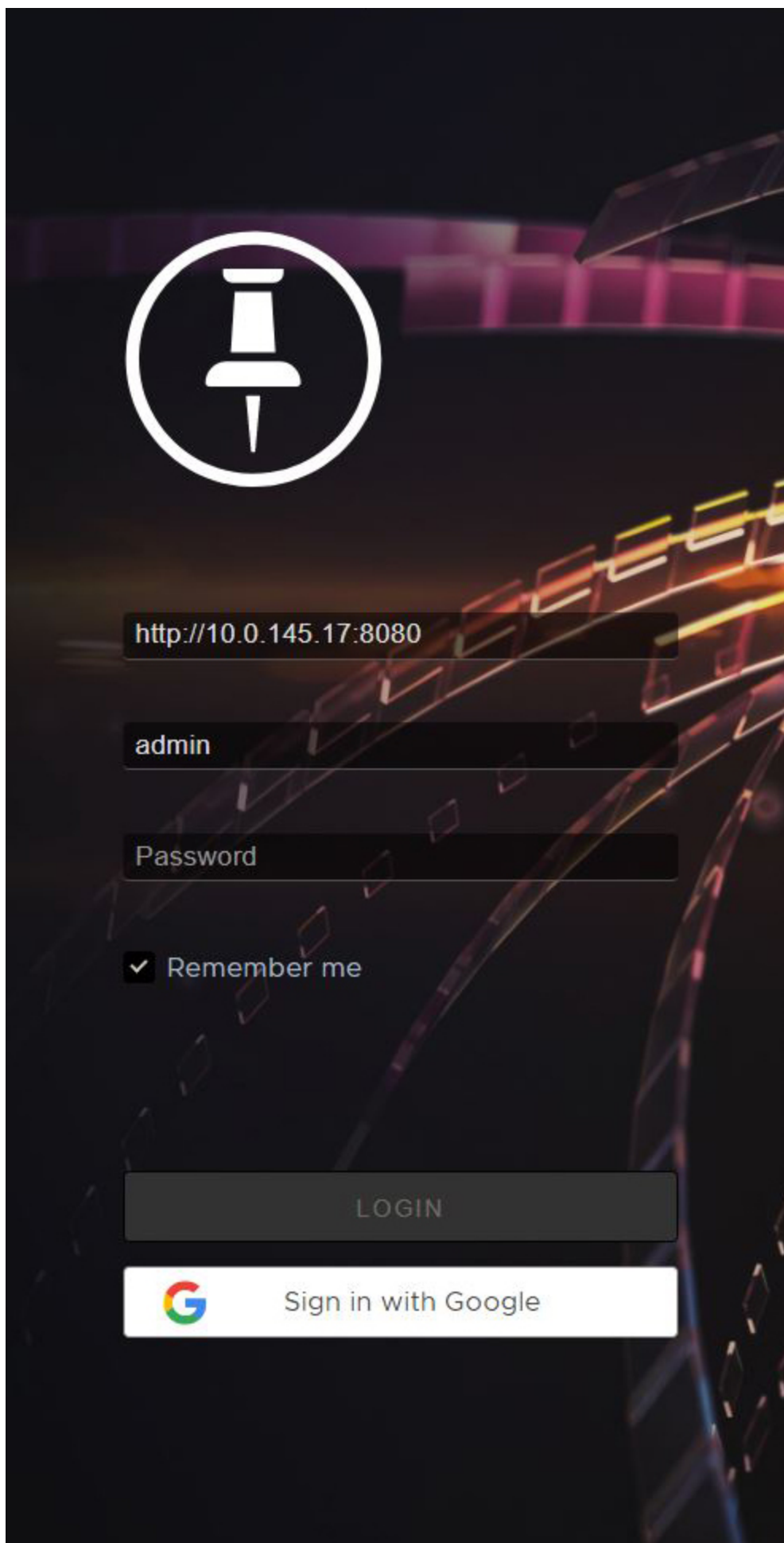
`http://[my-flix-server]:[port]/download`

For example:

`http://flix-server-1:8080/download`

The following steps guide you through getting up and running with the Flix Client desktop application.

1. Double-click the Flix application to open the login page.



2. In the **Server Hostname** field, enter the server address. These credentials can be obtained from your System Administrator. See [Installing Flix Server](#).



Note: Flix remembers any servers that have been successfully connected to, for the next time you log in. Click the **x** next to a hostname to remove the server from the list.

3. Log in using one of the following authentication methods:

Flix User Management- Enter your **Username** and **Password** then click **Login**.

LDAP - Enter your **Username**, and **Password** then click **Login**.

Oauth - Click **Sign in with Google**.

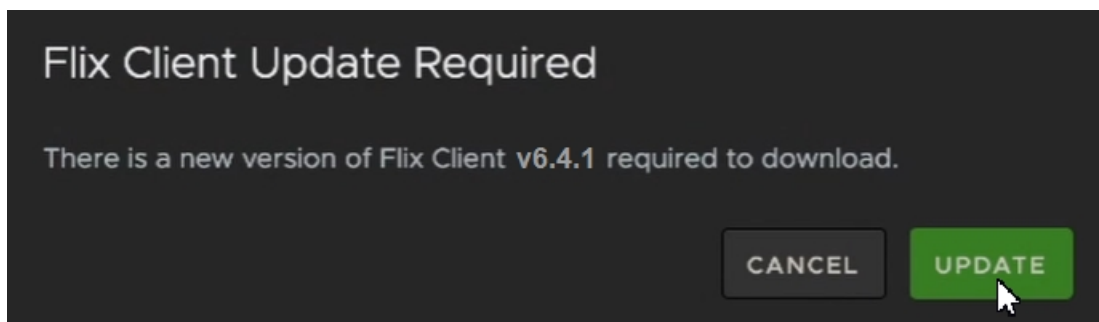


Note: For more information on authentication methods refer to [Setting Up User Authentication](#).



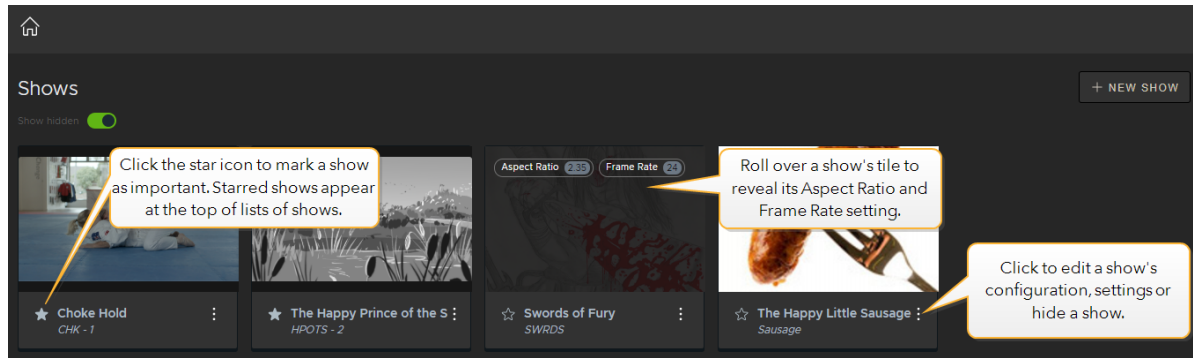
Tip: Check the **Remember me** box to make your next login easier.

The Flix Client and Flix Server versions must match to launch Flix. If your client version is older than the server version, you're prompted to update your client automatically.



Logging in successfully opens Flix at the **Shows** level.

From here you can open an existing show, create a new one or set user preferences for shows.



Article: If you are forcibly logged out and receive an authentication error, it's possible your client machine's date, time or timezone setting is not synced with Flix Server's. To learn more, take a look at the Knowledge Base Article [Why Flix's security protocol may forcibly log users out](#).

Setting Up the Client for HTTPS Connection to the Server

When you set up the Flix environment to use HTTPS, this needs to be set up manually on each client machine for Flix users to connect to the server.

- If your client machines are using TLS certificates **signed by a Certificate Authority (CA)**, your client machine only needs internet access for Flix Client to connect to Flix Server.
- If, however, your client machines are using **self-signed TLS certificates**, your CA certificate needs to be installed on each client machine.

If your environment has multiple Flix Servers set up to use HTTPS, you only need one CA certificate.

Go to [Setting Up HTTPS](#) for more information on setting up Flix Server with HTTPS.

Running Flix in a Web Browser (Beta)

Not all collaborators may have Flix installed or may not be able to access a machine that does, so you can run Flix Client using a web browser. Everything Flix Client does, with the exception of using extensions and plugins, is possible in a browser window.

To allow your Server to accept connections from a web browser, add the following config option (false by default):

webui:

enabled: true

For security purposes, the Flix web interface only works with HTTPS. See [Setting Up the Client for HTTPS Connection to the Server](#)



Note: You can also log in through HTTP but this will limit what you can do in Flix. For example, you can move existing panels, change color tags and manage your shows and Flix deployment but you cannot add any assets or export, pitch or annotate.



Note: Flix supports both self-signed certificates and CA.

To access Flix in a browser window, enter the same hostname that you would in the Flix Client to login: **https://flix-url:8080**.

The web interface of Flix functions the same as the Client, but with the following exceptions:

- Extensions will not work with the web browser (cannot send to or open from Flix)
- Reconfirming to Storyboard Pro from Flix or importing from Storyboard Pro to Flix will not work in the web browser
- The Dialogue workspace is not available
- Log downloads are not available

Setting Up User Authentication

Flix provides three ways to authenticate users: [LDAP \(Lightweight Directory Access Protocol\)](#), [OAuth](#) and [Control User Access with Group and Role Permissions](#).



Note: The first time the Flix Server is run, it automatically creates an admin user with the username and password both set to **admin**, which you can use to log in to Flix for the first time. It is recommended that you change the default password after the first log in. In case the admin user login details are lost, you can reset the admin account username and password to back to **admin** using the **-reset-admin** flag.

Control User Access with Group and Role Permissions

Flix allows you to control who can access shows and functionality using group and role permissions, rather than splitting permissions using admins and regular users. Flix can handle remote artists and collaboration between multiple studios on the same shows, while still providing security for all intellectual property.



Note: You can make admin users with full access to all shows and functionality, but we recommend limiting the number of admin users you create. Using group and role permissions gives you fine-grained control over who can access what in Flix.

Creating a User Account

As an administrator or a user that has specifically granted permissions you can create user accounts and assign them to specific groups and roles.

To create a user account:



Note: Only administrators and users that are specifically granted permissions can create user accounts.

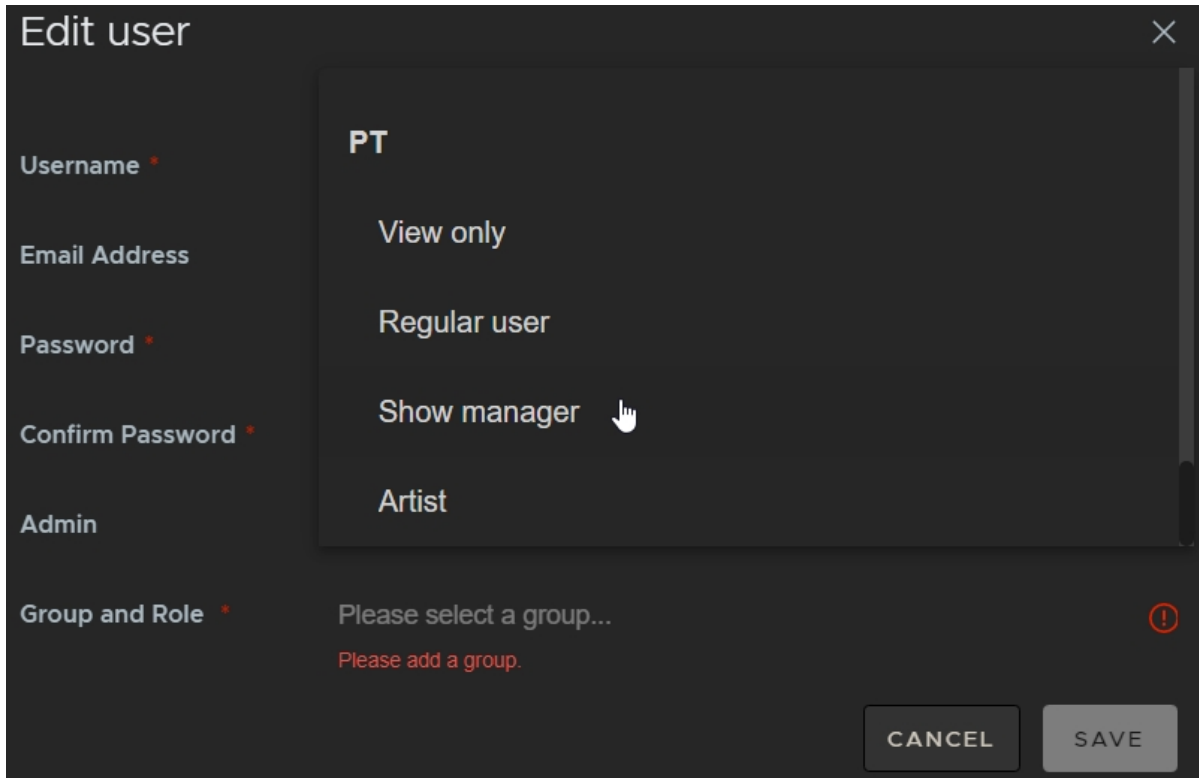
1. Navigate to **File > Management Console**.
2. In the **User Management** tab, click **+New User**.
3. Fill in the **Username**, **Email address**, **Password**, and **Confirm Password** fields.
4. Select whether you want to give the user an **Admin** permissions.

The toggle turns green to indicate the admin permissions.



Note: Only accounts with administrator permissions can create new admins.

5. Select a **Group** and **Role** within the group to assign permissions to the new user. In this example, **PT** is the group name and **Show manager** is the role name.



You can add as many groups and roles as you like to a user.



Note: To create a group, see [Creating Groups and Assigning Roles to Manage Permissions](#).

6. Click **Create**.

The new user account is added to the **User Management** table.

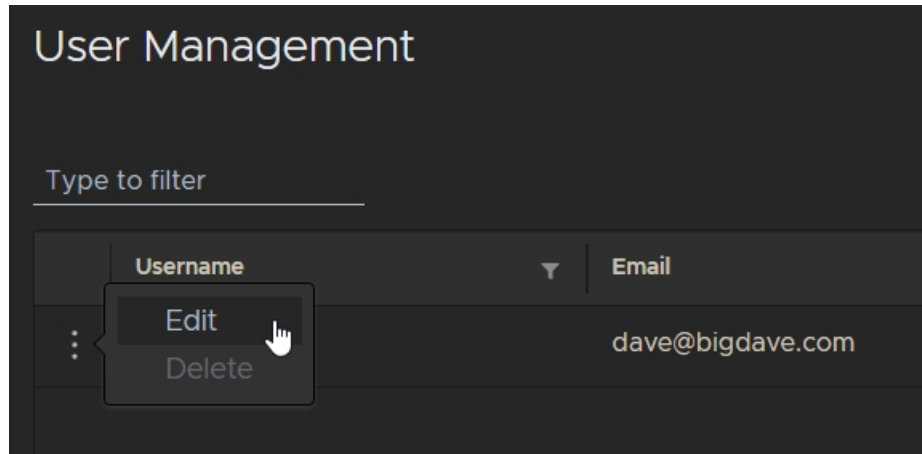
Editing and Deleting User Accounts

To make modifications to existing user accounts:



Note: Only administrators and users that are specifically granted permissions can edit user details.

1. Navigate to **File > Management Console**.
2. In the **User Management** tab, click the more options  button and select **Edit**.



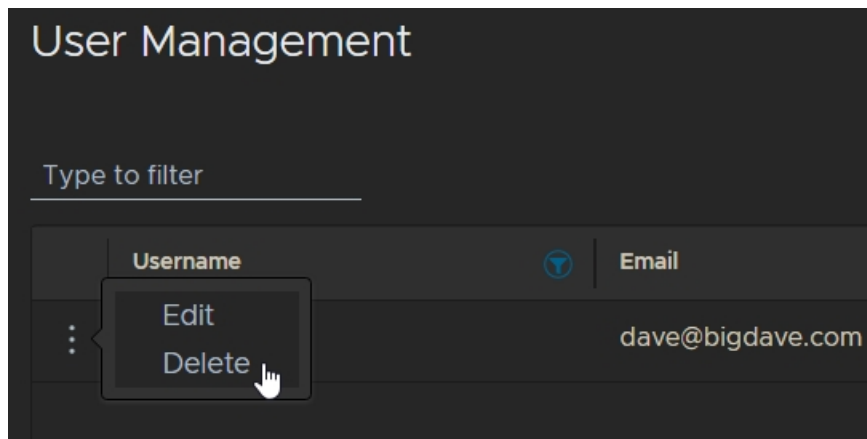
3. Make the required changes in the **Edit user** dialog.

The screenshot shows an 'Edit user' dialog box. It has a close button (X) in the top right corner. The fields are: 'Username' with the value 'dave.mustaine', 'Email Address' with 'dave@bigdave.com', 'Password' with masked dots, and 'Confirm Password' with masked dots. There is an 'Admin' toggle switch which is currently turned off. The 'Group and Role' field shows a dropdown menu with 'default - Show manager' selected and a 'Please select a group...' prompt. At the bottom right, there are two buttons: 'CANCEL' and 'SAVE'.

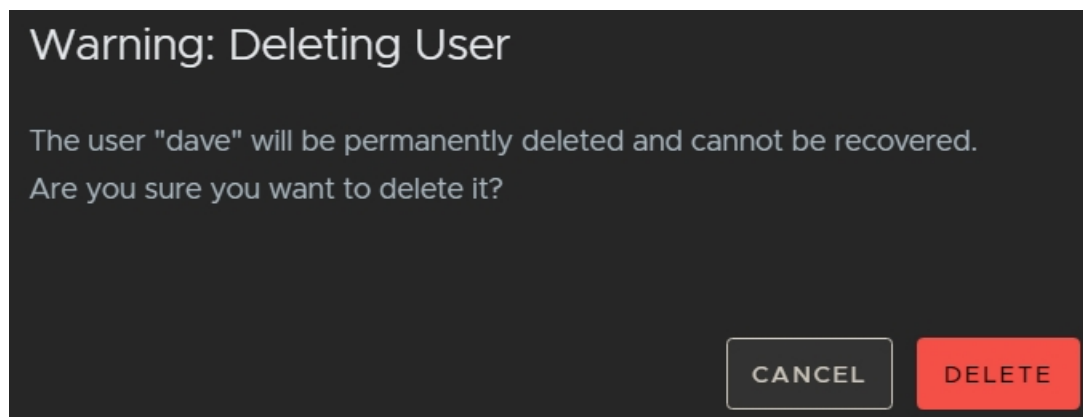
4. Click **Save** to apply the changes.

To delete a user account:

1. Navigate to **File > Management Console**.
2. In the **User Management** tab, click the more options  button and select **Delete**.



3. A warning is displayed because accounts are not recoverable if they are deleted.



4. Click **Delete** to remove the account permanently.

Creating Groups and Assigning Roles to Manage Permissions

You can use groups and roles to limit access to shows. Only some groups can access shows and those groups include certain permissions assigned by roles. For example, you can create a group called **Review** with the role **View only**. You can then add the users responsible for sign off on a show's story.

Create New Group

Group Title * **Review**

Assign Show **Cat and Moth** **Under the Crypt** Please select a show...

Assign Role & User **View only** Please select a role...

View only **katie** **maria** **dave** Please select a user...

CANCEL **CREATE**

In the example, the **Review** group only has access to **Cat and Moth** and **Under the Crypt** with **View only** permissions and the group only contains three users: **katie**, **maria**, and **dave**. This way, only those three users can access the shows and they can't make any edits.

Assign Permissions to a Role

Role permissions determine the Flix functions that a role can access, such as **Contact Sheets**, **Exports**, and **Group** administration. Flix ships with three roles by default, but you can add your own as well. See [Add Custom Roles](#) for more information.



Note: Administrator can also be considered a role, but there is no entry in the **Permissions Management** table for admins because they have access to all shows and functionality. Administrator permissions are assigned as part of user creation and can only be created by other admins. See [Creating a User Account](#) for more information.

To assign permissions to a role:



Note: Only administrators and users that are specifically granted permissions can edit user permissions.

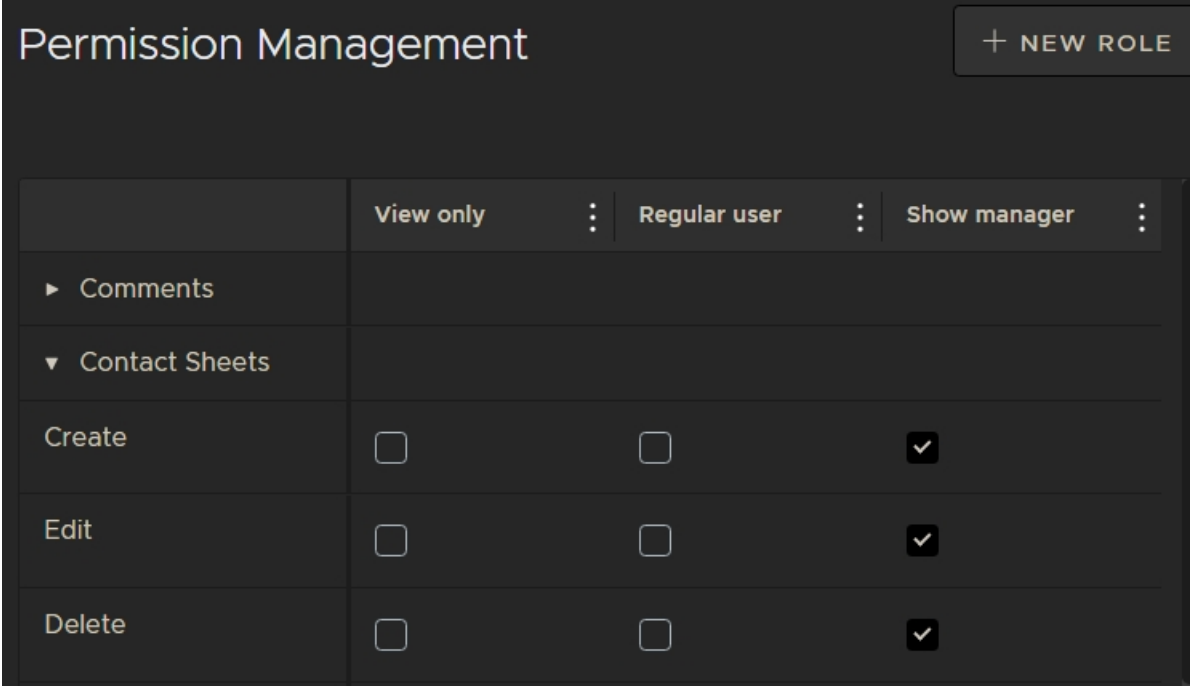
1. Navigate to **File > Management Console**.
2. Click the **Permissions** tab to display the default Flix roles: **View only**, **Regular user**, and **Show manager**.

3. Click the more options  button to edit the role name or delete a role.

See [Add Custom Roles](#) for more details on creating your own roles.

4. Click a dropdown on the left of the table to open up functions for that category. The permission categories determine what each role can access.

For example, clicking the **Contact Sheets** category reveals the **Create**, **Edit**, and **Delete** functions and the roles that have permission to access those functions.



The screenshot shows a 'Permission Management' interface. At the top right is a '+ NEW ROLE' button. Below the header is a table. The first row of the table has three columns: 'View only' with a more options button, 'Regular user' with a more options button, and 'Show manager' with a more options button. The second row has a dropdown menu with 'Comments' selected. The third row has a dropdown menu with 'Contact Sheets' selected. Below this are three rows for permissions: 'Create', 'Edit', and 'Delete'. Each row has three checkboxes corresponding to the roles in the header. The 'Show manager' role has all three permissions checked.

	View only 	Regular user 	Show manager 
► Comments			
▼ Contact Sheets			
Create	☐	☐	☒
Edit	☐	☐	☒
Delete	☐	☐	☒

5. To add or remove permissions to a certain function, toggle the checkbox on or off.

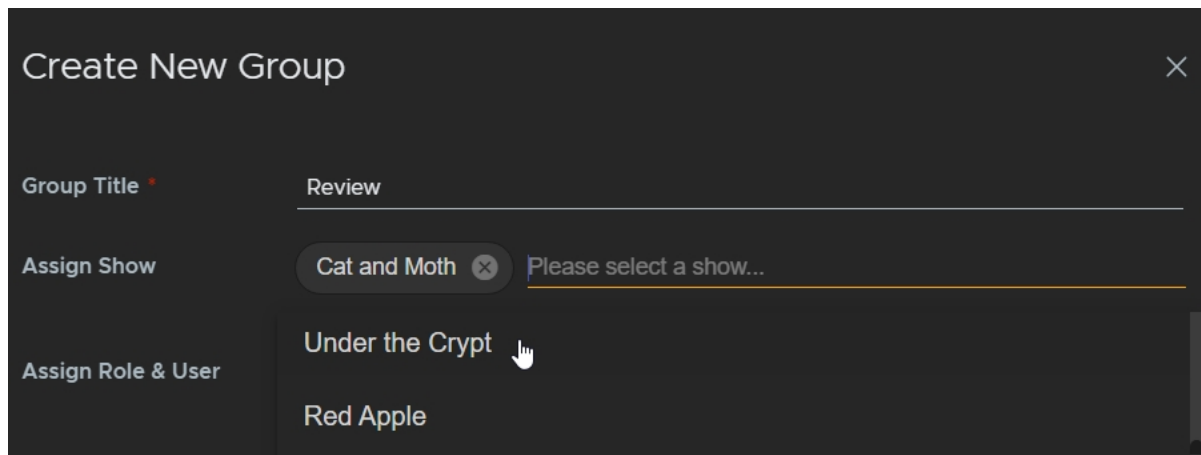
Add a New Group to Control Access

To add a group:

1. Navigate to **File > Management Console**.
2. In the **Group Management** tab, click **+New Group**.

The **Create New Group** dialog is displayed.

3. Enter a name for the group. The name can be anything you like, but a descriptive name helps other users understand a group's purpose.
4. Click **Please select a show** to assign the new group to the shows you want users to access.



Create New Group

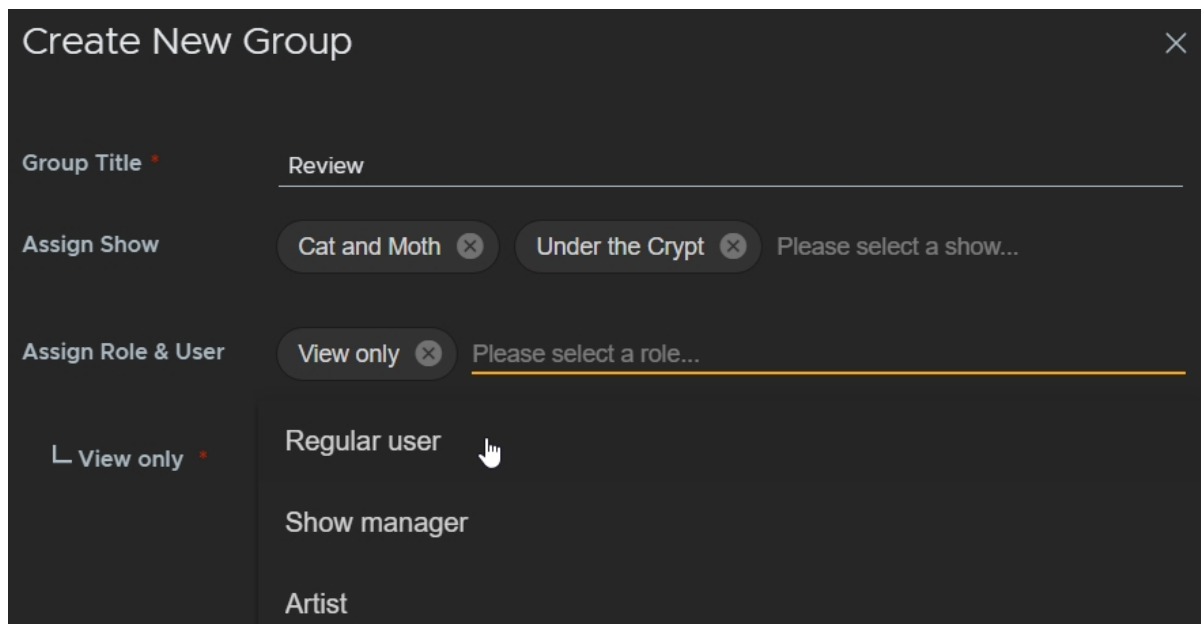
Group Title * Review

Assign Show Cat and Moth Please select a show...

Assign Role & User Under the Crypt Red Apple

You can assign as many shows as you like to a group.

5. Click **Please select a role** to assign roles to the new group. Flix ships with three roles by default, but you can add your own as well. See [Add Custom Roles](#) for more information.



Create New Group

Group Title * Review

Assign Show Cat and Moth Under the Crypt Please select a show...

Assign Role & User View only Please select a role...

View only * Regular user Show manager Artist

6. Click **Please select a user** to assign users to the new group.

Create New Group

Group Title *

Assign Show

Assign Role & User

You can assign as many users as you like to a group.

7. Click **Create** to save the new group in the **Group Management** table.

In the example, the **Review** group only has access to **Cat and Moth** and **Under the Crypt** with **View only** permissions and the group only contains three users: **katie**, **maria**, and **dave**. This way, only those three users can access the shows and they can't make any edits.

Add Custom Roles

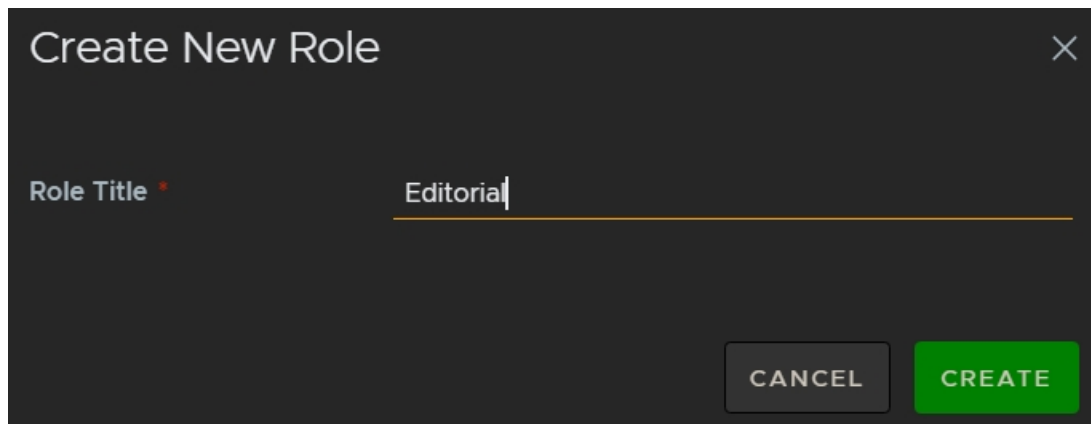
You can create your own roles in addition to the defaults if you need more control over who can access what in Flix.

To add a role:

1. Navigate to **File > Management Console**.
2. In the **Permissions** tab, click **+New Role**.

The Create New Role dialog is displayed.

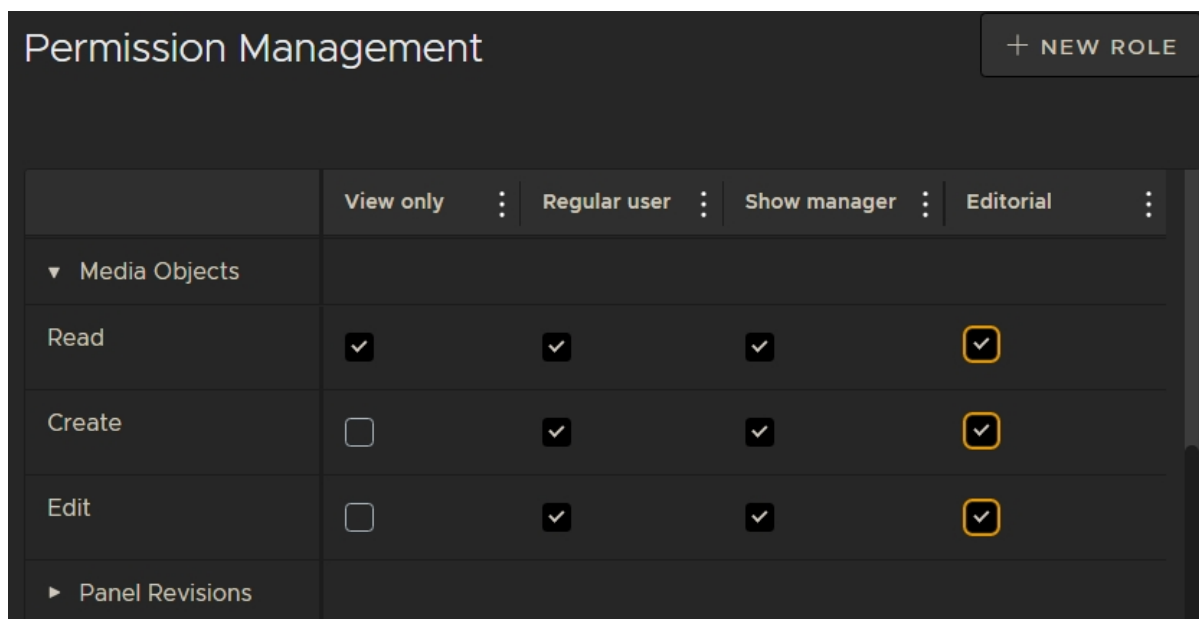
3. Enter a **Role Title** and click **Create** to save the new role.



A dark-themed dialog box titled "Create New Role" with a close button (X) in the top right corner. It contains a text input field labeled "Role Title" with a red asterisk, which has the word "Editorial" entered. At the bottom right, there are two buttons: "CANCEL" and "CREATE".

The new role is added to the **Permissions Management** table.

4. Click the permission categories you want to allow and check all the required boxes for the new role.



A table titled "Permission Management" with a "+ NEW ROLE" button in the top right. The table has columns for "View only", "Regular user", "Show manager", and "Editorial", each with a dropdown arrow. The rows are grouped under "Media Objects" and "Panel Revisions". The "Media Objects" group includes "Read", "Create", and "Edit" permissions. The "Panel Revisions" group is currently collapsed.

	View only	Regular user	Show manager	Editorial
▼ Media Objects				
Read	☒	☒	☒	☒
Create	☐	☒	☒	☒
Edit	☐	☒	☒	☒
► Panel Revisions				

You can now assign the new role to users. See [Creating Groups and Assigning Roles to Manage Permissions](#) for more information.

Single Sign-On

Flix makes secure account provisioning easier by seamlessly integrating with popular Identity providers.



Note: An Identity Provider (IdP) is a system that stores and verifies the identities of users, allowing them to log into various applications and services.

The login page dynamically displays the appropriate login options depending on your Flix server configuration.



SSO login

Depending on your studio's requirements, you can now configure Flix to sync with your IdP using either SAML or OpenID Connect



Note: If your IdP supports multi-factor authentication and it is configured, it is triggered automatically. There is no need for additional setup in Flix.

IdP Settings

To set up SSO, you must include the following settings for your IdP:

- Sign-on URL/callback URL (SAML & OpenID Connect): `http://server-name:port/authenticate/sso/callback`
- (For Okta SAML) Audience URI: `http://server-name:port/saml/metadata`
- (For Okta SAML) An Attribute Statement:

- Name: whatever you specify for sso.saml.email_attr in the Flix config
- Value: user.email

Linking Flix to Your IdP



Note: You cannot enable both Single Sign-On and LDAP/AD in the Flix Server config simultaneously. If your LDAP/AD is connected to an identity provider, we suggest using the new SSO settings for the extra benefits.

To link Flix to your IdP, you need to add the following options to the Flix Server config.yml file:

Link Flix to your IdP

```
# shared SSO options

default_group: default

default_role: Regular user


# SAML options

saml:

  # enable SAML

  use_saml: true

  # the metadata URL for the identity provider

  idp_metadata_url: https://xxx.okta.com/app/.../sso/saml/metadata

  ## optional: the attribute assertion to use for the email address

  ## defaults to http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress

  ## should be specified if you have added custom attribute assertions

  # email_attr: emailaddress

  ## optional: the attribute assertion to use for the display name (not currently used)

  ## defaults to http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name

  ## should be specified if you have added custom attribute assertions

  # display_name_attr: name
```



```
# OpenID Connect options
oidc:
  # enable OIDC
  use_oidc: true

  # the OIDC provider URL
  provider: https://xxx.okta.com

  # the OIDC client ID
  client_id: ...

  # the OIDC client secret
  client_secret: ...

  ## optional: the claim to use for the email address
  ## defaults to "email"
  #email_attr: email

  ## optional: the claim to use for the display name (not currently used)
  ## defaults to "name"
  #display_name_attr: name
```

LDAP (Lightweight Directory Access Protocol)

LDAP/AD (Active Directory) Authentication allows Flix users to log in using their LDAP/AD credentials. The first time a user logs in, Flix obtains the group names they belong to in LDAP/AD and associate those groups with the user in Flix. This allows Flix administrators to add permissions in Flix based on which groups a user belongs to in LDAP/AD.



Article: For more information on setting up LDAP and for troubleshooting tips, please refer to Knowledge Base article [Q100563](#).

Flix reads all configuration options, including LDAP authentication, from a **config.yml** file, stored in the same directory as the Flix Server install by default. See [Installing Flix Server](#) for more information. For guidance on formatting, please refer to the example at the end of this section.

The following **config.yml** attributes are available:

`use_ldap` (optional) - This turns on or off the LDAP authentication method for this server. Values: **true** or **false**.

`base` - The base dn is the point from where a server searches for users in your LDAP/AD. You must supply at least the Domain Component (DC).

`host` - The hostname or IP address of your LDAP/AD server.

`port` (optional) - The port number to be used when connecting to the LDAP/AD server.

`use_ssl` (optional) - This indicates whether or not to use SSL/TLS when connecting to your LDAP/AD server. Values: **true** or **false**.

`bind_user` (optional) - This is an account that binds to the LDAP server and performs user and group searches. It can be a read-only account. Make sure the bind user you want to use has permissions to search through the desired paths. The value of this setting can be in one of the following formats:

`username`

`cn=username,dc=domain,dc=com`

`username@domain`

`bind_pass` (optional) - The password for the name provided in `bind_user`. If you don't use `bind_user`, or if it does not require a password, you don't need to set this.

`self_auth` (optional) - If this is set, `bind_user` and `bind_pass` are ignored. Instead, Flix attempts to use the username and password from the user logging in to bind.

User Search

`dn` (optional) - DN from where to start the search from. If this value is not set the **base** will be used.

`filter` (optional) - Filter to apply when searching the directory. Specify the **objectClass** for your users. The default value is: **(objectClass=organizationalPerson)**

`user_attr` - The attribute to use for the username matching for the authentication. On most AD servers, the default setting is - **sAMAccountName**.

`name_attr` (optional) - The attribute used to return the user's full name. On most AD servers, the default setting is - **displayName**.

email_attr - Defines a custom attribute for the user email address to be retrieved from, other than the default 'mail' attribute. This might be useful in cases when the mail field is used for personal email addresses and the cn field for company email addresses.



Note: It is not currently possible to specify which users/groups should be notified upon Editorial publishes. However, the email_attr option does make it possible to retrieve a different mail attribute for users, which can remain blank in LDAP for those who don't wish to receive notifications for Editorial publishes.

Group Search

dn (optional) - DN from where to start the search from. If this value is not set the 'base' will be used.

filter (optional) - Filter to apply when searching the directory. The default value is empty.

user_attr - The name of the attribute from the user search which can be found in a group attribute such as **member**. Common values are **distinguishedName**, **uid**, **sAMAccountName**.

group_attr - The group attribute that has the same value as the user attribute set above. On most AD servers the default setting is 'member'.

name_attr (optional) - The name of the group. On most AD servers the default setting is either 'name', 'cn' or even 'description'.

group_prefix (optional) - Only groups that start with this string will be added to Flix when a user logs in.

group_suffix (optional) - Only groups that end with this string will be added to Flix when a user logs in.

Auto-assign Group-Role Pairs

LDAP groups can be used to automatically assign group-role pairs. This is done by a new YAML section within the LDAP section roles. See [Example LDAP Flix Config File](#) for more information..

Additionally, four optional fields have been added:

allowed_users - an explicit list of LDAP usernames which are permitted to log into Flix (cannot be set if blocked_users is set).

blocked_users - an explicit list of LDAP usernames which are not permitted to log into Flix (cannot be set if **allowed_users** is set).

required_groups - a list of groups users must have to be permitted to log into Flix, users must have all groups in this list. Cannot be set if **forbidden_groups** is set.

forbidden_groups - a list of groups which if a user has any of, that user will not be permitted to log into Flix. Cannot be set if **required_groups** is set.

Example LDAP Flix Config File



Note: This example is for illustration purposes. The entry preceding the ':' is a key that Flix reads, which needs to be named as in the example, but the entry following the ':' follows the exact naming of the attribute name in your AD.

```
ldap:
  use_ldap: true
  base: dc=flix,dc=ad
  host: 10.10.10.10
  port: 385
  use_ssl: false
  self_auth: false
  bind_user: CN=Flix,OU=Flix-Users,DC=flix,DC=ad
  bind_pass: PASSWORD
  user_search:
    dn: OU=Flix-Users,DC=flix,DC=ad
    filter: (objectClass=organizationalPerson)
    user_attr: sAMAccountName
    name_attr: displayName
    email_attr: description
  group_search:
    dn: OU=Groups,DC=flix,DC=ad
    filter: (objectClass=group)
    user_attr: distinguishedName
    name_attr: name
    group_attr: member
    group_prefix: flix-
    group_suffix: -flix
  roles:
    - role: Show manager
      group_suffix: "Manager"
```

- role: View only
group_suffix: "Readonly"
- role: Regular user
group_prefix: "Flix"
- role: View only # fallback role

OAuth

The OAuth configuration allows users to log in to Flix using their credentials provided by an OAuth service. Using the example below, configure your Flix server config file to use OAuth. Once the configuration is enabled, restart your Flix Server, and users should now be able to log in using the **Sign In with Google** button.

use_oauth - This determines whether to use OAuth authentication or not.

username - This determines whether artists are authenticated by 'username' or 'email' address and is used to create their Flix username. Setting the username parameter to email forces Flix to create unique usernames.



Note: If the username parameter is blank, misspelled, or omitted, authentication defaults to 'username'.

domain - This specifies the domain name for your OAuth provided credentials.

providers - This specifies the OAuth provider.

Example of OAuth:

```
oauth:
  use_oauth: true
  username: email
  domain: my_domain.com
  providers:
    - Google
```



Warning: Please be certain not to have LDAP and OAuth both enabled.

OAuth doesn't currently obtain the user groups from Google like it does for LDAP, so group management for project access needs to be done using Flix's built-in Group Management.



Warning: OAuth is not supported when using HTTPS. Due to the design of OAuth, it requires making a HTTP callback to the Flix Server. With HTTPS enabled, this callback would also need to be via HTTPS, however without external access to your certificate authority, this request would fail. Currently this limitation prevents OAuth and HTTPS from being used at the same time.



Note: Currently the only OAuth provider supported is Google.

Managing Extensions & Plugins

Flix currently includes extensions for Photoshop, Storyboard Pro and Maya, providing powerful support for story development.

Flix's APIs and feature set also support building custom extensions, Take a look at the [General Remote Client API documentation](#) for front-end integrations, or the [Server API docs](#) for back-end integrations.

The Extension & Plugin Manager helps you install, update, or uninstall extensions from your Flix client.

Not only does the Extension & Plugin Manager simplify the process, but it also allows you to manage extension versions independently of Flix versions, decoupling the Flix Client version from that of the Flix extension. This means that you can upgrade or roll back versions of Flix and Flix extensions independently.

- To access the Extension & Plugin Manager select **Extensions > Extension & Plugin Manager** or use the keyboard shortcut **Ctrl/Cmd+.**



Note: Some extensions may require the operating system's admin permission to install. If so, this will be noted on the extension's official documentation.



Installing Extensions

The **Available Extensions** tab shows the list of extensions available from Flix which you can install locally. The list includes the **Extension Name**, the **Extension Version**, and the supported versions of the target third-party application.

- You can see a list of available versions using the **Extension Version** drop-down.
- Click the **INSTALL** button to add the selected extension version to Flix.

The version status is shown as **INSTALLED** if the plugin is active.

Changing Extension Versions

1. Switch versions using the **Extension Version** drop-down to a version that is not installed. The status switches to show the **UPDATE** button.
2. Click the button to install the selected version in place of the existing version.



Note: You can only have one version of an extension installed at once.

Checking Versions

- Click the **Installed Extensions** tab to see active extensions.

The Installed tab also indicates if an extension is **Server Managed**. A server managed extension is one that has been installed using the Local Extension Manager as opposed to a manual install outside of the application.

Viewing Extension Documentation

- You can open the documentation for the extension using the  button.

Uninstalling Extensions

- Click the **Installed** tab to see a list of active extensions.
- From here, use the **UNINSTALL** button to remove and deactivate the extension.

Automated Server Polling for Extension Updates

To keep up with the latest Flix updates, you may want to use **automated server polling** for your extensions.

This feature is enabled by **default**, and can be configured via the **config.yaml** file required by the server.



Note: This is the common location for all server configuration within Flix.

By default, the extension system within the Flix server will poll a Foundry hosted URL for new extensions. If the Flix team have released updates to the extensions, the server will fetch them from this URL automatically, making them available for installation on the client.

To change these settings:

- Open the Flix Server **config.yaml**
- Find the section named **extension**. Within this section, find the field **http_url**, which has the default value (if not set): **https://extensions.flixaccess.com/extensions.json**
- To turn polling **off**, you can set **http_fetch_enabled** to **false**, and the extension service will not poll the URL. To keep polling **on**, make sure **http_fetch_enabled** is set to **true**, and the extension service will make an HTTP GET request for an up-to-date list of extensions every hour. You can then choose which extensions to install.



Note: For further information, refer to Flix's **config.example.yaml** file which is distributed with the server.

Manually Adding Extensions to the Server

If you have turned polling off and **http_fetch_enabled** is set to **false**, the extension service will not poll the URL, and you will need to add extensions to the server manually.

To do this:

- Place the extensions files in the **thirdparty/extensions** directory, relative to the **flix_server** binary. The server may create this automatically, but it can be created manually if missing.
- Once this is set up, the Flix Server will check that directory for new extensions once every minute.



Note: When working with multiple servers, ensure shared storage is configured so that all servers have access to the extension files. Read more at [Setting up Shared Storage](#).

Plugins

You can use Flix Client Plugins, which use the [Chrome Plugin framework](#), to have direct control over UI customization and behavior.

You can:

- **Add Custom Buttons:** Introduce new buttons that trigger scripts, enabling studio-specific workflows and automations directly from the Flix Client.
- **Remove Flix UI Buttons:** Restrict access to specific workflows or disable error-prone actions to streamline operations and reduce User mistakes or sanctioned workflows.
- **Modify UI Fields & Behaviors:** Customize UI behavior—for example, pre-populating the hostname on the login page for easier access.

Flix provides example plugins for:

- Button removal
- UI modifications for pre-populating the hostname on the login page

You can find these examples in the Flix [Public Examples Repository](#).

Plugin Deployment

Plugin deployment is simple, just place the plugin files in the local Flix Client Plugins folder, and they'll load automatically on the next launch:

- Windows: %APPDATA%/Flix/plugins

- macOS: ~/Library/Application Support/Flix/plugins

For added flexibility, you can specify additional plugin folder locations—either locally or on network storage—using the `FLIX_PLUGIN_DIRS` environment variable. This variable accepts a semi-colon-separated list of paths to support multiple plugin sources.

Managing Plugins

To manage plugins from within Flix Client, go to **Extension & Plugin Manager > Plugins**.

From here you can:

- Enable / Disable plugins using the available buttons.
 - After enabling/disabling a plug in, a **Reload Windows** button will appear. Reload to update changes.
- Open the folder where the plugins are stored using the **Default Plugin Directory** button. Note that this will only open the default directory.

Safe Mode

A new Safe Mode option at startup allows plugins to be temporarily disabled at launch, so you can troubleshoot easily by ruling out any potential plugin issues.

To open up flix in safe mode, launch from the terminal with a specific flag '**--safe-mode**'. Alternatively, set the `FLIX_SAFE_MODE` environment variable to **true**.

When Flix is in safe mode, it will not check for plugins or load them. Flix Client will display text at the top of the window indicating it is in safe mode.

Flix User Guide

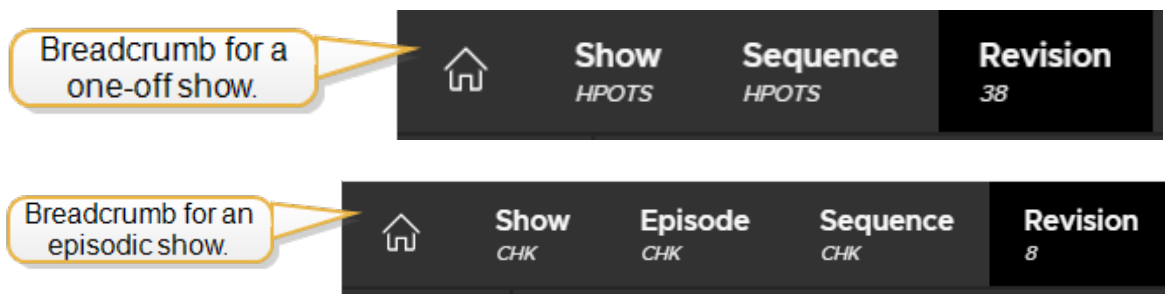
Getting Started

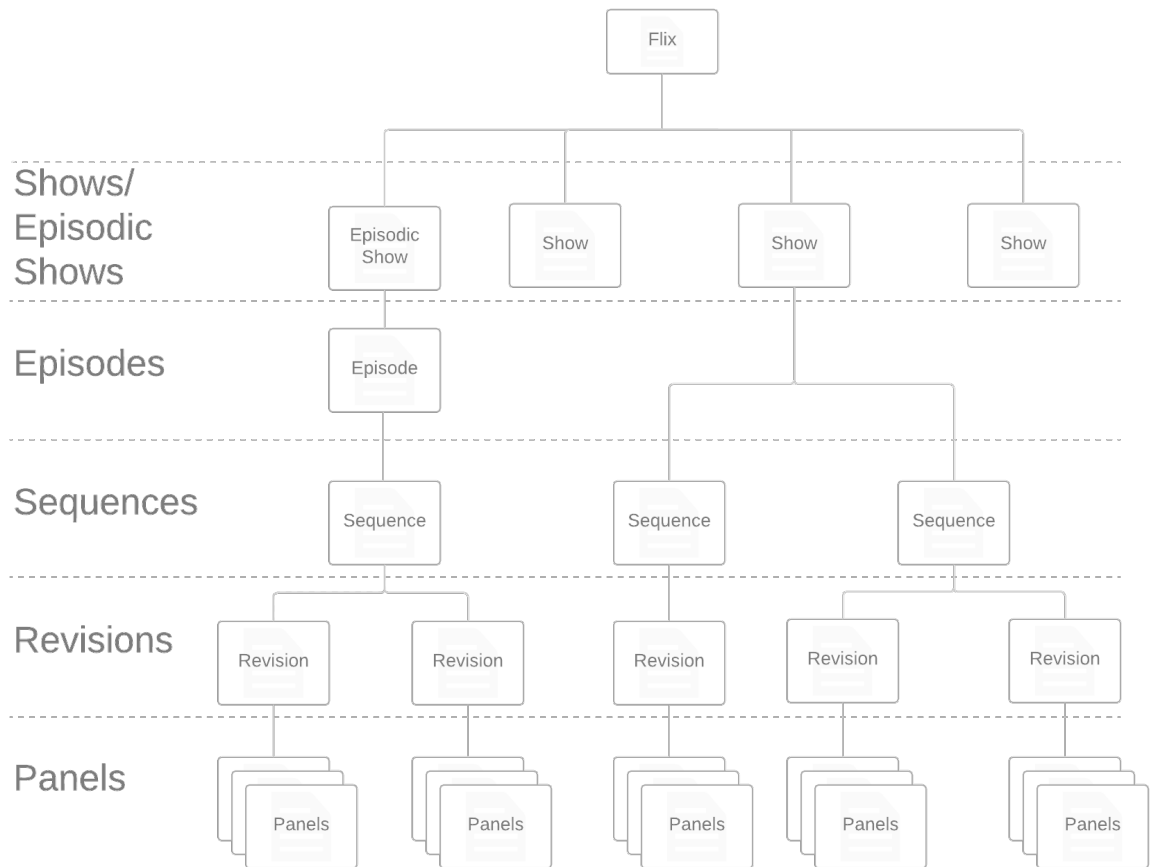
Copyright © 2025 The Foundry Visionmongers Ltd.

Flix is a story development hub for animated film and TV, gaming, and other visual narrative mediums. Watch the video below for a brief overview of how Flix works.

Navigating Through Flix

When you first log in to Flix, the **Shows** level displays the shows you are assigned to. The diagram below shows the hierarchy of how shows are organized along the breadcrumb. The breadcrumb is used to navigate back and forth through the levels of a project.



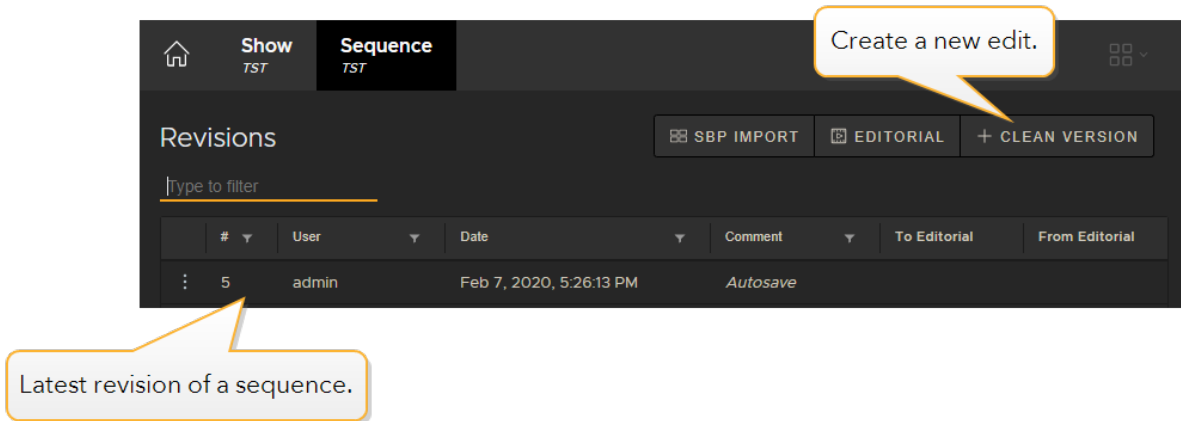


Loading an Existing Sequence

To open an existing sequence:

1. At the **Shows** level, click on the show to which you have access.
2. At the **Sequences** level, the most recent edit, or sequence revision, is always at the top of the list. If the list is long, you may want to use the filter to search for a specific comment. For example, "George's version".
3. Click on the revision to open it.

The image below shows the **Revisions** level of a show. Every revision is listed on this page, with the latest always at the top. Use the filter to narrow down the list by comments that contain specific terms.



Creating a Clean Version of an Edit

To create a brand new edit, or sequence revision, from scratch:

- At the **Revisions** level, click **Clean Version**.

A blank Panel Browser opens. Here you can import new image files to start a new sequence revision.

Creating a Show

When you log in to Flix, you start at the **Shows** level. This is where you can open existing shows or create a new one.

The video below details how to set up a new show.

In the video:

- Setting up a new Show
 - Setting up Flix with Photoshop
1. Click the + **New Show** button to create a new show.
 2. Fill in the **Details** as described in the table below.

3. Click **Create**.

Your new show is added at the **Shows** level.

Details

Property	Description
Tracking Code	This information is used to keep track of shows.  Note: The Tracking Code is mandatory and must contain between 1 and 10 characters.
Title	Input the working title of your show here.
Description	A short paragraph description of your show, which can be viewed at the Shows level when you hover over the show's thumbnail.
Preview Image	Adds a thumbnail image for your show, which can be viewed at the Shows level.  Note: You can use .jpg, .gif, .png files. The maximum resolution is 800 x 800 pixels.

Configuration

Property	Description
Frame Rate	Sets your show's frame rate. Choose from the common film and television frame rates up to 30fps.
Episodic	Toggles whether your show contains episodes or not. This exposes a new option to set the Season number.

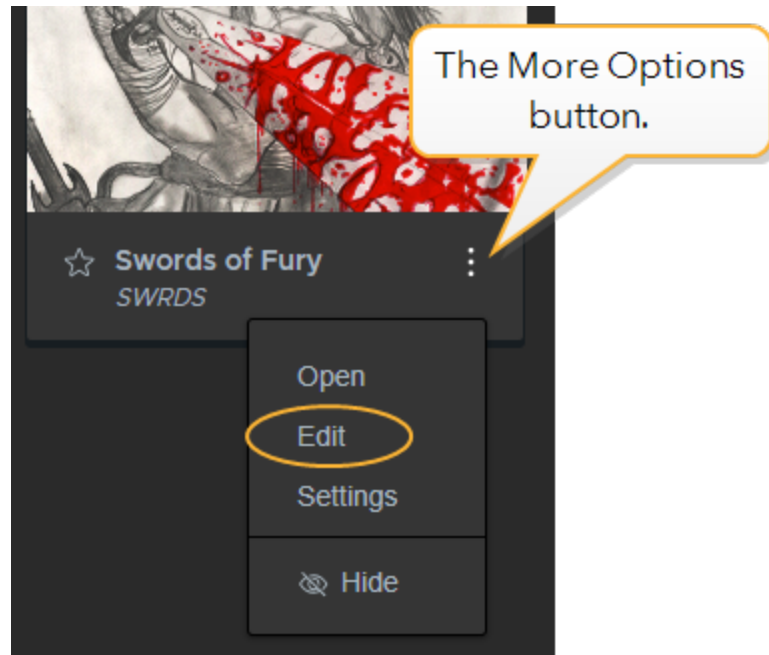
Property	Description
Aspect Ratio	Sets your show's aspect ratio. Choose from the following ratios: 1.0:1 1.77:1 1.85:1 2.0:1 2.2:1 2.35:1 2.39:1

Permissions

Property	Description
Groups	Specifies the groups of users who can access this show. For more information on creating groups, please refer to Creating Groups and Assigning Roles to Manage Permissions .

Editing a Show

1. Navigate to the **Shows** level in the breadcrumb, click the kebab menu  and select **Edit**.



This opens the **Edit Show** dialog.

2. Here you can edit the **Details**, **Configuration** and **Permissions** for your show. For example, use this menu to change the frame rate or access permissions for a user group.



Warning: Changing the frame rate and/or aspect ratio of a show once story artists have begun work is not recommended, due to the fundamental impact on your production pipeline.

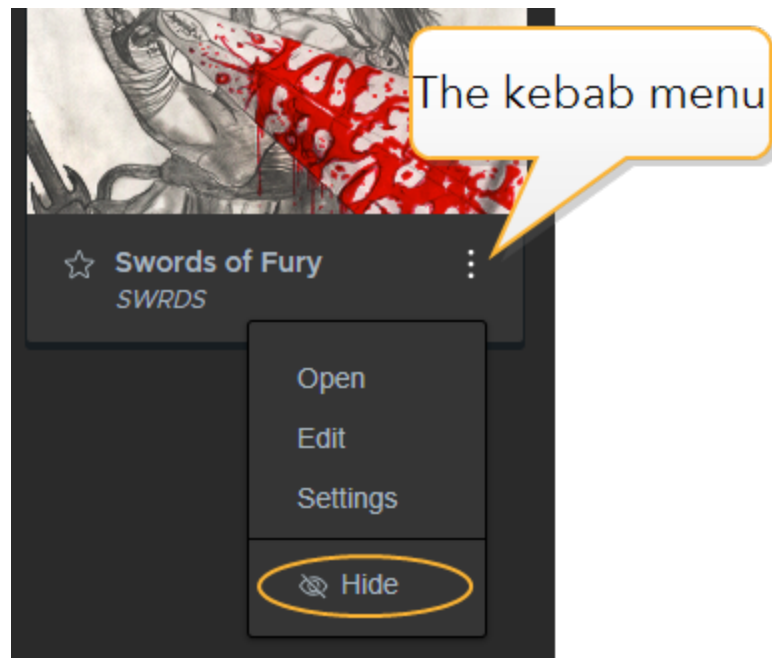
3. Click **Update** to save your edits.

Hiding a Show

Admin users and users that are specifically granted permissions can choose which shows are displayed at the Shows level.

To hide an existing show:

1. Navigate to the **Shows** level in the breadcrumb, click the kebab menu  and select **Hide**.



The show is invisible at the Shows level.



Note: Users with access to a show can still view it if it is hidden by toggling the **Show hidden** button on. This button is set to off by default. To completely disable access to a hidden show for a user, you must remove the user from the relevant permission group(s). See [Control User Access with Group and Role Permissions](#) for more information.

2. To unhide the show, click on the kebab menu  and select **Unhide**.

Starring a Show

If you have access to lots of shows, it can be useful to mark specific ones so they appear at the top of lists. Flix allows users to star shows and remembers this setting on a per-user basis, meaning if you mark a show as starred it will only appear starred for you.

To star a show:

1. Navigate to the **Shows** level in the breadcrumb.
2. Click the  icon on a show's tile.

The show is now marked important and appears at the top of your show list.



Creating an Episode

Episodes are shown at the first level down from Shows in the breadcrumb.

1. In the breadcrumb, navigate to the **Shows** level and open your show.



Note: If your show does not contain any episodes, a menu for creating a new episode opens here.

2. Click **+ New Episode**.

The **New Episode** menu opens. Enter the details as listed in the table below.

3. Click **Create**.

You can continue creating additional episodes in this window. Click **Close** when finished.

Your new episode/s are listed at the **Episodes** level.

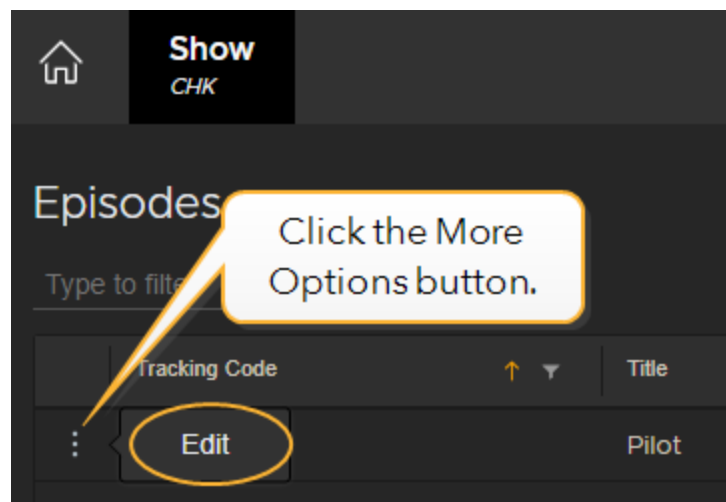


Note: Open an episode and click **+ New Sequence** to create sequences, then follow the steps above.

Property	Description
Tracking Code	This information is used to keep track of shows.  Note: The Tracking Code is mandatory and must contain between 1 and 20 characters.
Title	Use this to name your episode. For example, "The Big Wedding".
Episode Number	Enter the episode number. For example, a common naming convention for is 101, 102 etc.
Description	Here you can write a brief synopsis of the episode.
Comments	Enter additional comments here. These can be viewed at the Episodes level in the Comments column.

Editing an Episode

- Using the breadcrumb, navigate to the **Episodes** level and click the kebab menu , then Select **Edit**.



This opens the **Edit Episode** dialog.

- Enter the new information for your episode.
- Click **Update** to save your edits.

Hiding an episode

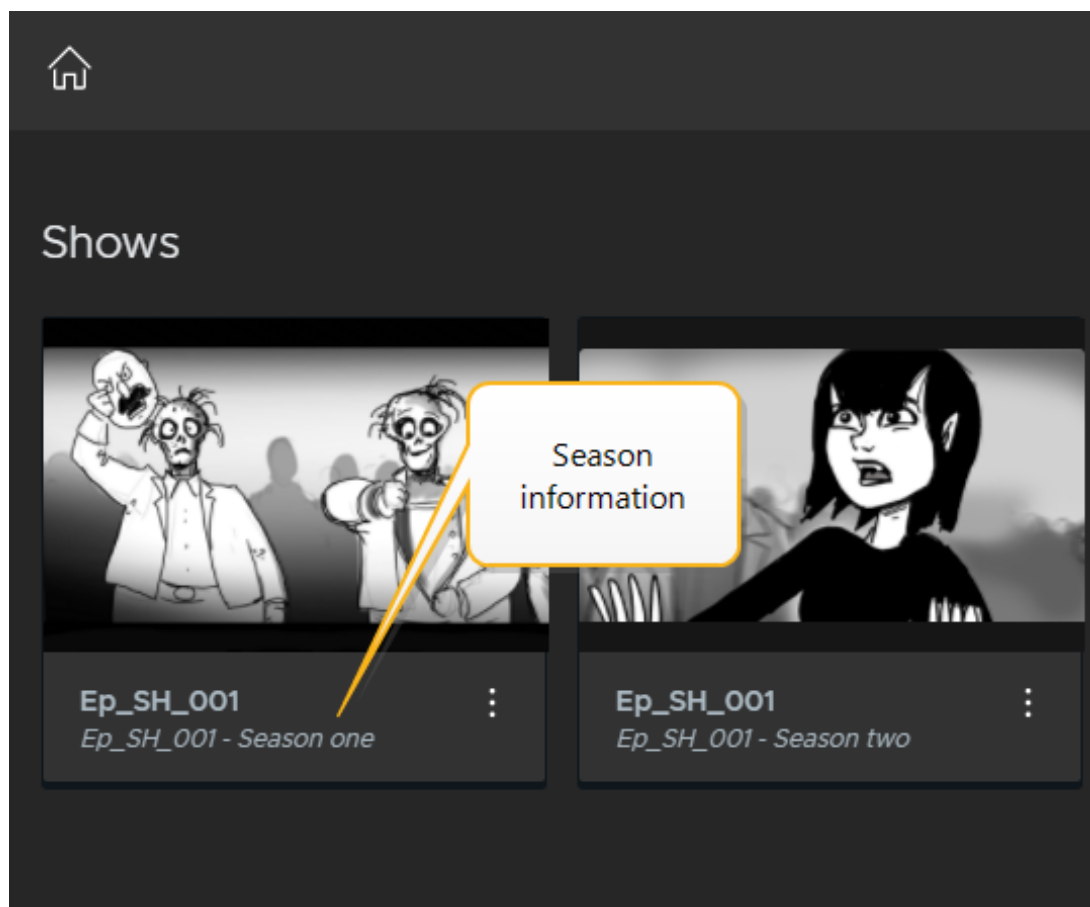
- Navigate to the **Episodes** level and click the kebab menu , then select **Hide**.
- Switch the **Show Hidden** toggle on in the UI to display hidden shows. From this view you can click the kebab menu , then select **Unhide** to restore the show's visibility.

Creating Additional Seasons

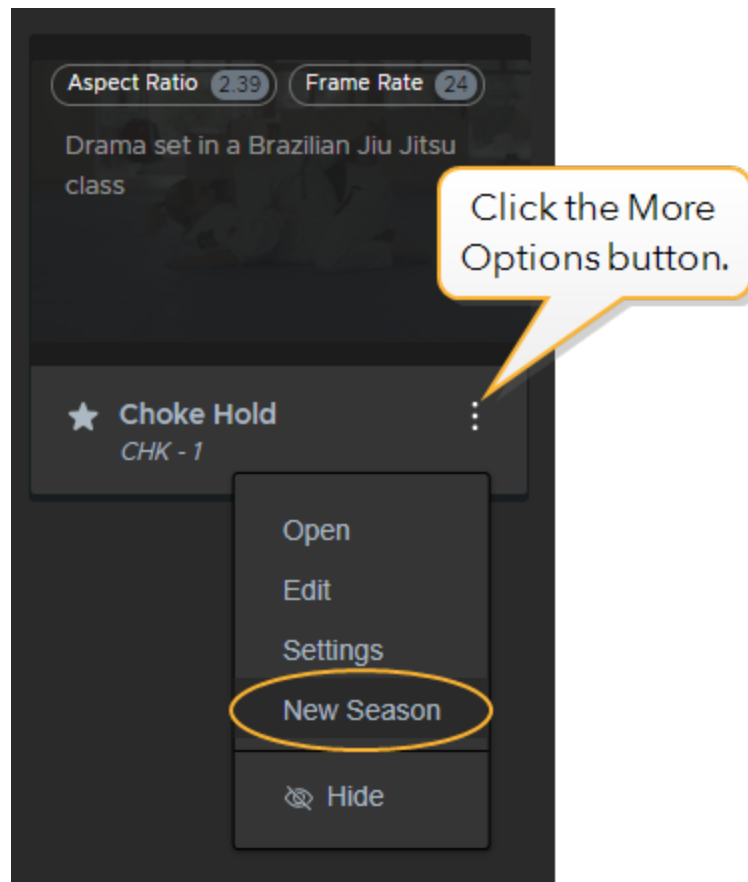
A season of a show is stored at the **Shows** level and contains a number of episodes.

To create additional seasons:

1. Using the breadcrumb, navigate to the **Shows** level and select the required season. The season information is displayed under the thumbnail of your show.



2. Click the kebab menu  and select **New Season**.



This opens the **Create Show** dialog. The details are pre-filled with the information you provided for the previous season.

3. In the **Details** section, you can add a thumbnail for the new season.
4. In the **Configuration** section, you can change the **Season** information. You can also change any other information if required. Click **Next**.
5. In the **Permissions** section, update the **Groups** if needed, then click **Create**.

Your new season is added at the **Shows** level.

Creating a Sequence

Sequences are shown at the first level down from Shows in the breadcrumb.

1. In the breadcrumb, navigate to the **Shows** level and open your show.



Note: If your show does not contain any sequences, a menu for creating a new sequence opens here.

2. Click + **New Sequence**.

This opens the **New Sequence** menu. Edit the values as listed in the table below.

3. Click **Create**.

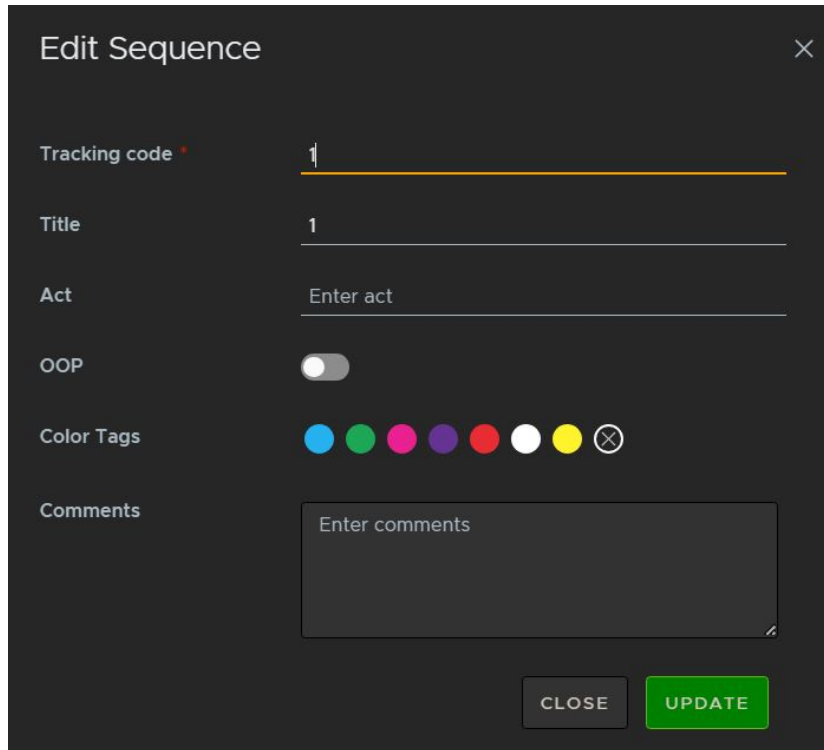
You can continue creating additional sequences in this window. Click **Close** when finished.

Your new sequences or episodes are listed at the **Sequences** level.

Property	Description
Tracking Code	This information is used to keep track of shows.  Note: The Tracking Code is mandatory and must contain between 1 and 20 characters.
Title	Use this to name your sequence or episode. For example, "Wedding montage".
Act	Specifies the act of the story in which your sequence occurs.
Comments	This description can be viewed at the Sequences level in the Comments column.

Editing a Sequence

1. Using the breadcrumb, navigate to the **Sequences** level and click the kebab menu , then Select **Edit**.

The image shows a dark-themed 'Edit Sequence' dialog box. It has a title bar with a close button (X) in the top right. The dialog contains several input fields: 'Tracking code' with a red asterisk and the value '1'; 'Title' with the value '1'; 'Act' with the placeholder 'Enter act'; 'OOP' with a toggle switch currently turned off; 'Color Tags' with a row of seven colored circles (blue, green, pink, purple, red, white, yellow) and a clear button (X); and 'Comments' with a text area containing the placeholder 'Enter comments'. At the bottom right, there are two buttons: 'CLOSE' and 'UPDATE'.

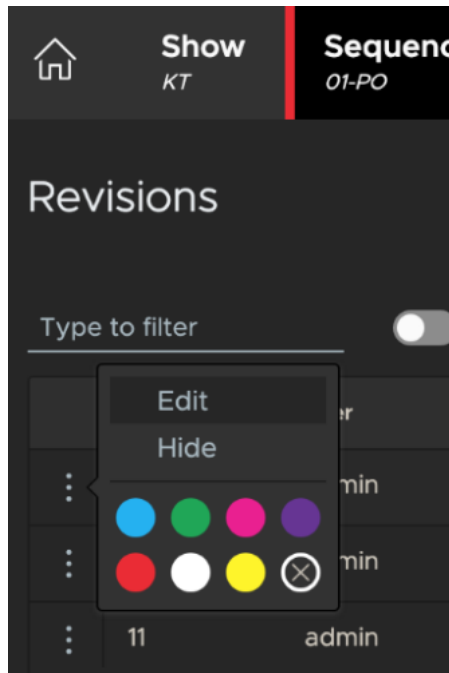
This opens the **Edit Sequence** dialog.

2. Enter new information for your sequence or choose a color tag for it.
3. Click **Update** to save your edits.

Color Tagging a Sequence

You can assign colors to Sequences to indicate ownership, state, purpose or use. For example the color green may be used to indicate sequences that have been signed off.

- Navigate to the **Sequences** level and click the kebab menu  on a sequence, then select a color.



- Select the  button to remove a color tag.

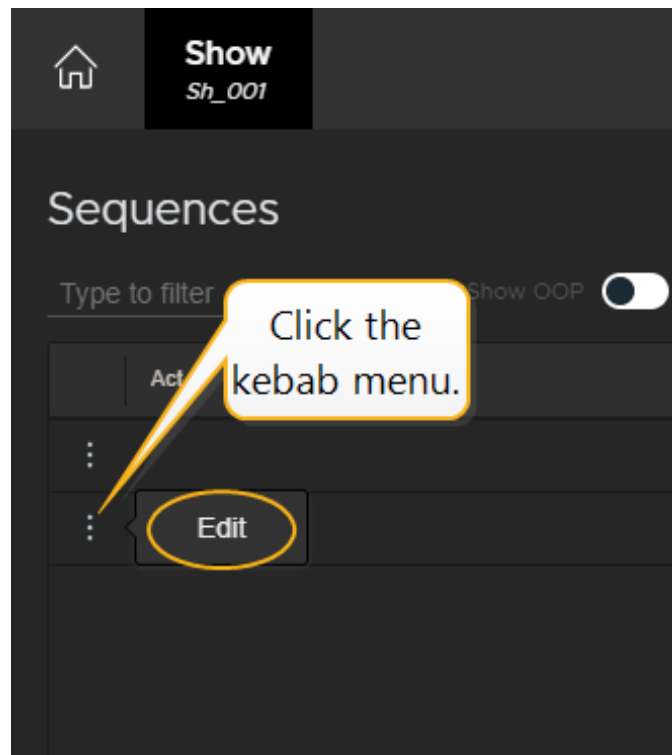


Note: Color selection is also available when hovering over the **Sequence** tab in the breadcrumb at the top of the user interface.

Hiding a Sequence

You may want to hide a sequence from your list if it has been cut from the story or shelved for later use. To do this:

1. Navigate to the **Sequences** level, click the kebab menu  on the required sequence and select **Edit**.



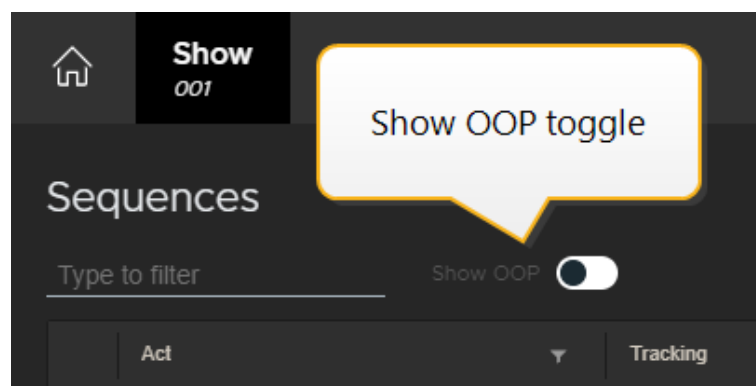
This opens the **Edit Sequence** menu.

2. Click the **OOP** (Out Of Picture) toggle button.

The sequence is now hidden from the list.

3. Click **Update** to save this setting.

At the **Sequences** level, toggle the **Show OOP** button to reveal hidden sequences.



Keeping Edits Organized

To keep your edits organized, you can filter the **Comments** column at the sequence level by keyword.

For example:

1. Go to the **Sequence** level of your show.
2. Find a sequence revision at random in the list, then click on the kebab menu  and select **Edit**.

Type in the comment "Use for screening" and click **Save**.

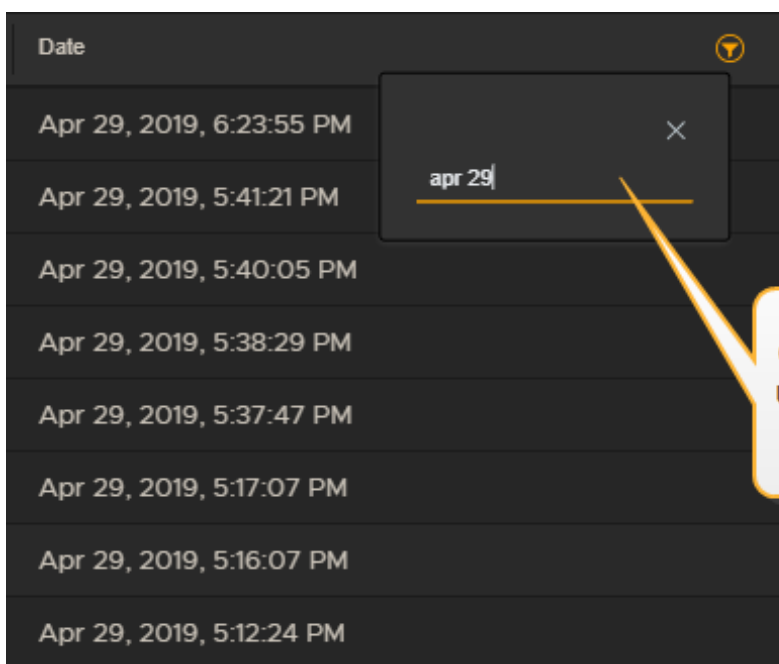
3. Above the Revisions list, type the word "screening" into the filter.

Only sequence revisions with comments containing that word are now displayed. This is a handy way of organizing a long list of sequence revisions.



Tip: If you are on the Panel Browser and make any change to a revision, such as adding dialogue or annotations, you can also add a comment and after saving it will be filterable at the Sequence Level.

The **User**, **Date** and **Comments** columns each have their own filters, so you can narrow down your search results even further. See below.



The screenshot shows a table with a 'Date' column. A filter dropdown is open for the 'Date' column, showing a search input with 'apr 29' and a close button. The table lists several dates from April 29, 2019.

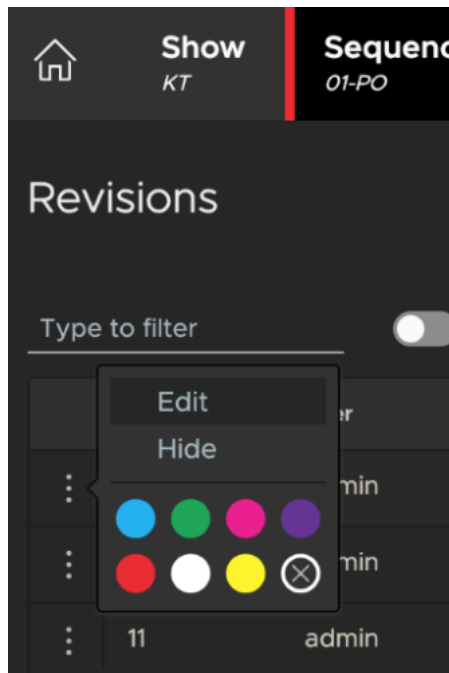
Date
Apr 29, 2019, 6:23:55 PM
Apr 29, 2019, 5:41:21 PM
Apr 29, 2019, 5:40:05 PM
Apr 29, 2019, 5:38:29 PM
Apr 29, 2019, 5:37:47 PM
Apr 29, 2019, 5:17:07 PM
Apr 29, 2019, 5:16:07 PM
Apr 29, 2019, 5:12:24 PM

Column filters are useful for targeted searches.

Color Tagging a Sequence Revision

You can assign colors to Sequence Revisions to indicate ownership, state, purpose or use. For example the color yellow may be used to indicate sequence revisions that are currently in editorial.

- Navigate to the **Revisions** level and click the kebab menu  on a sequence revision, then select a color.



- Select the  button to remove a color tag.
- To filter a list of revisions by color, type the color name into the filter menu. For example, type "green" into the filter to display only revisions tagged with the color green.



Note: Color selection is also available when hovering over the **Revision** tab in the breadcrumb at the top of the user interface.

Hiding Sequence Revisions

- Navigate to the **Episodes** level and click the kebab menu , then select **Hide**.
- Switch the **Show Hidden** toggle on in the UI to display hidden shows. From this view you can click the kebab menu , then select **Unhide** to restore the show's visibility.

Managing Autosaves

Autosaves help when you need to revert back to a previous Sequence Revision, but they can lead to a list of hundreds of iterations. You may want to hide them to keep your list of revisions set to only display manual saves.

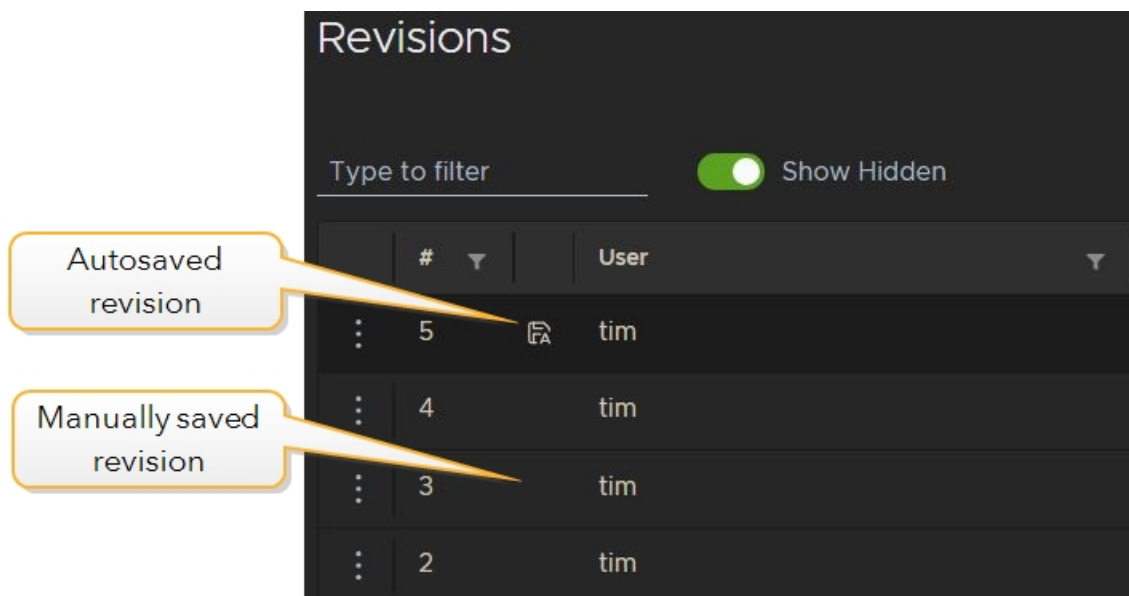
To automatically hide autosaves:

1. Navigate to **Preferences > General Settings**

Here you can enable or disable autosaves, set autosave frequency, and choose whether to hide autosaves automatically from view.

2. Toggle **Automatically Hide Autosaves** on.

Autosaves won't appear in your list of revisions until you toggle **Show Hidden** on the **Revisions** page.



Autosaves are easily identified in the list of revisions by their autosave icon.

Settings & Preferences

Before starting work on a production, it is a good idea for an administrator to establish the settings for all Flix users first.



Note: Only administrators and users that are specifically granted permissions can make changes to the **User Management**, **Group Management**, and **Studio Settings** tabs in the Management Console. See [Control User Access with Group and Role Permissions](#) for more information.

Studio Settings

To set preferences at the studio level:

1. Navigate to **File > Management Console > Studio Settings**.
2. Edit the required preferences and enable their corresponding checkboxes under **Enforce at Studio Level**.

This overrides the preferences set in the **File > Preferences** dialog. They appear as read only in the [Flix Preferences](#) dialog.

Show Level Settings

To set preferences at the show level:

1. Navigate to the **Shows** level, click the kebab menu  on the required show and click **Settings**.

This opens the **Show Settings** dialog for that specific show.

2. Edit the required preferences and enable their corresponding checkboxes under **Enforce at Show Level**.
3. On the **Branding** tab, you can select images and add a disclaimer to apply to any contact sheets you create during publish or export from the current show. See [Customizing Contact Sheet Templates](#) for more information.

These settings override preferences set in the [Flix Preferences](#) dialog and at the studio level for that specific show, with the exception of the **Branding** settings, which only apply at show level.

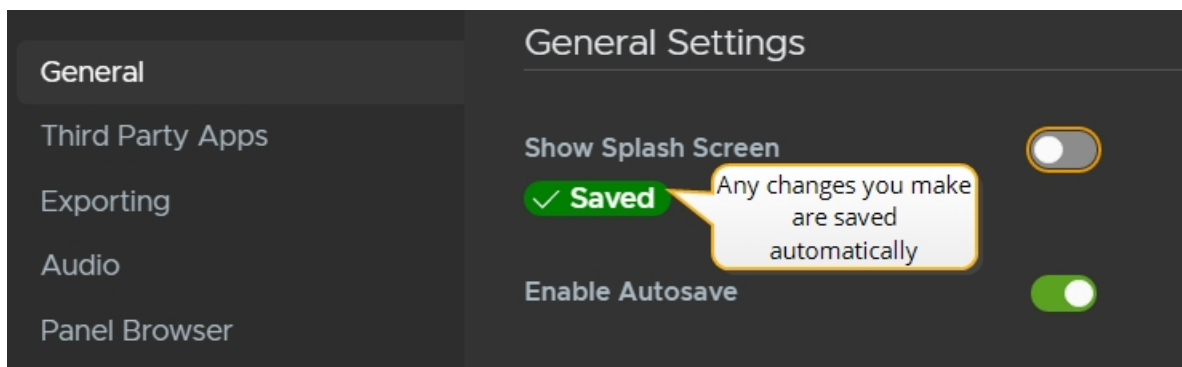


Note: Studio and Show Settings can only be changed by administrators and users that are specifically granted permissions. See [Control User Access with Group and Role Permissions](#) for more information.

User Level Settings

To set preferences at the user level:

1. Navigate to **File > Preferences**.
2. Edit the required preferences, for example set your audio output device.



Setting Naming Conventions

Flix allows you to set naming conventions for file exports and publish directories using 'chips', which are placeholder shortcuts to metadata. They appear in the following format:

[date] [show_tracking_code] [show_title] [episode_tracking_code] [episode_title] [sequence_tracking_code] [sequence_title] [sequence_revision]



Note: Naming conventions can be enforced at the studio, show and user levels.



Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **\${HOME}** on macOS.

Example 1

To set a default naming convention for exported sequences:

1. Go to **File > Preferences > Exporting**.
2. Click in the **Filename Format** field, then click on the chips at the bottom of the Preferences window to set your naming convention. The default is **[show_tracking_code]-[sequence_tracking_code]-v[sequence_revision]**.

For example, if your show tracking code is 'THP', sequence tracking code is 'WED', sequence revision version is '25' and your default export path was **/mnt/flix_publishes/** the full directory and exported file would be named **/mnt/flix_publishes/THP-WED-v25**.

All future exports follow this naming convention.

Example 2

To set a default naming convention for where Flix stores published files for Editorial:

1. Go to **File > Preferences > Third Party Apps > General**
2. Click in the **Publish Directory** field, then click on the chips at the bottom of the Preferences window to set your naming convention.

For example: Let's say your Publish Directory is set to **/mnt/flix_publishes/[show_tracking_code]/[sequence_tracking_code]/[date]**. If your show tracking code is 'THP', sequence tracking code is 'pilot' and the date is July 15 2020, the full directory would be **/mnt/flix_publishes/THP/pilot/20-07-15/**.



Note: The **Publish Directory** setting for both Windows and Mac is available at the studio or show level, for cases where multiple users might be on different operating systems.



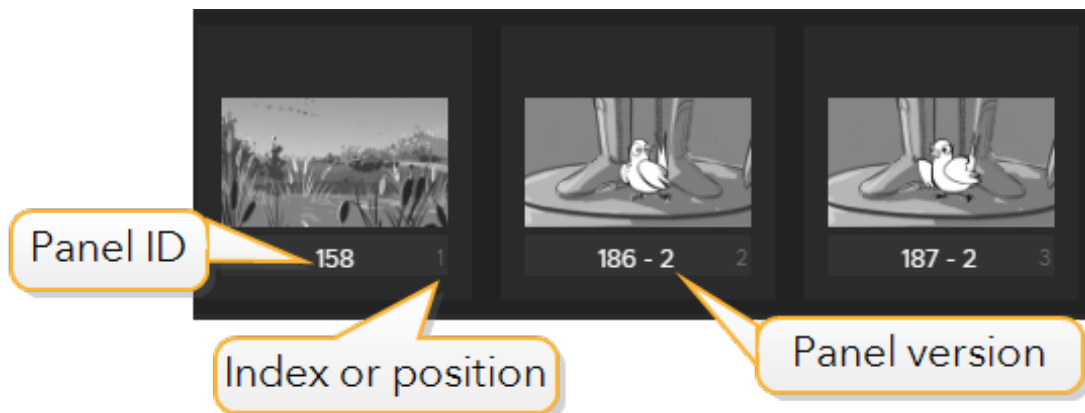
Note: Flix will automatically create missing directories if they don't already exist.

Using the Panel Browser

The Flix Panel Browser is the workspace in which you'll spend most of your time. The video below gives a quick overview of how the Panel Browser works.

In the video:

- Importing panels. For more information, please refer to [Importing Panels into Flix](#).
- Each panel displays a unique Panel ID and Index, or position number. If the panel has been updated, a new version is created and numbered.



Note: You can swap the position of each panel's unique ID with its Index number. See [Swap Panel ID with Index](#) in Preferences.

- Move panels around by selecting one or more, and dragging them where you want to place them. Hold **Shift** and click to select a sequence of panels, or **Ctrl/Cmd** and click to select individual panels.
- Remove panels from your current edit by clicking the Trashcan icon in the Edit toolbar.
- Hide panels by clicking on the panel and toggling **Visibility** in the **Properties** tab.

- Re-use panels by using the Copy and Paste buttons in the Edit toolbar. This creates new panels re-using the same panel, timing, and dialogue.
- Adding and versioning dialogue.

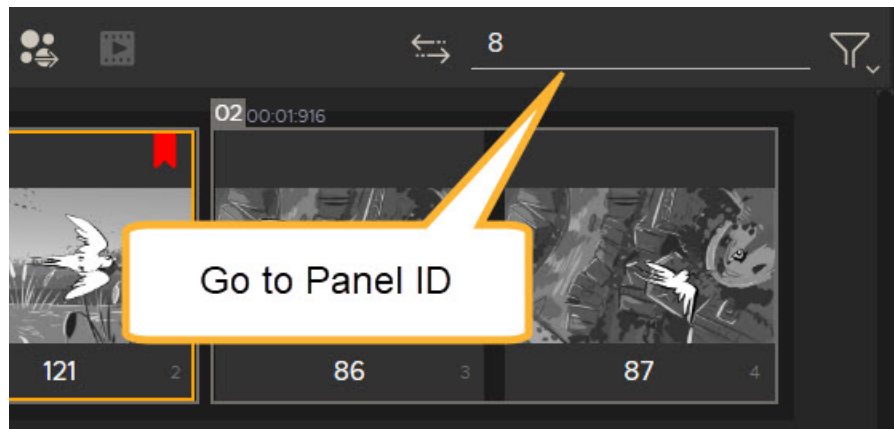


Tip: Duplicate selected panels with the  **Duplicate** button.

Quickly Locate Panels Using the Panel ID

A sequence can contain hundreds of panels, but you can search by panel ID in the dedicated search bar to quickly locate the panel you need. Enter a panel ID and press **Enter** to select any match in the sequence.

If no match is found, a warning icon  is displayed in the **Go to panel ID** field. Panel search does not hide other panels in the revision in the same way as the filter.

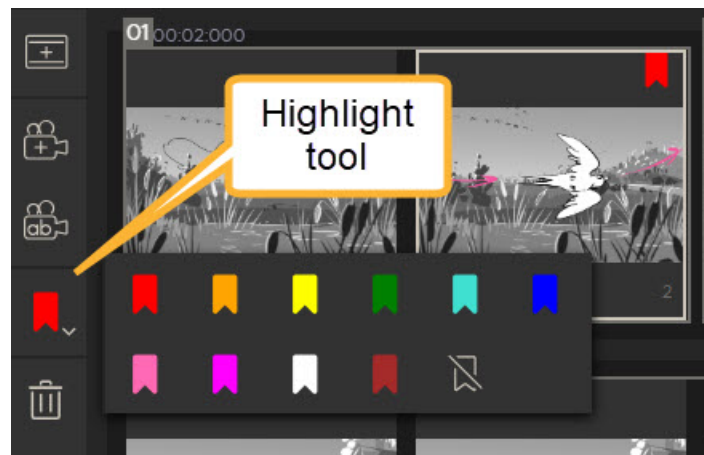


Highlighting Panels

To highlight panels in your sequence:

1. In your edit (sequence revision), select one or more panels.
2. In the left toolbar, click and hold the Highlight button and click on a color.

This selected panels are tagged with the chosen color.



To remove highlights:

1. Select the highlighted panels.
2. In the left toolbar, click the Highlight button.

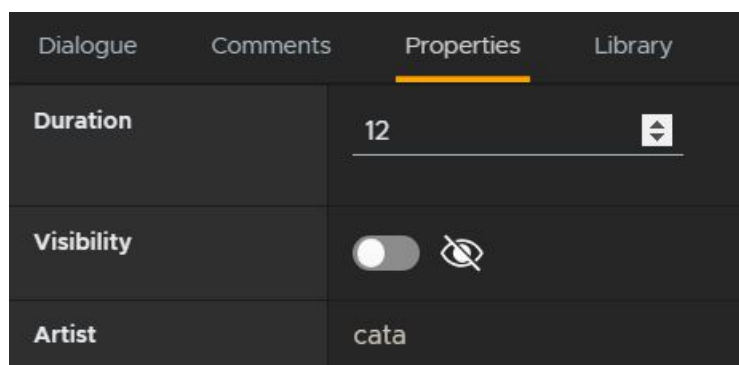
This removes the highlight from the selection.

Hiding Panels

Artists may have panels in a sequence that they need, but don't intend to send to editorial, such as reference artwork. Hidden panels are skipped during export, publish, pitch recording, and playback operations.

To toggle a panel's visibility:

1. Click on the panel.
2. Open the panel **Properties** tab.
3. Toggle the panel **Visibility** off.





Tip: It may be necessary to hide panels from visibility for troubleshooting purposes, like in cases where an artist needs to send a sequence to editorial that contains panels with errors. Hiding those erroneous panels means they are skipped in the export to editorial.

Using Shots in Flix

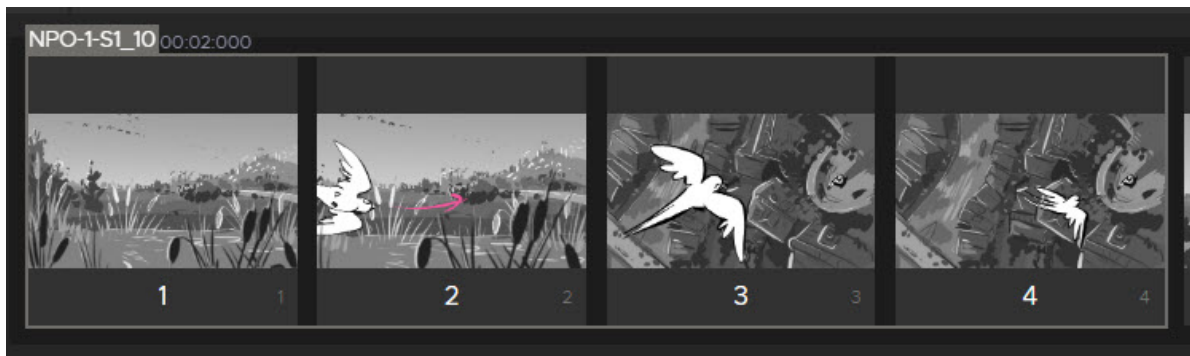
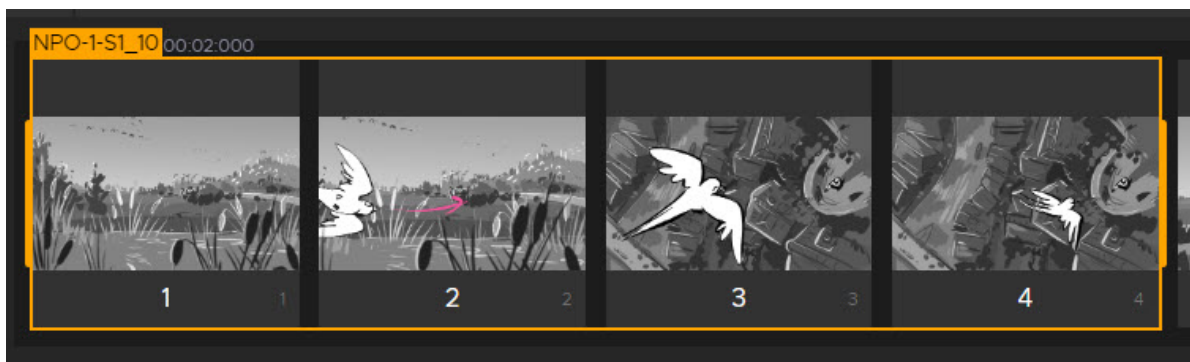
Shots are used to show certain sections of a sequence, such a character motion or camera move. In Flix you can create, modify and manage Shots within your Sequence Revision workflow. Shots work as a container for selected panels and can be exported to Storyboard Pro.

To create a new shot:

Select the panels you want to group into a shot using Shift+LMB. Then click the Shot icon in the Panel

Browser's left toolbar ()

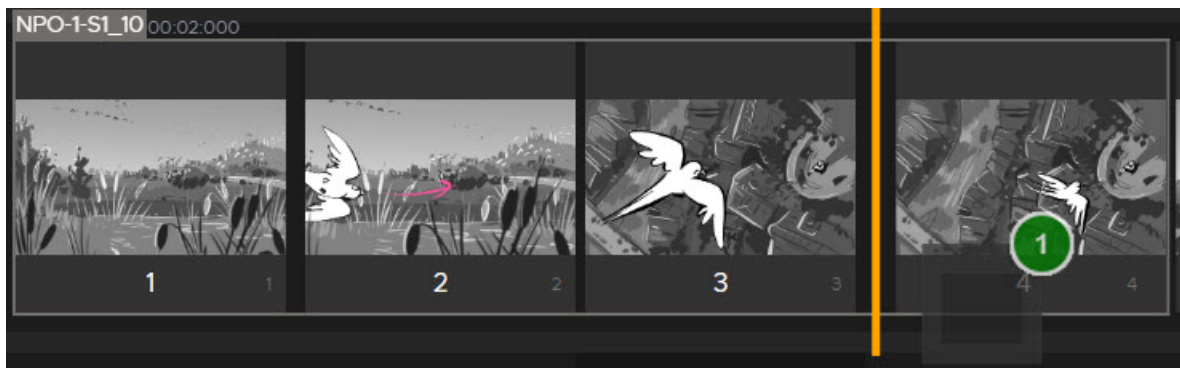
This creates a new Shot. Shots are represented as borders around the grouped panels. The border is yellow for selected shots and light gray for inactive ones.



Editing Shots

There are different tools to modify the panels that make up a shot.

- Reorder - You can reorder shots by selecting a shot and dragging it to the position you want it.
- Copy and paste - You can copy and paste shots. After copying, click the panel you want the shot to appear after, and then paste the Shot in.
- Add new panel to Shot - To add a new panel to the shot, click on the panel you want to add and drag it to the position you want it within the shot. A yellow line will appear to indicate where the new panel will be placed. Drop the panel in your desired location and it will now be part of the shot.



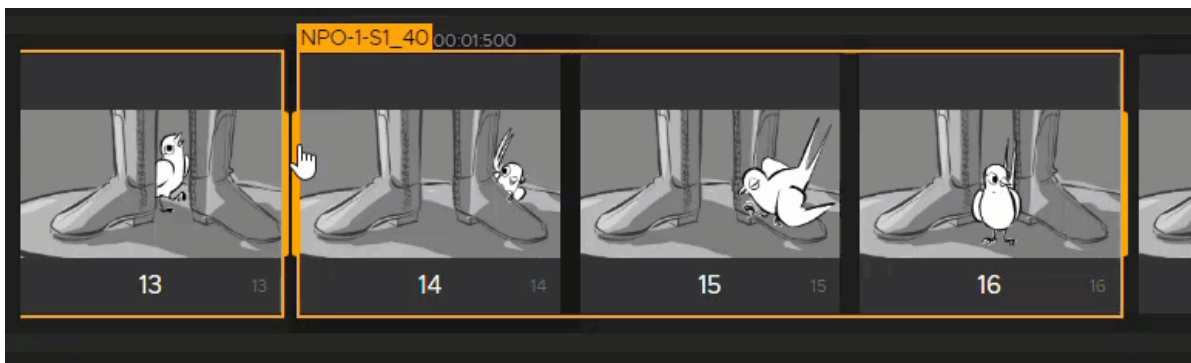
- Add panel to start/end of shot - To add a panel to the start or the end of a shot, drag the panel to the border of the shot and when the arrowed line appears, drop the panel in that position.



- Change Shot length - You can also drag the handles on either side of the shot border to include adjacent panels. If the handle reaches the start of a different shot, it cannot go any further.



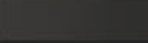
- Edit multiple shots - with two or more shots selected, a handle will appear between the start and end of each shot. Drag any of these intermediate handles to change the endpoint of the shots either side. Dragging the start handle of the first shot will change the start frame of that shot - this behavior also applies to the end handle of the last selected shot.



- Merge Shots - You can merge shots by selecting a shot and moving it inside of a different shot. This will place the added panels to the end of the existing shot.

Shots Properties Panel

The Shots properties panel provides a centralized view of shot details. It includes a breakdown of the panels that make up each shot.

Dialogue	Comments	Shot Properties	Library
Name	NPO-1-S1_30		
Duration	00:03:500		
Panels	 7  0  0		
Panel Artists			
Created	Jul 30, 2025, 2:20:38 PM - 		
Origin	Flix		

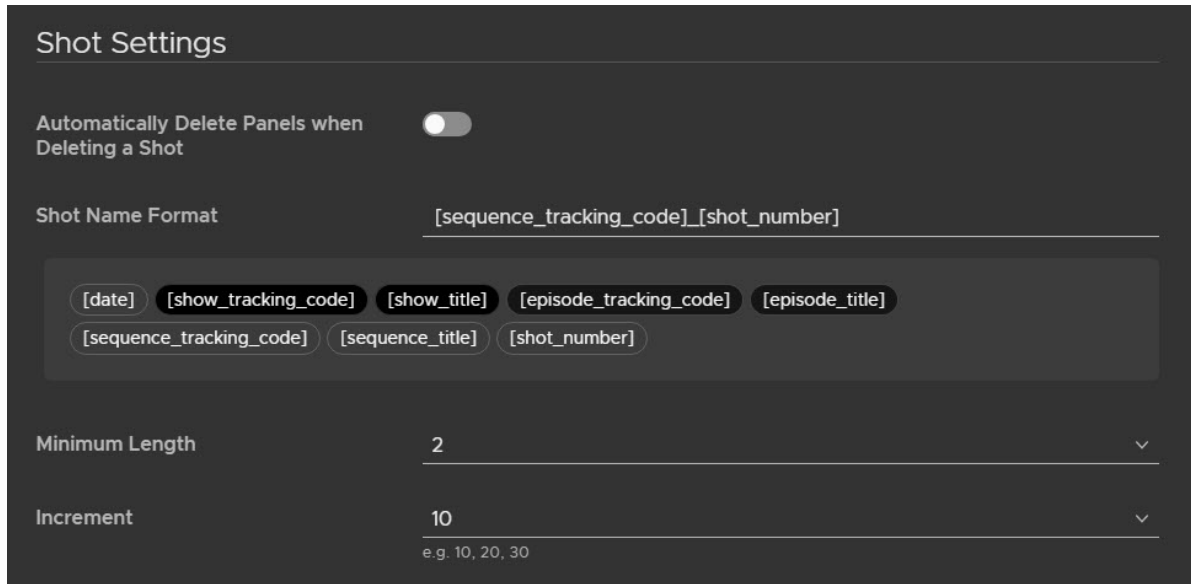
Custom Workflows

For custom workflows and automation, all shot functionality is fully accessible through the [API](#).

Naming Shots

You can configure the naming convention for shots created in your Flix sequence to show information such as shot number and sequence title on a shot. Naming Shots in Flix is fully customizable. You can rename shots at any stage of the production.

Shots are automatically named using a configurable **Shot Name** preference. You can define default naming conventions and incremental numbering at **Settings > Panel Browser**. You can also toggle whether to automatically delete panels when deleting a shot.



The screenshot shows the 'Shot Settings' panel. At the top, there's a toggle for 'Automatically Delete Panels when Deleting a Shot' which is currently off. Below that is the 'Shot Name Format' field, which contains the text '[sequence_tracking_code]_[shot_number]'. Underneath the format field is a collection of buttons representing different metadata fields: '[date]', '[show_tracking_code]', '[show_title]', '[episode_tracking_code]', '[episode_title]', '[sequence_tracking_code]', '[sequence_title]', and '[shot_number]'. At the bottom, there are two dropdown menus: 'Minimum Length' set to '2' and 'Increment' set to '10'. A small text 'e.g. 10, 20, 30' is visible below the 'Increment' dropdown.

Shot data is saved per revision and is not contained in version history.

Setting Shot Number Format

- Navigate to **Preferences > Panel Browser > Shot Number Format**.
- Choose from the **Minimum Length** dropdown menu to set the minimum character length of shot numbers displayed.
- Choose from the **Increment** dropdown menu to set the increment for shot numbers displayed.

Example:

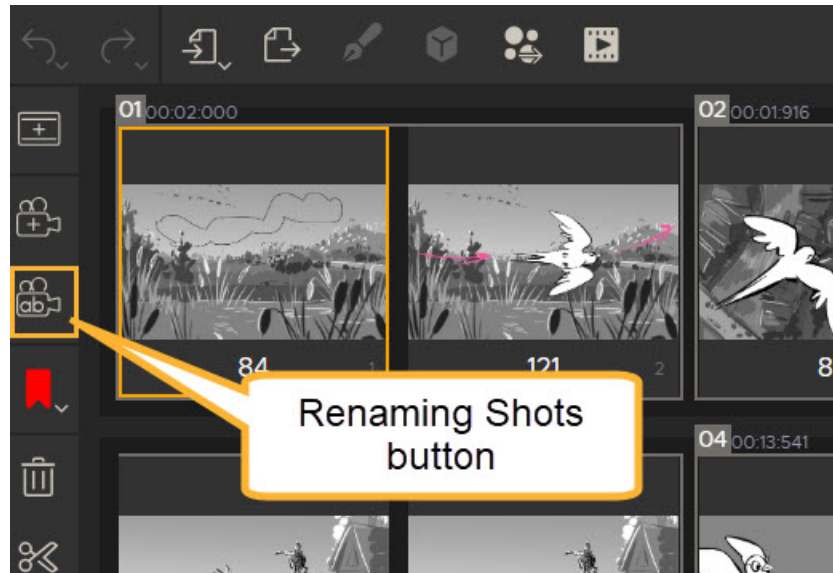
Shot numbers displayed as 01, 02... have a **Minimum Length** set to 2 and **Increment** set to 1.

Shot numbers displayed as 0010, 0020... have a **Minimum Length** set to 4 and **Increment** set to 10.

Renaming Shots

When you move shots around in your sequence, you can rename them sequentially. To do this:

1. If you only want to rename certain shots, select the ones you want to rename. If you want to rename all shots, you do not need to select them all.
2. Click the **Rename Shot(s)** button. Alternatively, go to **Edit > Rename Shot(s)**, or press **F2**.



3. Adjust the **Shot Name Format**, shot length and incrementation. A preview of the new shot names will appear in the **Preview** window.



Note: The Shot Name Format must include the **[Shot_number]** option.

4. Click **Rename Selected Shots** or **Rename All in Sequence**.

Editorial - Sequential Shot Renaming

×

Shot Name Format

[episode_tracking_code][episode_title][shot_n...

[date]

[show_tracking_code]

[show_title]

[episode_tracking_code]

[episode_title]

[sequence_tracking_code]

[sequence_title]

[shot_number]

Start

10

Minimum Length

1

Increment

10

Preview

Shot Name00:03:00

Shot Name00:03:00

Shot Name00:03:00

11

22

11

RENAME SELECTED SHOTS

RENAME ALL IN SEQUENCE

CANCEL

You can also rename shots by double-clicking on the Shot border name, or by pressing **Enter** when a shot is selected.

Exporting Shots During Publishing

When publishing your sequence to Editorial software, the shot data can be exported as timeline Markers. If you are using Avid Media Composer, it can also be exported as a filler on a second track.

Using Markers, you can choose between having the shot name appearing in the marker name or the comment field. The input you don't use for the shot name will have the name of the shot creator.

To edit this, go to **User settings > Third Party Apps > Editorial Tools > Send / Receive Markers**. This is ticked by default.

From here you can control whether the shot name or creator will appear as the Marker name or comment.

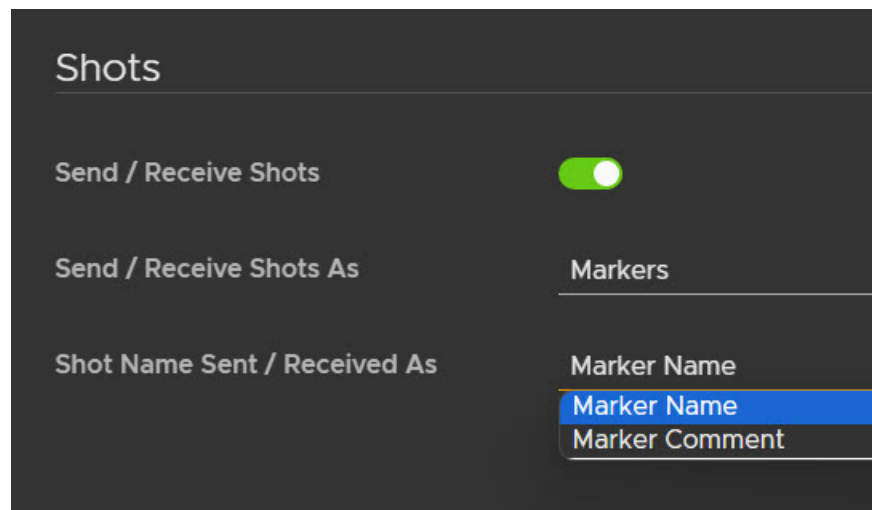
The default setting is **Shot Name Sent / Received As = Marker Name**. This will mean the **Shot name** → **Marker name** and **Shot creator** → **Marker comment**.

Alternatively, if you set the **Shot Name Sent / Received As = Marker** comment, the **Shot Creator** will be **Marker Name** and **Shot Name** → **Marker comment**.

If a shot is non visible (e.g. for handling gaps), it will be marked with a **Null_Shot** Marker name when exported to editorial.

When shots are returned to Flix they will have their naming and structure preserved.

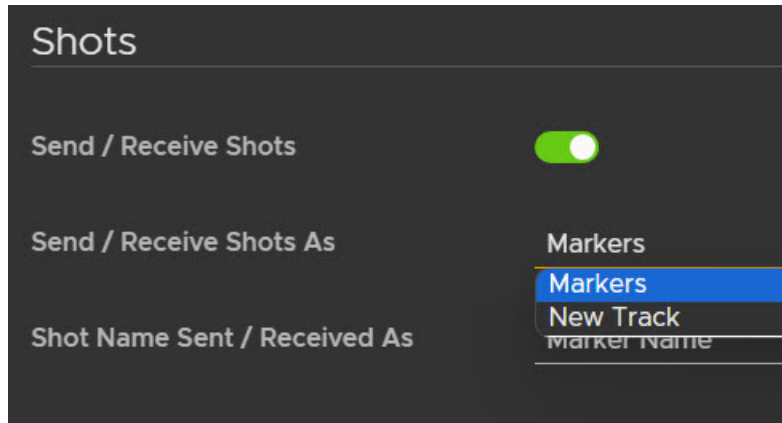
Shot names are embedded in clip metadata, enabling them to be displayed as overlays in the edit and removing the need for manual labeling.



Avid Media Composer

There are two ways to import shots into Avid Media Composer.

- Export them as Markers, set to the current track.
- As Filler entities on a dedicated New track.



Adobe Premiere

You can export shots to Adobe Premiere as Timeline Markers.



Note: Second track is not an option for Adobe Premiere.

Shots and Storyboard Pro

You can export shots to Storyboard Pro, and they will be represented as Scenes.

When importing a Storyboard Pro project into Flix, Storyboard Pro scenes will automatically be converted to shots in Flix.



Note: Panels that aren't part of a Shot in Flix will appear in Storyboard Pro as **Null_Shot_(num)**. If there are multiple of these panels and they are non-sequential, each must have its own uniquely numbered **Null_Shot_(num)**, as Storyboard Pro does not allow multiple scenes with the same name in a sequence.

Giving and Managing Feedback

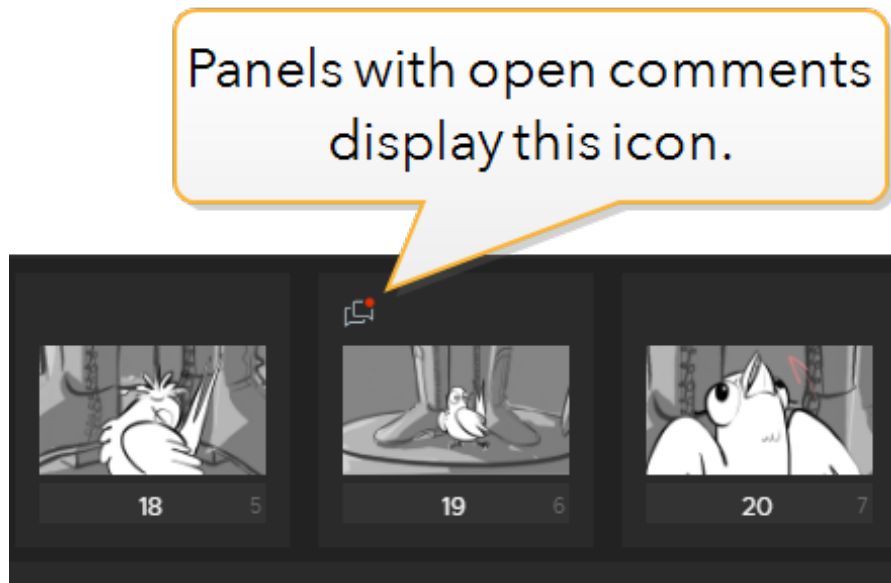
Adding Comments to Panels

You can add comments to individual panels, which allows users to create a feed of notes and feedback on a sequence. Any comments written on a panel are flagged, so you can see at a glance which boards require attention.

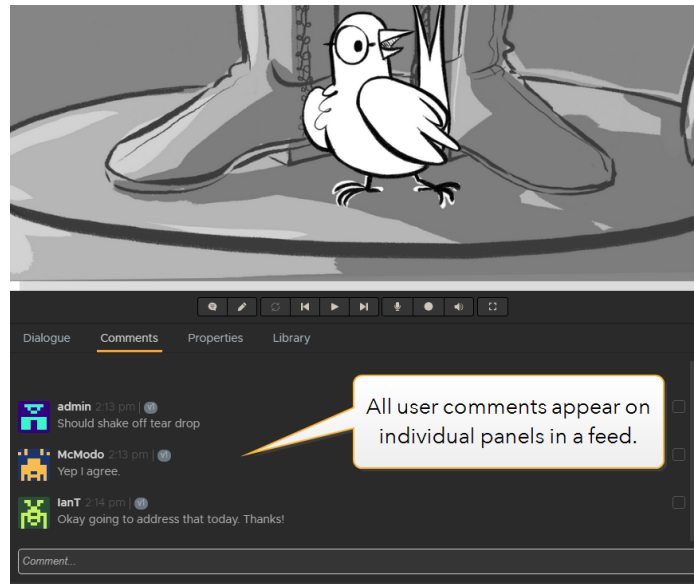
To add a comment:

1. Click on **Comments** under the Player.
2. Type your comment into the **Comment** window and press Enter to publish it.

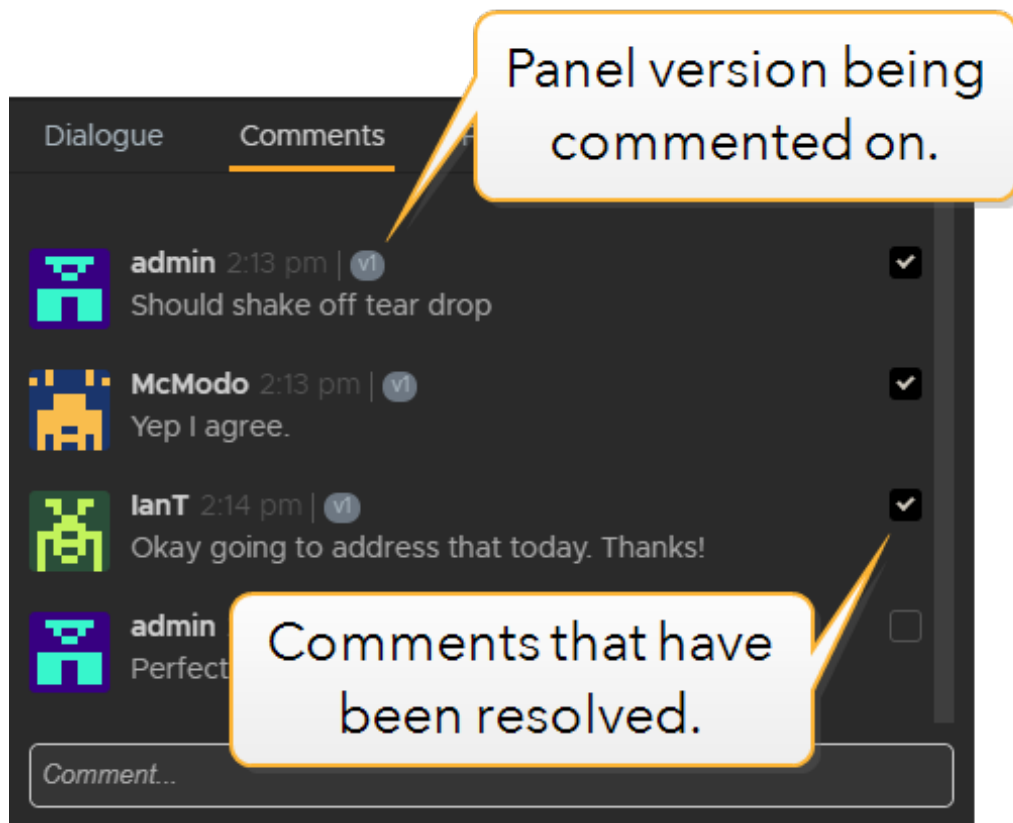
The panel on which you've commented now displays a comment icon. Hover over the icon to see the latest comment.



Your comment appears in a feed with any other comments other users have added to that panel.



Every comment has a checkbox which allows users to mark as resolved. For example, once feedback in a comment has been addressed, the user would tick the comment so everyone in the production knows that feedback has been actioned.





Note: Once all the comments in a feed have been resolved, the comment icon disappears from the panel.

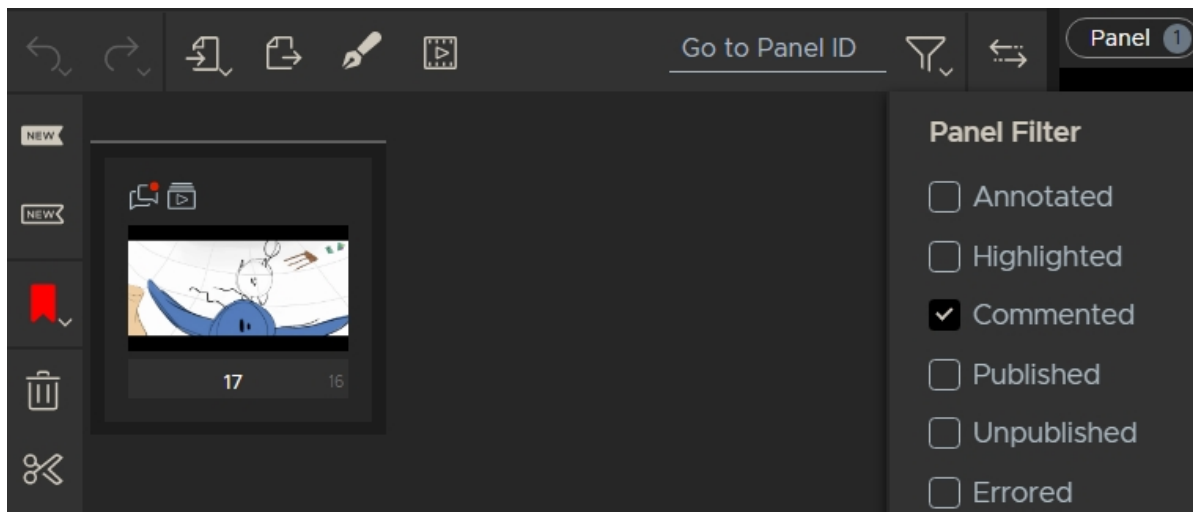
Filtering a Sequence by Comment

You may want to quickly filter your sequence to display only panels with comments made on them.

To filter by comment:

1. Click on the filter icon in the panel browser.
2. Select **Commented Panels**

Flix displays only panels with active comments on them.



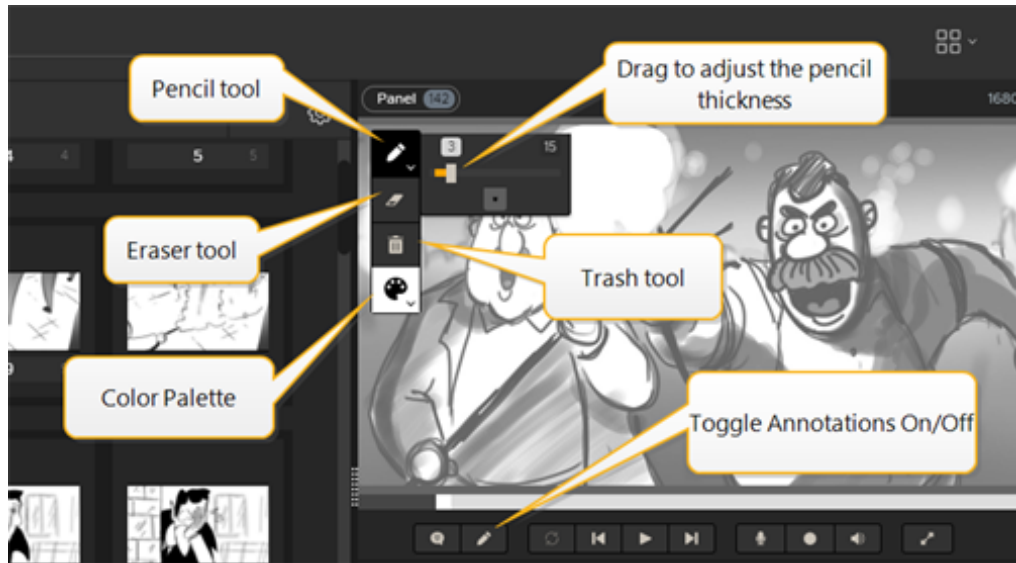
Annotating Panels

You can annotate your panels directly in the Player, which is located at the top-right corner of the application in the **Story** workspace.



Note: To add an annotation to a panel using Photoshop, refer to [Giving and Managing Feedback](#).

1. In the Panel Properties pane, toggle the annotations on by clicking the Pencil button at the bottom of the Player.

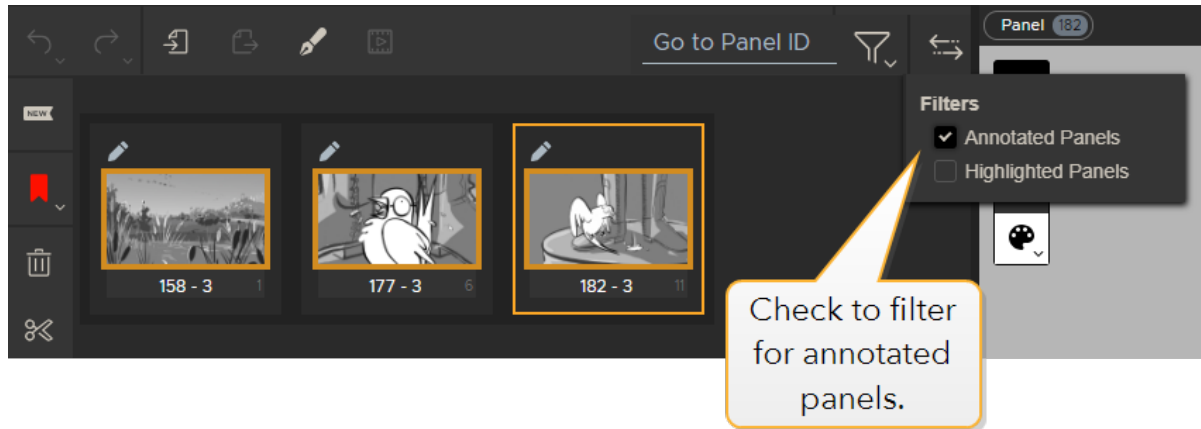


2. Click the Color Palette button then click again to select a color.
3. Click and hold the Pencil button then drag the slider to adjust the thickness of the pencil.
4. Make annotations on your panel.
5. Use the Eraser tool to partially erase your annotations or use the Trash tool to completely remove the annotations.

Annotated panels are tagged with the pencil icon and display a border around the thumbnail.



6. To display only panels containing annotations, click the filter  button and enable the **Annotations** filter.

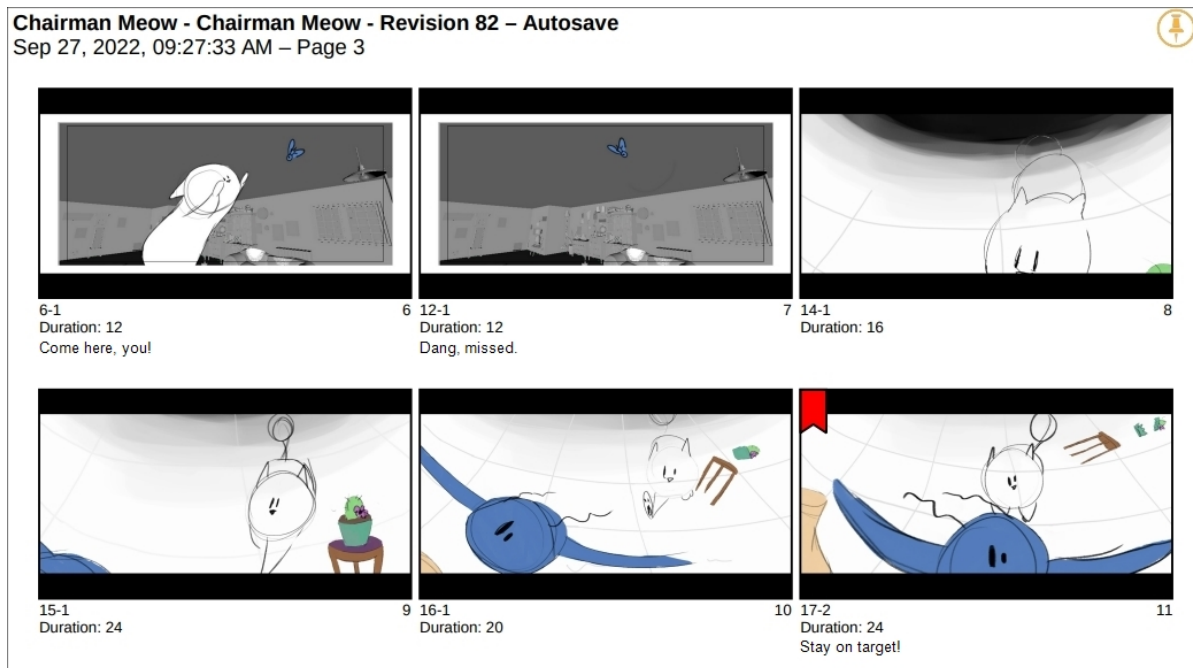


Tip: By default, annotations appear as an extra layer in PSD files open in Photoshop. You can disable this by going to **File > Preferences > Third Party Apps > Adobe Photoshop** and disabling the **Send Annotation as Layer** preference.

Customizing Contact Sheet Templates

Contact sheets provide a quick overview of all the panels in a revision, allowing you or a peer reviewer to get a feel for the story without having to open Flix, Photoshop, or another application.

Contact sheets include useful information such as the show, sequence revision, panel number, and duration. Any dialogue associated with the panels can also be included as well highlights, markers and other useful data.



A 3x2 Storyboard (Landscape) Contact Sheet

Flix ships with several default contact sheet templates, but you may find that creating your own templates covers a wider range of shows.



Note: Only administrators and users that are specifically granted permissions can edit or create templates. Regular Flix users can only select which template to use from the available publish and export options. See [Control User Access with Group and Role Permissions](#) for more information.

The default templates are fully editable and you can duplicate and delete them quickly and easily. The default templates are:

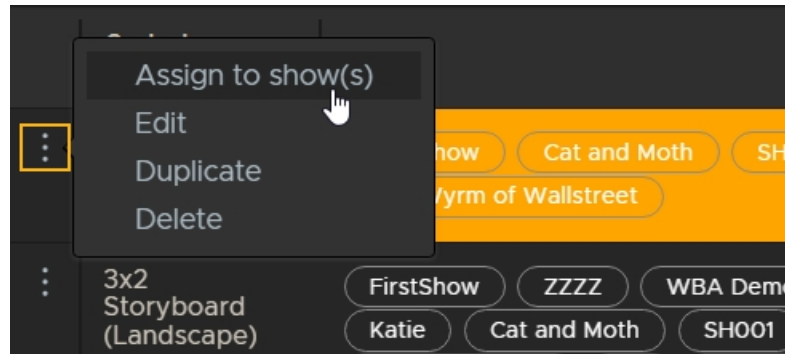
- Single Screen (Landscape) - each panel has its own page in the PDF output with dialogue under each panel.
- 3x2 Storyboard (Landscape) - six panels per page in the PDF output with dialogue under each panel.
- 3x3 Storyboard (Landscape) - nine panels per page in the PDF output with dialogue under each panel.
- 5x4 Storyboard (Landscape) - 20 panels per page in the PDF output with dialogue under each panel.
- 3 Dialogue Review (Portrait) - three panels per page and any associated dialogue on the right.
- 6 Dialogue Review (Portrait) - six panels per page and any associated dialogue on the right.

Assigning Templates to Shows

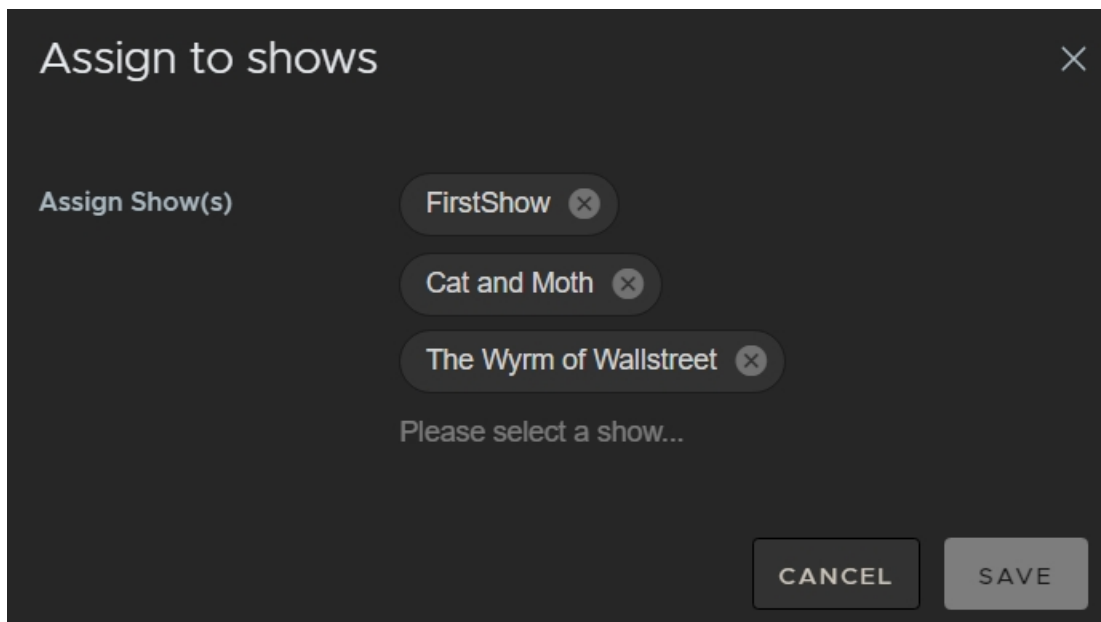
Flix system and custom templates are not assigned to any shows by default. You must assign each show manually so that publishes and exports from that show can select a contact sheet template,

To assign a show to a template:

1. Click the menu  icon next to the template you want to edit and select **Assign to show(s)**.



The **Assign to shows** dialog is displayed.



2. Click **Please select a show...** and then choose the show you want to assign.
3. Add as many shows as required and then click **Save**.



Tip: You can remove a show by clicking the  icon on the right-hand side of its name.

The assigned template is now selectable in the **Show Settings**. See [Editing a Show](#) for more information.

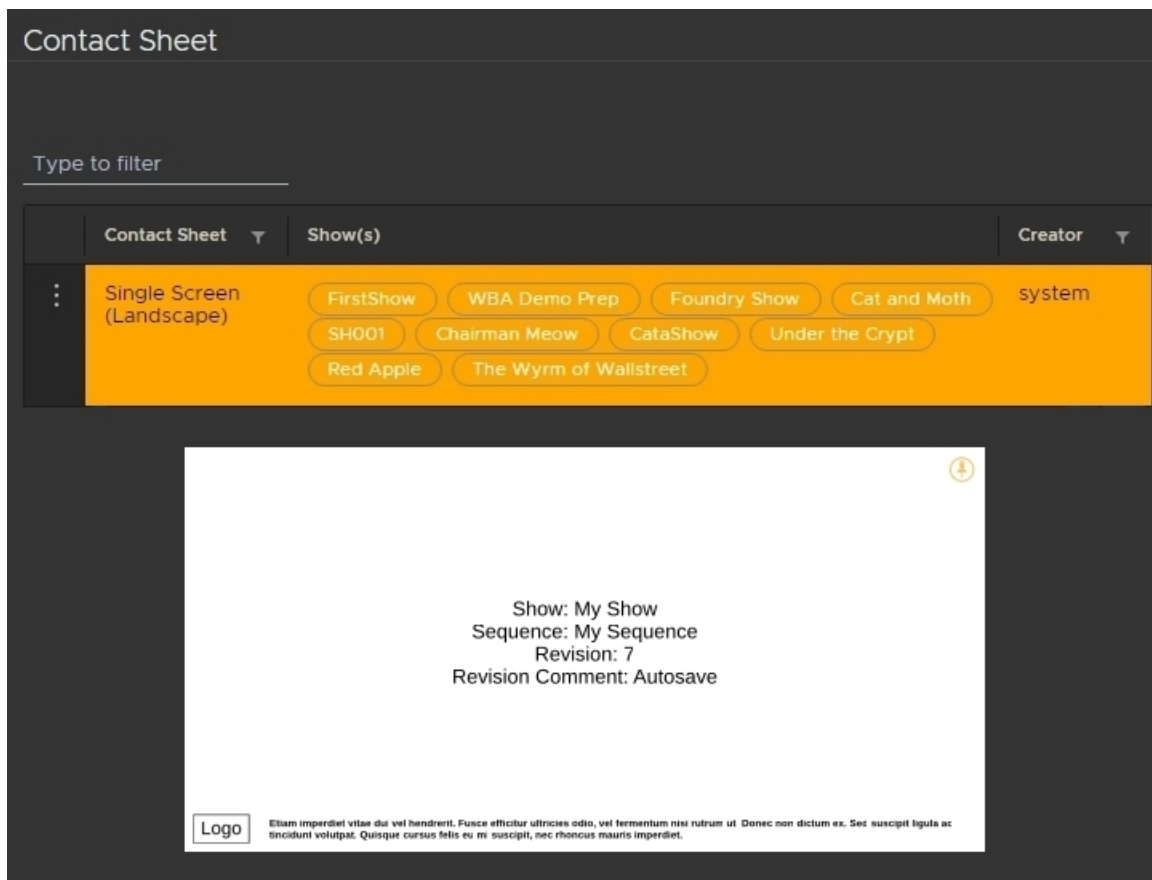
Editing an Existing Template

As an administrator, you may find that the existing templates don't quite suit your needs. Editing the default templates is the fastest way to meet those needs, but be aware that any shows using these templates are forced to use the updated version. If you'd rather create a new template, see [Creating a New Contact Sheet Template](#) for more information.

To edit a template:

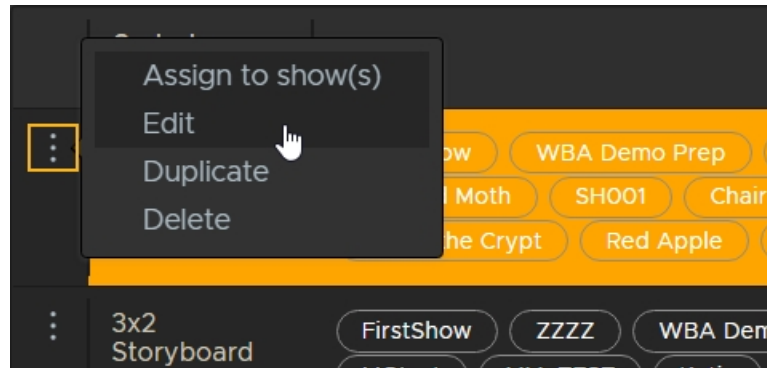
1. Login as an administrator.
2. Go to **File > Management Console** or use the **Ctrl/Cmd+. (period)** keyboard shortcut.
3. Click the **Templates** tab to display all the currently available templates.

Each template displays its name, the shows to which it is assigned, the creator, and a preview.



A Single Screen (Landscape) Contact Sheet and Preview

4. Click the menu  icon next to the template you want to edit and select **Edit**.



5. Make the required changes on the **General**, **Branding**, and **Cover** tabs.



Note: Enabling the **Branding** > **Watermark** and **Company** options displays the branding set in the **Show Settings** > **Branding** tab. See [Show Level Settings](#) for more information.

6. Click **Save & Apply**.

You can then set shows to have access to certain templates. See [Customizing Contact Sheet Templates](#) for more information.

Creating a New Contact Sheet Template

Creating a new template takes more time, but gives you the freedom to set up the layout of the contact sheet without worrying about overwriting templates in use by other shows.

To edit a template:

1. Login as an administrator.
2. Go to **File** > **Management Console** or use the **Ctrl/Cmd+.** (period) keyboard shortcut.
3. Click the **Templates** tab to display all the currently available Contact Sheet templates.
4. Click **+ New Contact Sheet** on the right-hand side to get started.
5. New templates default to **Landscape, A4, 1 Panel Per Page** and the only required field is the **Template Name**.
6. On the **General** tab, change the **Style** to **Grid** or **List** and use the **Columns** and **Rows** fields to determine how many panels appear on each page of the PDF.
7. Checking **Panels** items adds the specified information to the PDF output. For example, checking **Panel ID**, **Panel Number**, **Highlight Tag**, and **Clip Duration** exports panels to contact sheets as shown.



8. On the **Branding** tab, enable **Watermark** and **Company** if you want to display them on the contact sheet.



Note: The **Watermark** image and **Company** name are set at the show level. See [Editing a Show](#) for more information.

9. And finally, on the **Cover** tab, check the **Information** boxes you want displayed on the contact sheet cover and enter a **Description**. If you don't require a cover, leave **Cover** disabled.
10. Click **Save & Apply**.

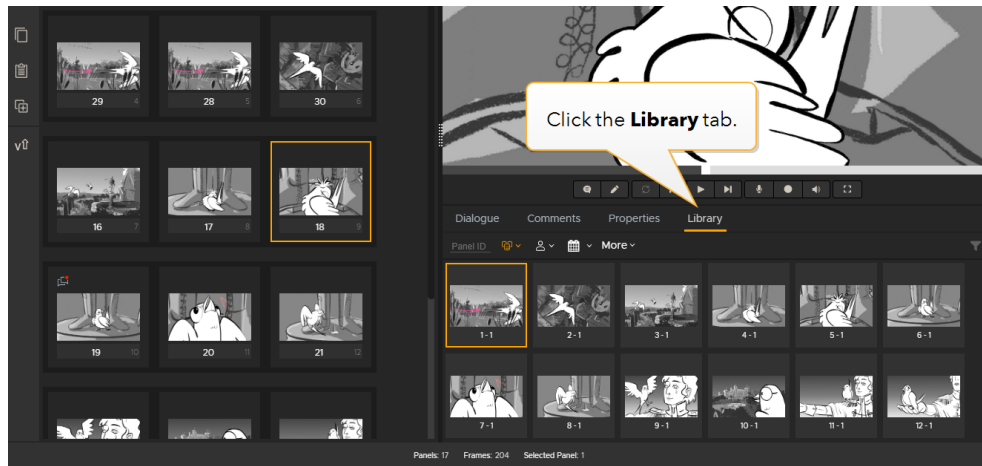
You can then set shows to have access to your new template. See [Customizing Contact Sheet Templates](#) for more information.

Locating Panels and Source Files

Locating Panels in the Library

You can use the Library to find old panels:

1. In the Panel Properties pane, select the **Library** tab.



2. Click the Filter by Sequence button and select a specific sequence or episode.

This displays the panels created in that particular sequence or episode.

3. You can also use different filters to narrow your search down:

Enter the Panel ID of the panel you are looking for.

Click the Filter by username button to display panels created by a specific artist.

Click the Filter by date range button to display panels created on a specific date or between two specific dates.

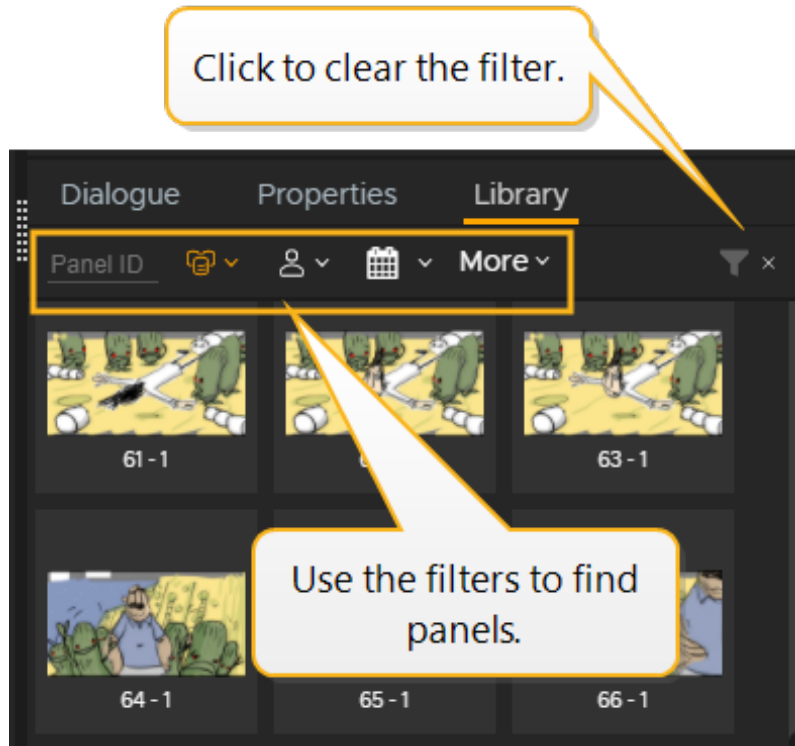
Click the **More** button to display the Master Images and/or all revisions as well.

4. When you found the required panels, select them and drag them in your current edit (sequence revision).

If a panel already exists in the current edit, Flix creates a new instance of that panel rather than using the same one.



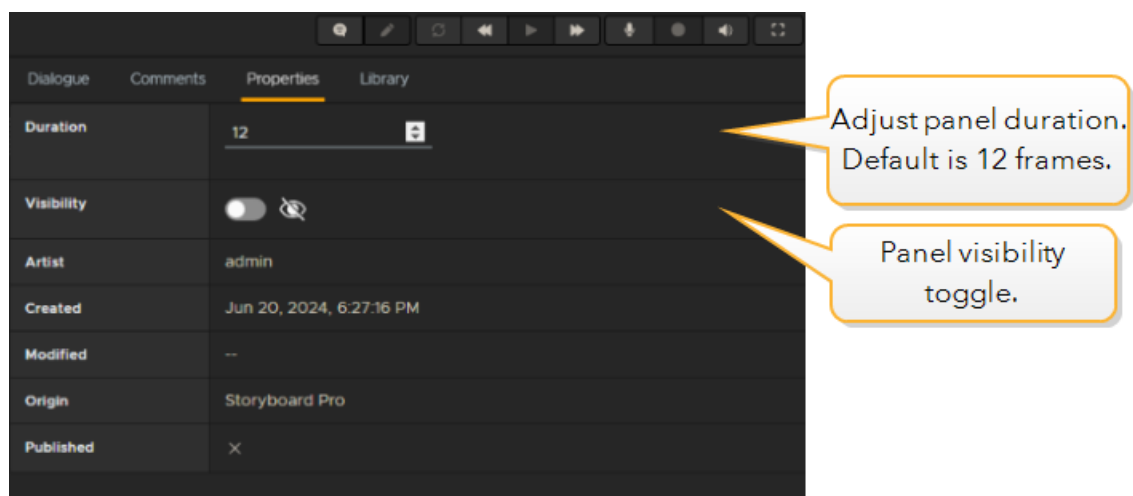
Note: Click the Reset Filters button to clear all applied filters.



Panel Properties

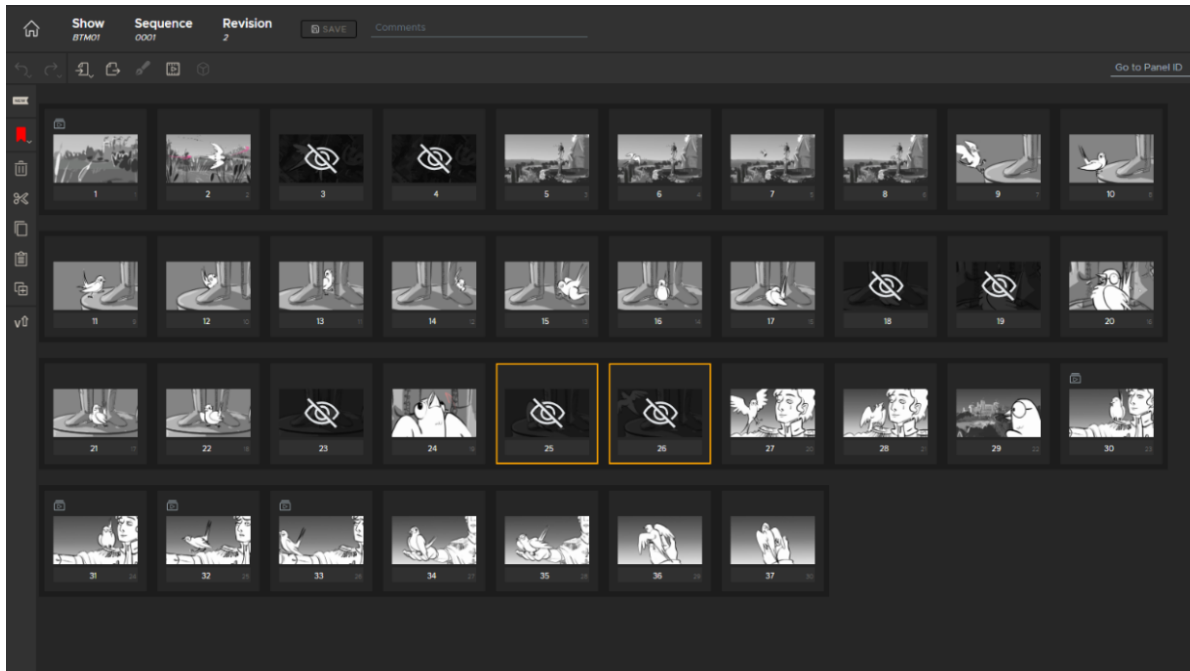
To check the specific properties of a panel in a sequence:

1. Select the panel.
2. Click on the **Properties** tab under the Player.





Note: Hidden panels are skipped during export, publish, pitch recording, and playback operations.



Browsing Source Files in the Source File Library

Flix lets you store and access source files associated with panels, such as Maya ASCII and binary files, Photoshop **.psd** files, or other integrated applications.

Adding files to the source file library on the server makes it available to any artist assigned to the show using the same Flix server.

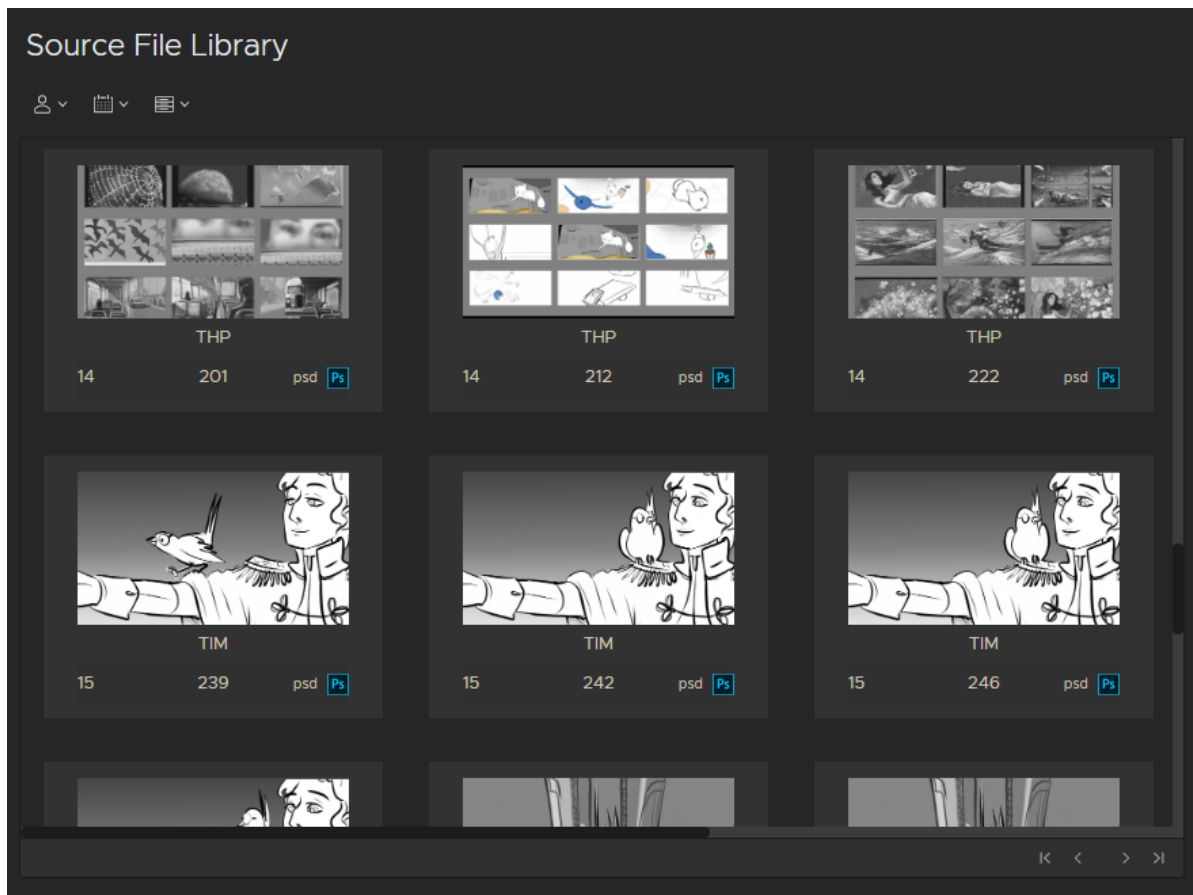


Note: Before Flix 7, a Photoshop source file was known as the "master image" and was accessed via the "MORE" menu.

To access the source file library:

- On Windows, go to **File > Source File Library**
- On macOS, go to **Flix > Source File Library**

The Source File Library displays files from all the shows that your account has been assigned to.



The Flix Source File Library

Filtering the Source Files

You can filter the source files by author, date created, show, and sequence using the drop-downs at the top-left.

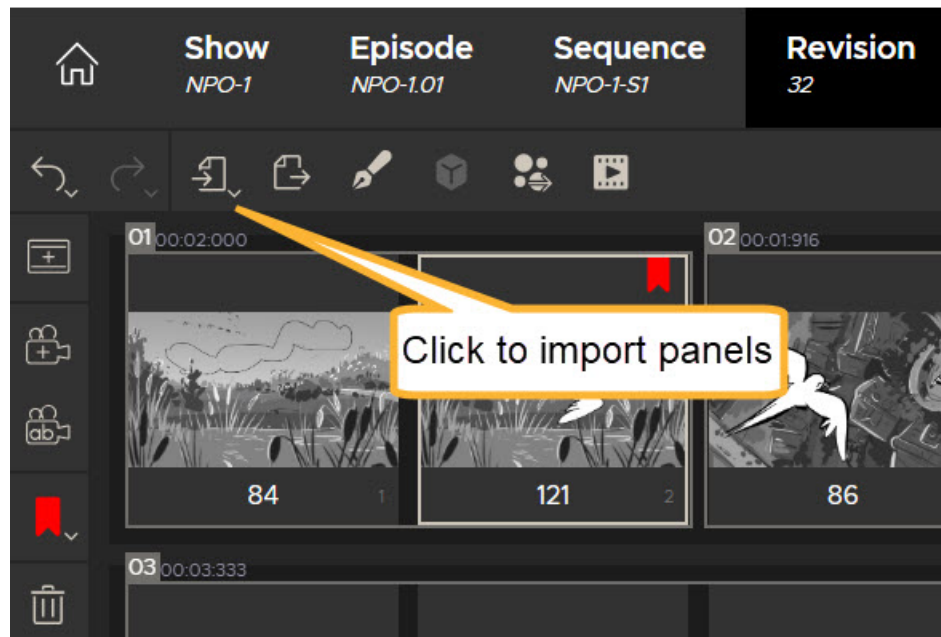
- Filter by author - Click on the person icon and select the name of the source file creator. Select again to remove the filter.
- Filter by date - Click the calendar icon and select a date range from the calendar. Click the **X** to remove the filter.
- Filter by show - Click the cabinet icon and select a show.
- Filter by sequence - The sequence icon appears when a show filter is active. Select a sequence from the list.

Flix for Story

Importing Panels into Flix

You can import panels and other file formats using the file browser or by dragging directly into Flix.

1. In the main toolbar, click the Import to Flix button.



This opens a file browser.

2. Select one or multiple files.



Note: You can import the following files: **.png, .psd, .jpeg, .jpg, .mov, .mp3, .tiff** and **.wav**.

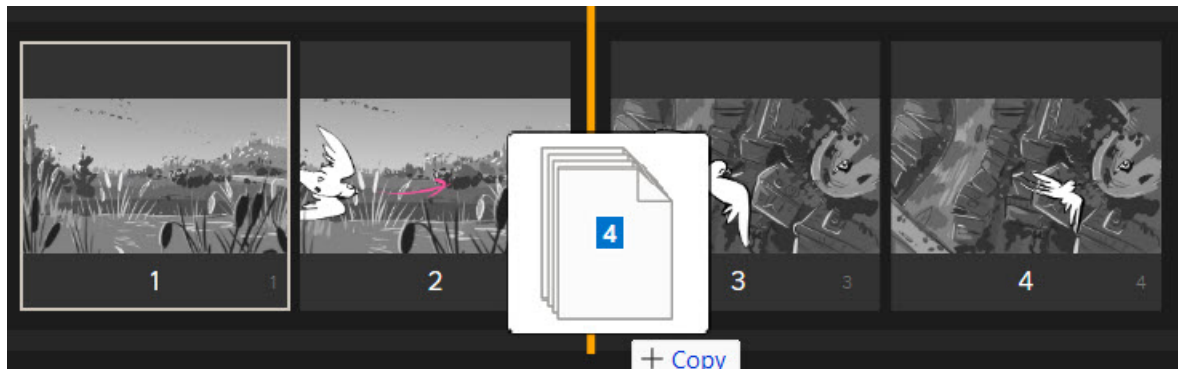
3. Click **Open**.

OR

From your file browser, drag one or more files directly into the Flix Panel Browser. You can also drag folders into the Panel Browser for Flix to import their contents.

Your files are imported in the sequence.

When you drag a file (or files) into the Flix panel, a yellow bar will appear to indicate where in the sequence the new panels will appear once dropped.

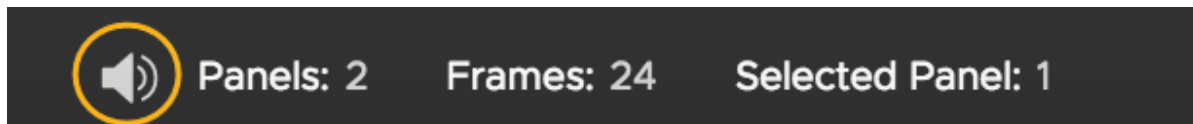


Warning: Make sure that you click the **SAVE** button when the import is complete or the new panels may not be added to the sequence.



Note: Flix keeps track of every imported image and reuses an existing panel instead of creating a new one. In cases where you need to reimport images that have been worked on previously, Flix recognizes which images it's seen before and only processes the new ones.

When audio, such as **.mp3** or **.wav** files are imported, Flix displays an audio icon in the Status bar.



Note: If you import a **.mov** file to the panel browser, any embedded audio contained in the file is not retained. Please see [Flix for Editorial](#) for details on importing movie files from editorial with audio.

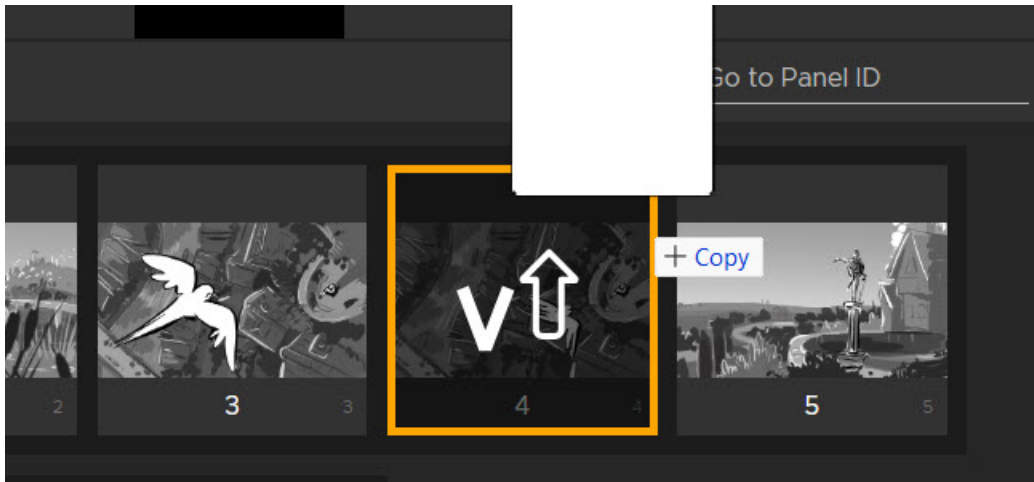
Panel Versioning

There are multiple ways to version up your panels in Flix.

You can drag new artwork directly onto an existing panel to automatically version it up with the new artwork. Full panel history is maintained so you can easily access earlier versions of the panel.

To do this:

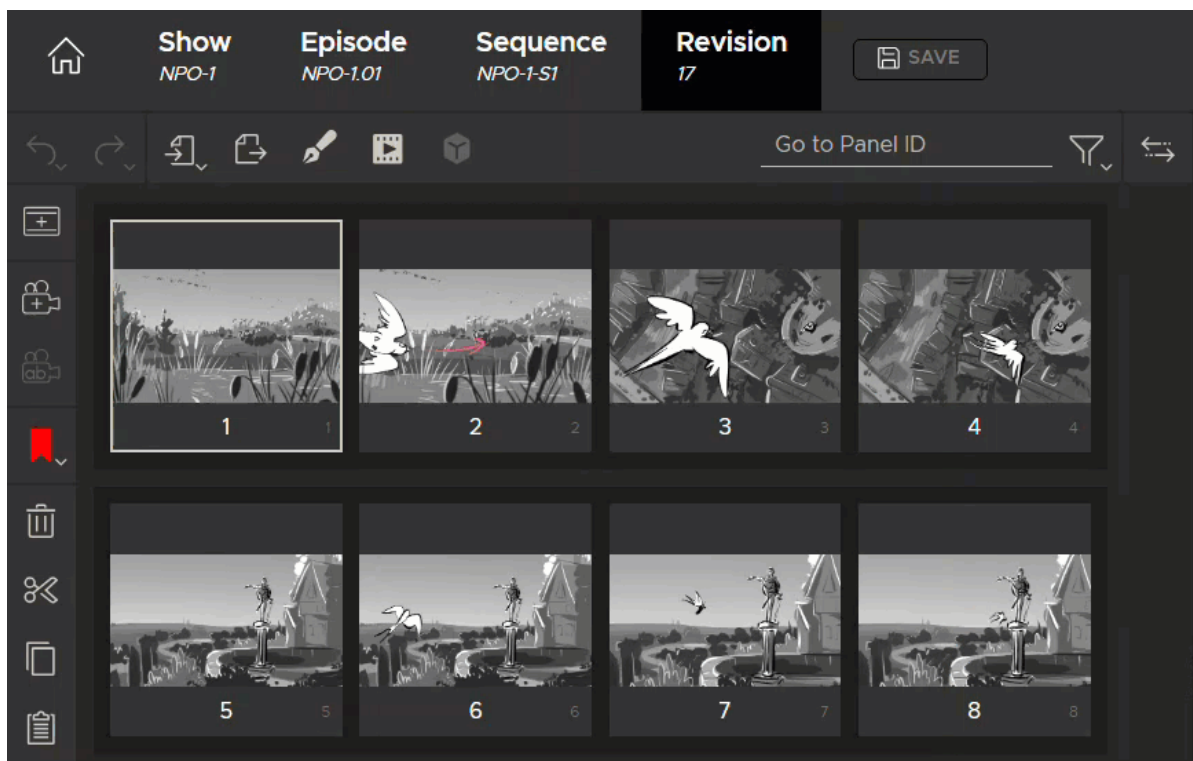
1. Drag the new file over the current panel.
2. The Version up icon will appear.



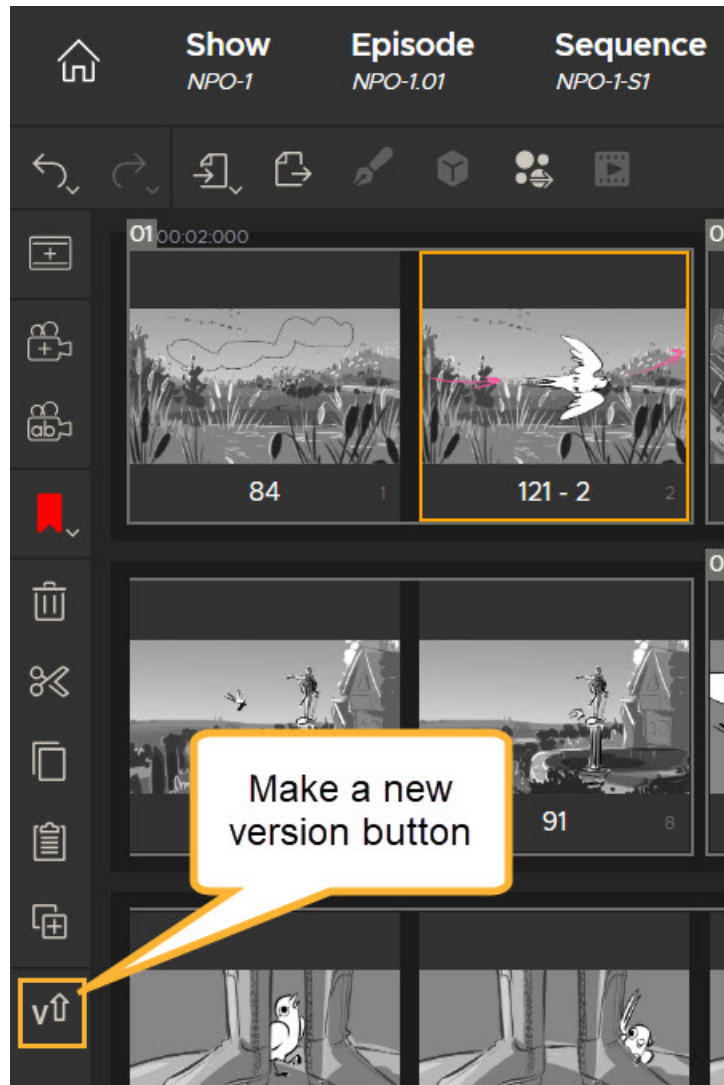
3. Drop the new artwork onto the panel. The Arrow icon will fill in when the new artwork is loading.
4. The revised panel becomes the selected panel after the drop is completed.

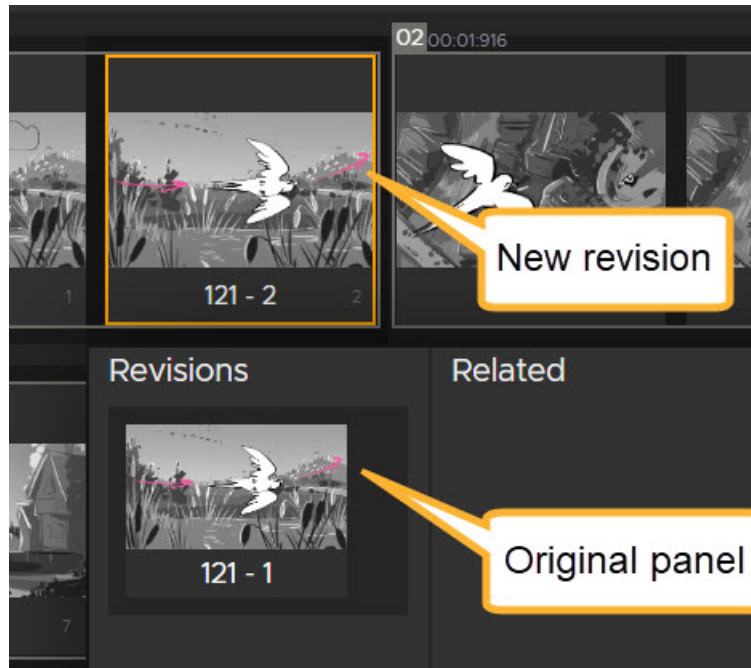


Note: If a panel is errored, you can still version it up by dropping a new file on top of it. This creates a new revision, and the errored panel remains part of the older revision history.



You can also use the Make a new version button, This will create a new version of your panel, based on the current version. This will duplicate your current panel to create a new revision.





Alternatively, you can version up panels by using the Import function to search and import your new panels:

1. In your Revision, select the panels you want to version up.
2. Click and hold the Import icon (.
3. Select **Import Panel Revisions**. This will open a file path window.
4. Select the panels you want to import and click **Open**.
5. The Arrow icon will fill in when the new artwork is loading.
6. The revised panel becomes the selected panel after the drop is completed.

You can version up multiple consecutive panels in a single action - they will update in parallel. If the panels are not sequential, new artwork addition and panel versioning will need to be performed in several separate actions.

When a panel is versioning-up, the versioning-up overlay image will animate to indicate import progress.



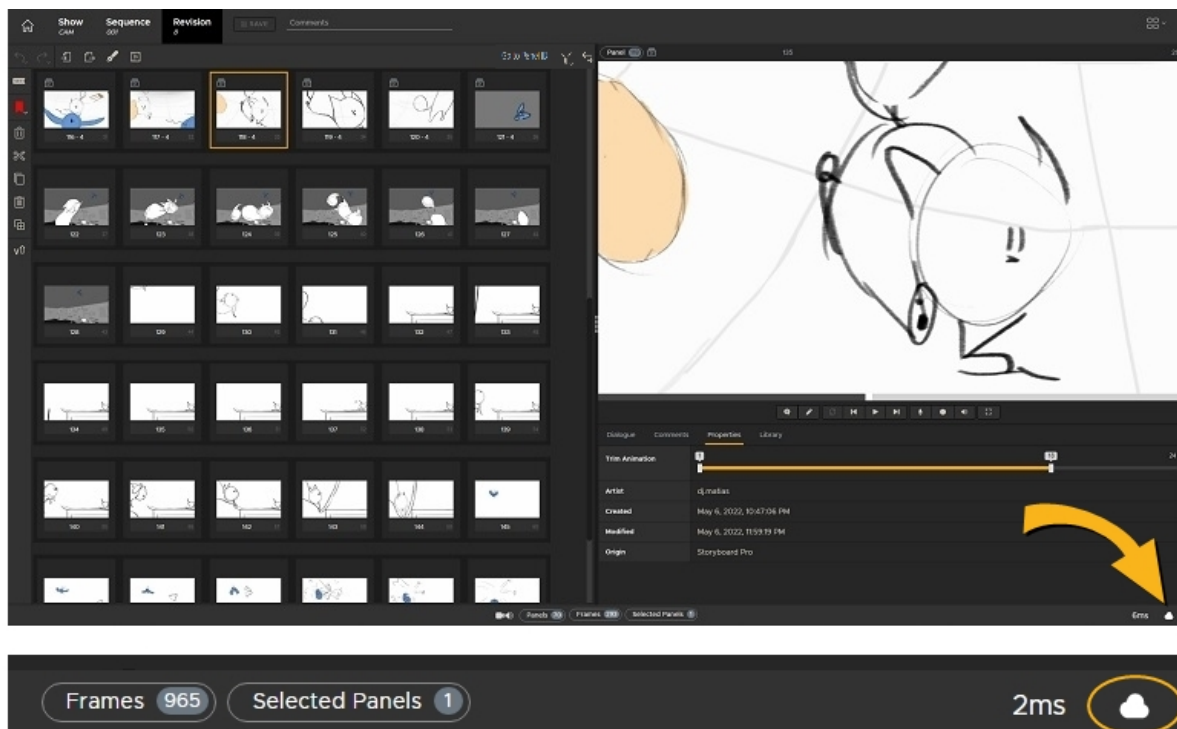
Note: Dragging a panel from the panel Library or from another Flix window will not display the version up option. You can only add as a new panel to your current sequence.

See Import Progress at a Glance in the Transfer Queue

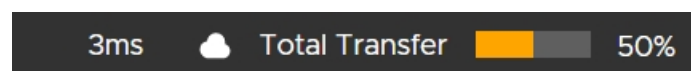
The Flix file transfer system allows you to easily transfer files between Flix Server and Flix Client quickly and reliably. The transfer queue's interface provides up-to-date reporting of the transfer process and allows you

to view and manage transfers more easily.

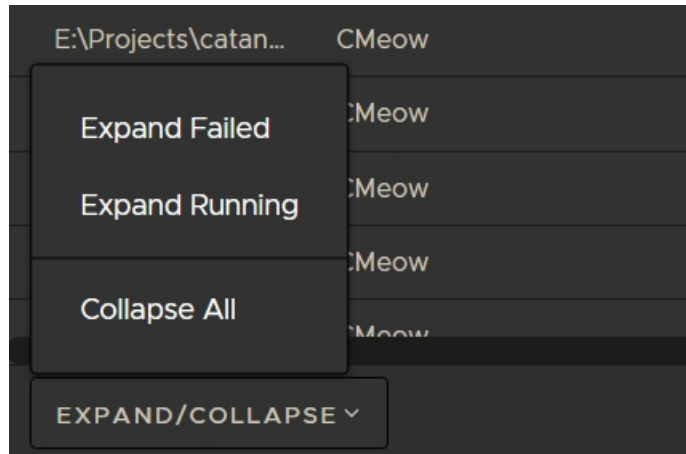
The Transfer Queue provides visual feedback of the files being imported into Flix. Regardless of the method you currently use to import files, the transfer progress of those files is visible by clicking on the  cloud icon at the bottom right of the **Revision** window.



The **Total Transfer** progress is shown when the Transfer Queue is closed, giving you an overall picture of the state of your file transfers.



Clicking the  cloud icon opens the queue interface. At the bottom of the window, the **Expand/Collapse** allows you to quickly check on the status of running and failed imports or collapse all entries in the queue.



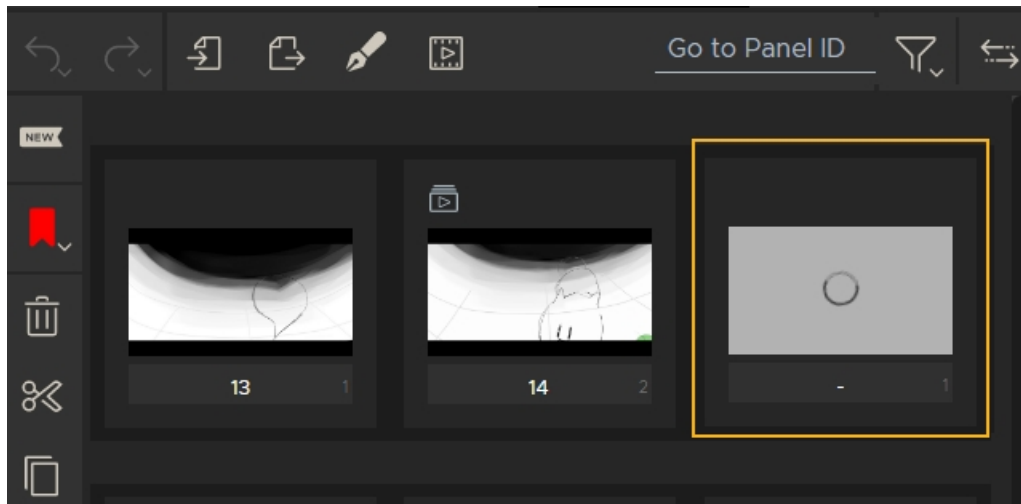
Expanding entries displays detailed information on individual files in the transfer, such as **Show**, **Panel ID**, **Progress**, and **Status**.

File Transfer								
Transfer	Show	Episode	Sequence	Revision	Transfer From	Panel ID	Progress	Status
► May 17, 2022, 11:06	CMeow		Chairman Meow	7	Manual Import		100%	Finished
► May 17, 2022, 11:06	CMeow		Chairman Meow	7	Manual Import		100%	Finished
▼ May 17, 2022, 11:24							70%	Running
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import		100%	Failed
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	236	100%	Finished
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	10%	Running
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue
E:\Projects\cata...	CMeow		Chairman Meow		Manual Import	Pending	0%	Queue

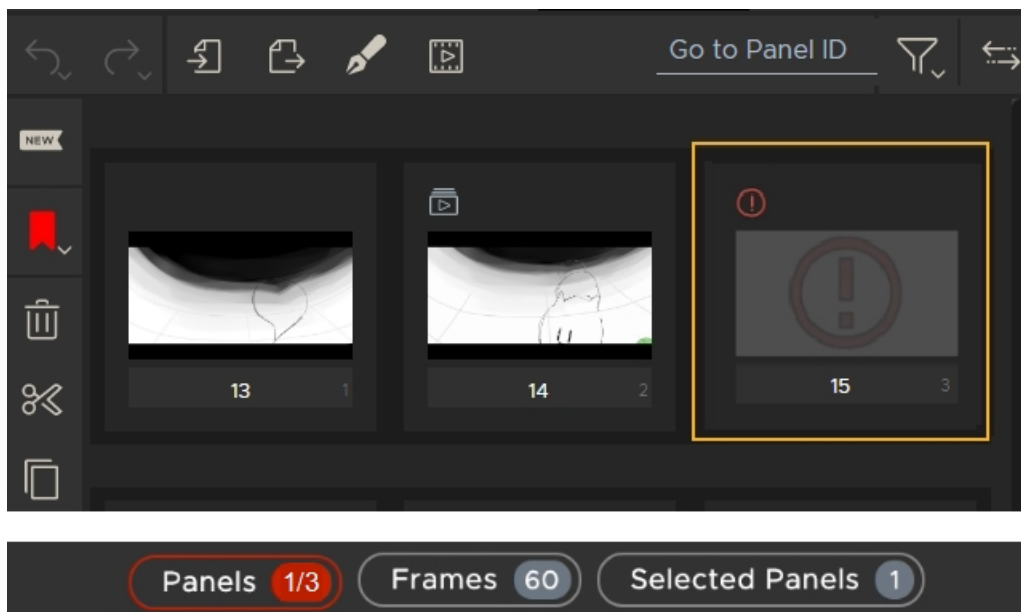


Tip: You can click the **Clear** button to remove **Finished**, **Canceled**, or **Failed** transfers or cancel any current transfers by clicking the **Cancel All** button.

Transfers that display a **Status** of **Finished** may still need time to render the thumbnails associated with the file as this is the last step of the transfer process. Panels in the browser waiting for a thumbnail appear as follows:



Transfers that display a **Status** of **Failed** are displayed in the panel browser with an exclamation point and are highlighted red in the **Panels** count at the bottom of the Flix Client.



If a panel fails to import correctly, you can delete the panel and restart the import or look at the logs to determine the cause of the failure. See [Transfer Logs](#) for more information.



Tip: You can use the  filter button at the top of the Panel Browser to quickly select failed transfers, or if the **Panels** button turns red to indicate failed transfers, click the button to automatically display failed imports.

Keep Working During File Transfers

The Flix transfer system gives you the freedom to do other tasks while the transfer is in progress. As files are imported in the background, you can continue to work, rather than having to wait for the import process to finish. The following functions are all still available during file transfer:

Workspace

Function	Conditions
Resizing, menu operation, Preferences, and so on	Always enabled.

Panel Browser

Function	Conditions
Reorder panels	Always enabled.
Add panels	Always enabled.
Shots	Always enabled.
Highlight panels	Always enabled.
Double click to open a panel in Photoshop	Disabled until thumbnails are created on a per-panel basis, rather than per sequence.

Function	Conditions
	 Note: Only available if Photoshop is set as the Sketching Tool in the Preferences . See Third-Party Apps for more details.
Double click to send a sequence revision to Storyboard Pro	Disabled until thumbnails are created on a per-panel basis, rather than per sequence.  Note: Only available if Storyboard Pro is set as the Sketching Tool in the Preferences . See Third-Party Apps for more details.
Delete panels	For completed transfers: Enabled, but leaves orphan assets in the DB. For Incomplete transfers: The user can only delete panels when the ID is created. The delete action cannot be undone.  Note: The incomplete transfer for deleted panels continues in the background.
Cut panels	Disabled until all panel IDs are created.
Copy panels	Disabled until all panel IDs are created.
Paste panels	Disabled until all panel IDs are created.
Duplicate panels	Disabled until all panel IDs are created.
Create a new version	Disabled until all panel IDs are created.

Top Toolbar

Function	Conditions
Undo/Redo	Disabled until the transfer batch is completed. Enabled for editing only.
Import	Enabled, the user can always add to the transfer queue.
Export	Disabled until all transfers are complete.
Publish	Disabled until all transfers are complete.
Open in Sketching app	Disabled until thumbnails are created on a per-panel basis, rather than per sequence.  Tip: With multiple panel selections if one of the panels doesn't have the thumbnail yet, no panels open until all of the selected panels have thumbnails.
Compare tool	Disabled until panel IDs are created.
Panel filter	Always enabled.

Viewer Tabs

Function	Conditions
Dialogue	The input box is disabled until selected Panel ID is created.
Comments	The input box is disabled until selected Panel ID is created.
Properties	Trim animation is disabled until selected Panel ID is created and thumbnail generated.
Library	Always enabled.

In addition to the Panel Browser functionality, Flix's transfer system allows you to leave the revision that is importing files and open other sequences or shows, ensuring there is no wasted downtime waiting for files to import. The **File Transfer** window is always available in any revision of any show that you are working on, and display any current and past imports that have not already been cleared from the queue.

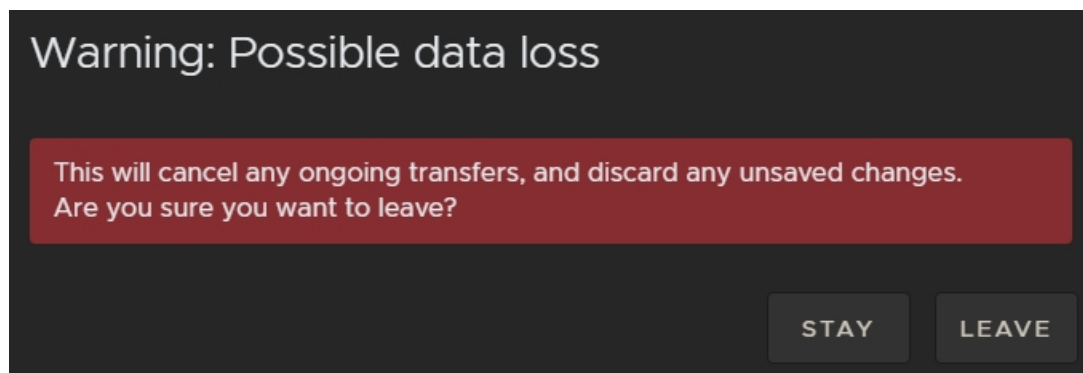
Actions That Stop File Transfers

Canceling imports using the **Cancel All** button, stops transfers manually. There are also a number of actions that cancel transfers that aren't so obvious:

- Closing the Flix Client window,
- Logging out of Flix,
- Exiting Flix,



Note: Closing, logging out, and exiting Flix during a transfer displays a warning. Click **Stay** to avoid canceling transfers and possible data loss. If you select **Leave**, any ongoing transfers are stopped.



- Reloading the Revision to which files are being transferred, and
- Flix closing unexpectedly due to an error.

Preview Artwork and Movies from Panels in a Sequence

You can set the default preview setting in the **Panel Browser** section of preferences to show either the full animation or the first frame only. You can quickly change this setting on each individual panel without having to go back to preferences, but any panel you override in the browser is no longer affected by changes to the **Animated Panels** preference.

Animated panels are set to **Preview all frames** by default, but you can change the default behavior to **Preview the first frame** as follows:

1. On Windows, navigate to **File > Preferences** or press **Ctrl+,** (comma) to open the **Preferences,**
OR

On macOS, navigate to **Flix > Preferences** or press **Cmd+,** (comma) to open the **Preferences.**

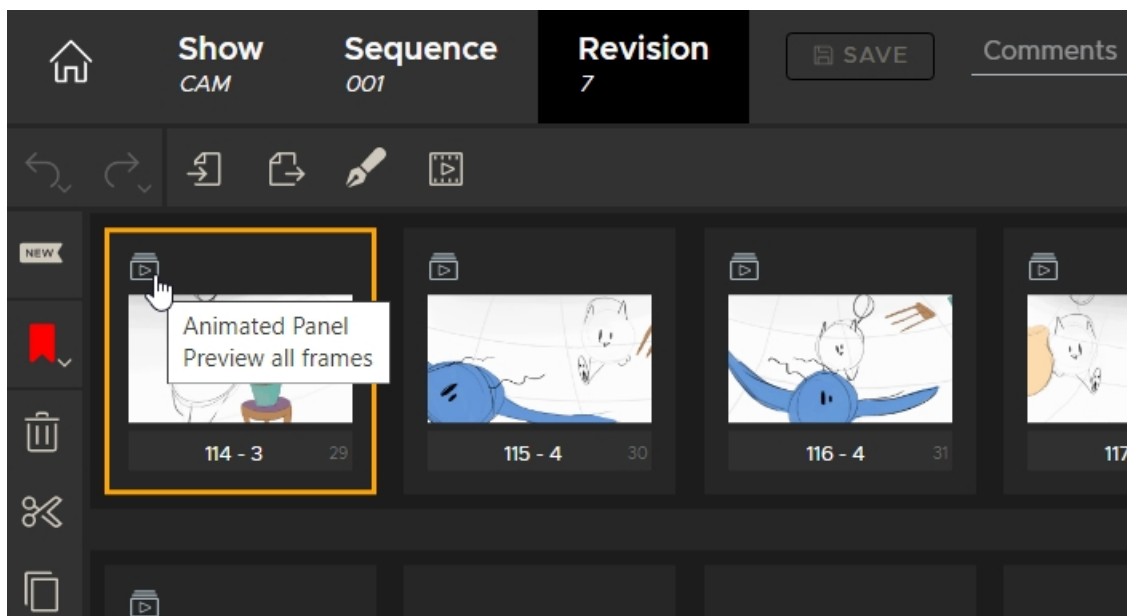
2. In the **Preferences** dialog, navigate to **Panel Browser.**
3. Set the default preview behavior for animated panels:

Preview all frames - Animated panels show all available frames in the panel viewer.

Preview first frame - Animated panels only show the first frame in the panel viewer and the **Trim Animation** control is disabled.

To manually override a panel's preview settings:

In the **Panel Browser,** click the icon in the top left to select the preview mode for that panel.

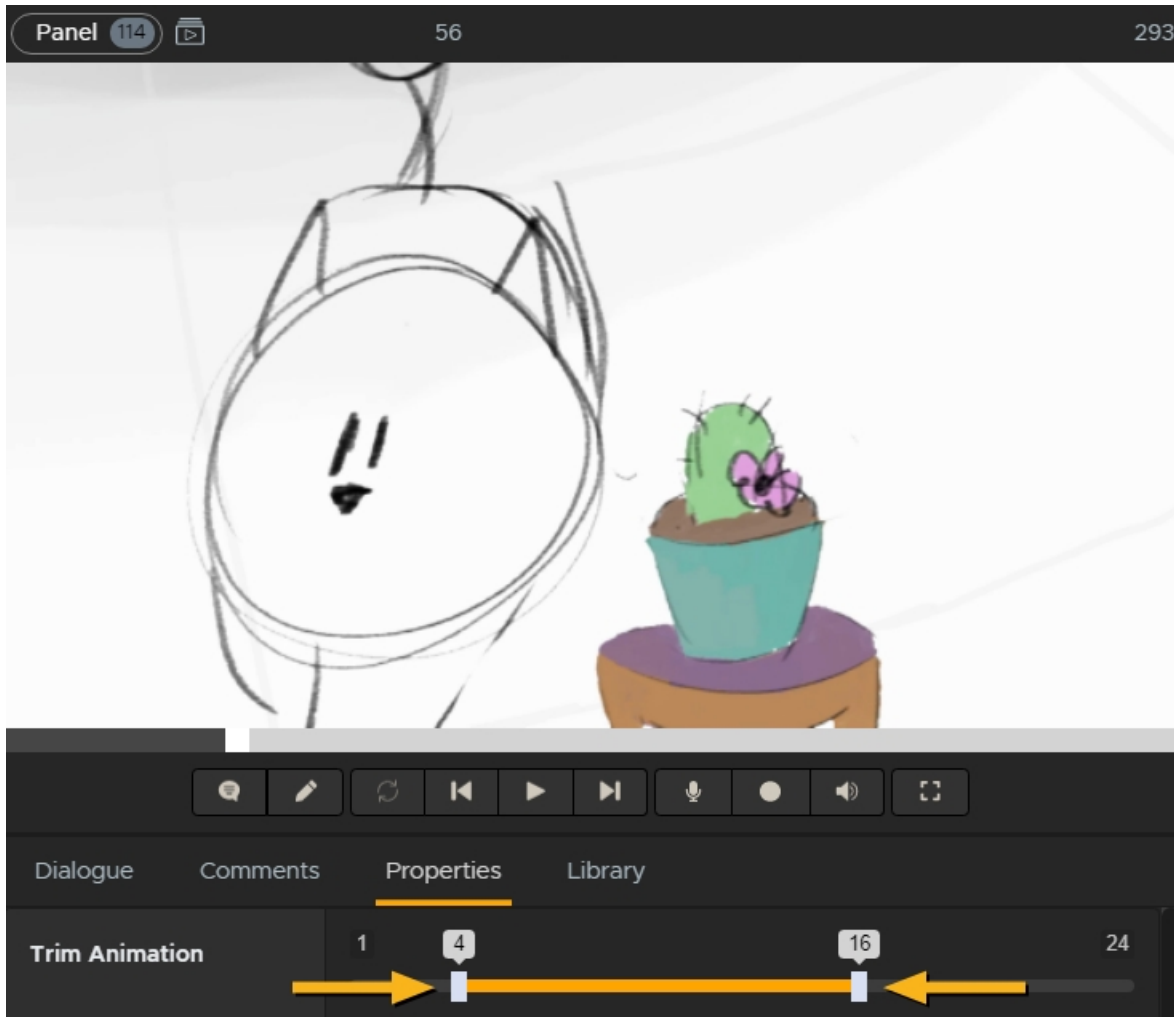


-  **Preview all frames** - Animated panels show all available frames in the panel viewer.
-  **Preview first frame** - Animated panels only show the first frame in the panel viewer and the **Trim Animation** control is disabled.



Note: Changing the default preference does not affect individual panels that you have set manually.

You can trim panels set to **Preview all frames** using the **Trim Animation** slider. Adjusting the in and out points causes the animation to play between only those frames.



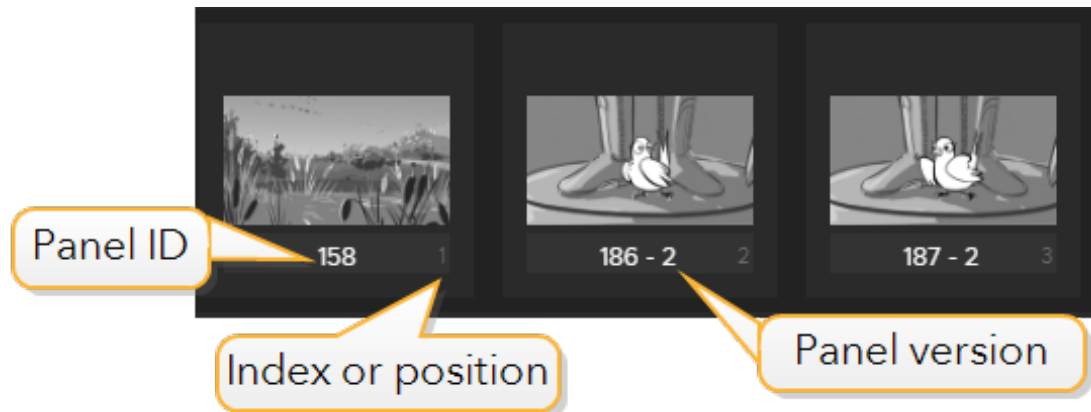
This trimmed animated panel only plays between frames 4 and 16.

Making Changes to the Edit

In Flix, you'll be working primarily in the Panel Browser. The following video gives a brief overview of how to navigate and work with your sequence in the Panel Browser.

Editing Actions:

- Importing panels. For more information, please refer to [Importing Panels into Flix](#).
- Each panel displays a unique Panel ID and Index, or position number. If the panel has been updated, a new version is created and numbered.



Note: You can swap the position of each panel's unique ID with its Index number. See [Swap Panel ID with Index](#) in Preferences.

- Move panels around by selecting one or more, and dragging them where you want to place them. Hold **Shift** and click to select a sequence of panels, or **Ctrl/Cmd** and click to select individual panels.
- Remove panels from your current edit by clicking the Trashcan icon in the Edit toolbar.
- Hide panels by clicking on the panel and toggling **Visibility** in the **Properties** tab.
- Re-use panels by using the Copy and Paste buttons in the Edit toolbar. This creates new panels re-using the same panel, timing, and dialogue.
- Using the annotation tool. For more information, see [Annotating Panels](#).
- Adding and versioning dialogue.

You can drag and drop new panels over existing ones to update them. View [Panel Versioning](#) to learn more.



Tip: Duplicate selected panels with the  **Duplicate** button.

Editing Panels Already in Flix

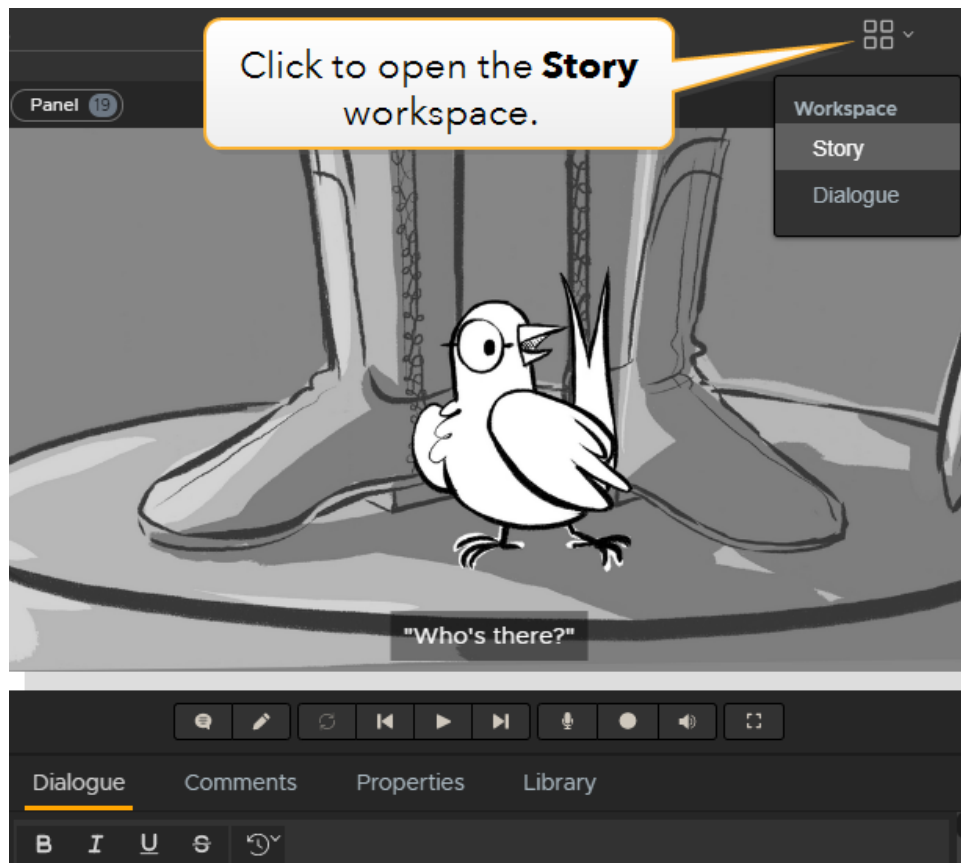
To make changes to panels in your edit (sequence revision), you can open and edit them in third-party applications. See the [Flix for Story](#) section.

Adding or Editing Dialogue

You can easily add dialogue to your panels in both the **Story** and **Dialogue** workspaces. Dialogue is displayed as subtitles in the Player, which you can turn on and off.

Method 1

Flix's **Story** workspace (default) contains the Dialogue controls in the Panel Properties pane in the **Dialogue** tab.



To add and edit dialogue:

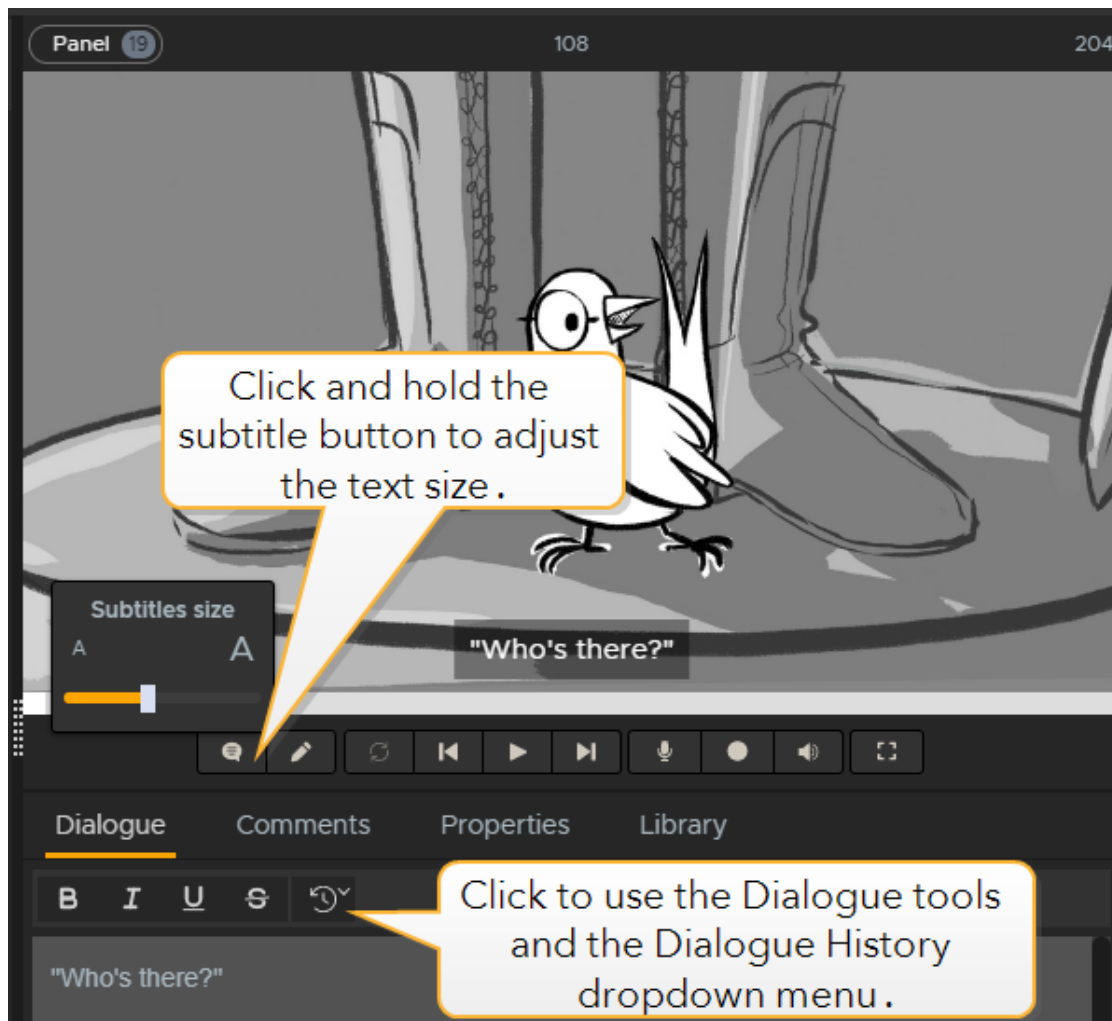
1. Select the required panel and enter text in the Dialogue box. Press **Enter** to add new lines.

This adds dialogue to the selected panel, which is displayed as subtitles in the Player.



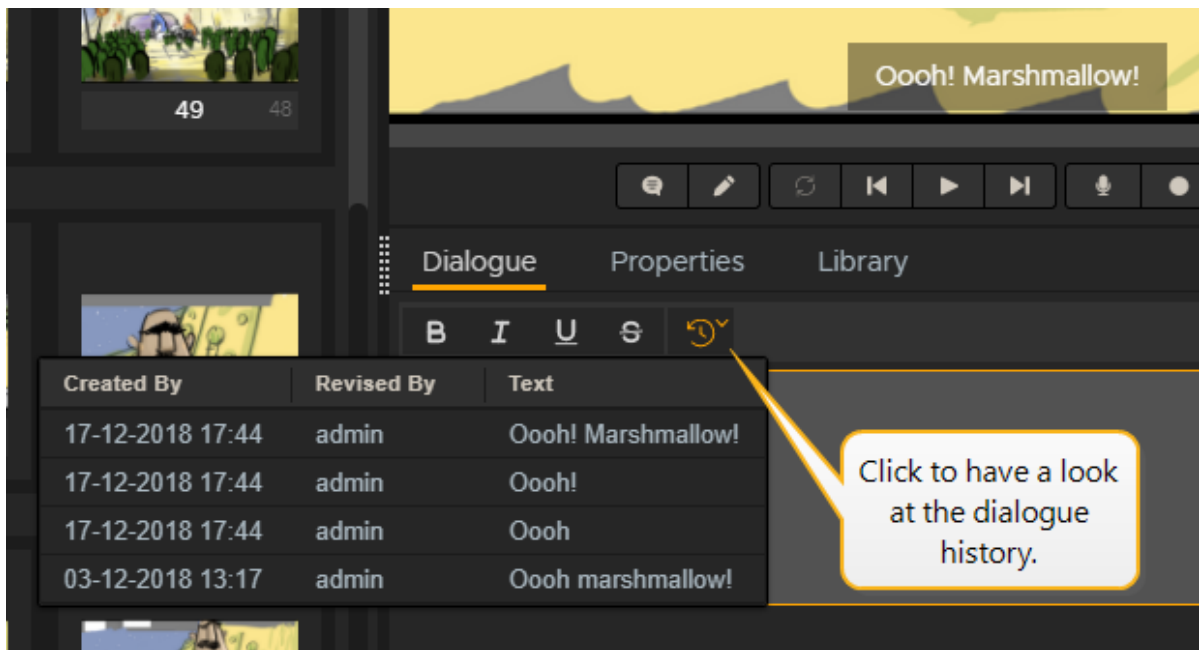
Note: If you select multiple panels, the same dialogue is added to all selected panels.

2. Use the Dialogue tools to modify the formatting of your text.
3. Click the Toggle Subtitles button to turn the subtitles on and off. Click and hold on the button to display the text size slider control.



Tip: Use the Undo and Redo buttons, located under the breadcrumb, to undo or redo your recent actions.

4. Click **Save** to save all new dialogue.
5. Keep track of the dialogue history by clicking on the Dialogue History dropdown menu.



Method 2

In the **Story** workspace at the top-right corner, click the Switch Workspace button and select **Dialogue** workspace. The **Dialogue** workspace contains a Panel Browser and a Dialogue pane.

To add and edit dialogue:

1. Select the required panel and enter text in the Dialogue box. Press **Enter** to add new lines.

This adds dialogue to the selected panel.

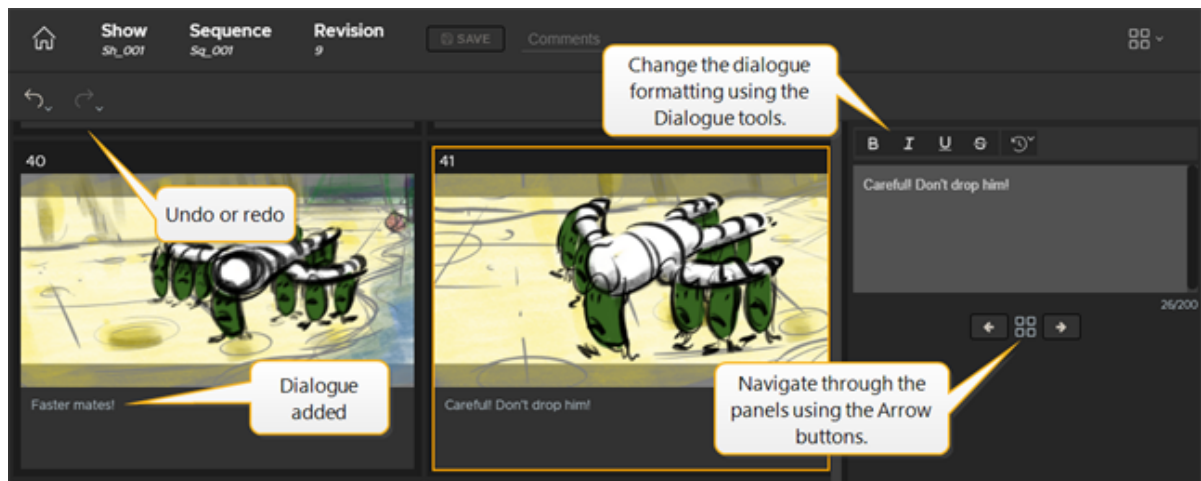


Note: If you select multiple panels, the same dialogue is added to all selected panels.



Tip: Press **Tab** to move to the next panel and **Shift+Tab** to move to the previous panel when entering dialogue on the Dialogue workspace.

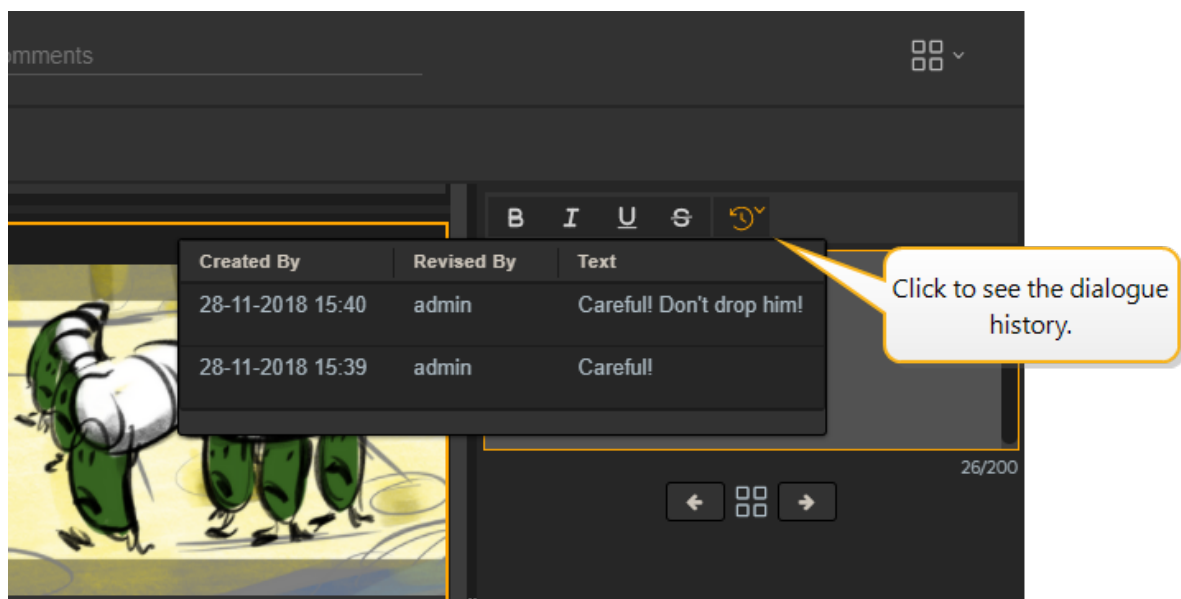
2. Use the Dialogue tools (Bold, Italics, Underline, and Strikethrough) to modify the formatting of your text.



3. **Save** your sequence revision to save all new dialogue.



Note: Keep track of the dialogue history by clicking on the Dialogue History dropdown menu.



Recording a Pitch

Periodically, you may want to pitch sequences to others involved in the project to present your work. Pitching allows you to do a rough timing for your sequence, for example depending on the action you may want to stay longer on one panel.

You can access the Pitch workspace through the Player in the Panel Properties pane.

1. If you want to record audio while pitching, navigate to **File > Preferences > Audio** and select an **Input Device**. If you already have audio in your sequence and don't want to override it, in the Player click the Microphone button to disable it.



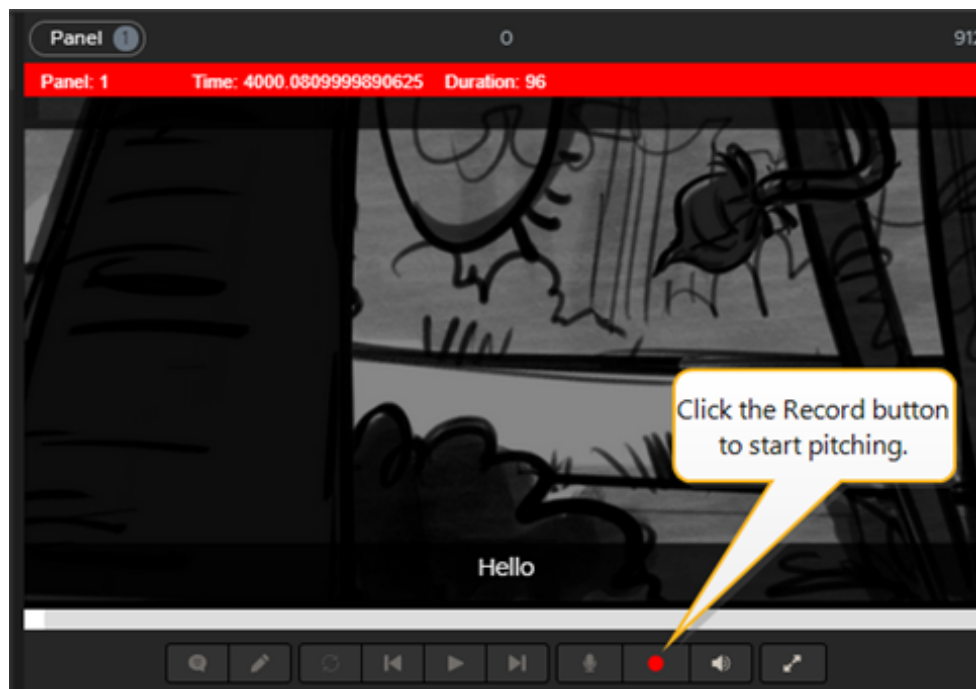
Note: When pitching with audio, use the Volume button to make sure the volume is set properly.



Tip: You can switch to Full Screen mode by clicking the Full Screen button. Press **Esc** to exit the Full Screen mode.

2. Click the Record button to start pitching.

In the red bar at the top of the Player, the Time and the Duration timers are running.



3. As soon as you reach the required duration, press the right arrow key to move to the next panel.
By switching to the next panel, you stop the recording of the duration for the current panel and start the recording for the next one.
4. Keep recording the duration for each panel until the end of the sequence or click the Record button again to stop recording.

In the Panel Properties pane, select the **Properties** tab to check the updated duration for your panels. You can also change the duration in the **Properties** tab directly by entering a new duration or using the arrows.



Tip: When playing the sequence, press **A** to add an audience at the bottom of your panels.

Flix for Editorial

Flix allows for roundtripping with your editorial department. This means an editor can make timing changes, rearrange shots, add camera moves and audio to the Flix sequence, as well as adding in external media such as Adobe After Effects compositions or a Maya playblast. All these changes to the sequence can then be imported back to Flix and appear as a new Revision.

Setting up Flix with Adobe Premiere

1. To set Adobe Premiere as your default editorial tool, go to **File > Preferences > Third Party Apps > Editorial Tools**.
2. Click the **Editorial Tool** dropdown menu and select **Adobe Premiere**.
3. Use the controls on this page to set your preferences for exporting **.xml** files to Premiere. See the table below for details.
4. In the **Publish and Export** section, enter the folder destination for your publishes into the **Directory** path or click **Browse** and then select a folder. For example, T:\flix_publishes[show_tracking_code][sequence_tracking_code].

Premiere Setting	Description
Use Flix Timing	Toggle ON to send every panel as a clip retaining its duration set within Flix. This is useful to retain the timing established by a recorded pitch, saving you time and avoiding redoing work. See Preserve Sequence Timings from Flix for more details.

Premiere Setting	Description
	 Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Premiere. If you disable this control, you can only send .mov files to Premiere and the Panels with Camera Move control is disabled.
Panels with Camera Move	Choose whether to deliver panels with camera moves from Storyboard Pro to Premiere as a .png file as a QuickTime .mov file, or send the animated panel as a full canvas .png still with keyframes. Images with Keyframes - this option preserves the camera move keyframes from Storyboard Pro so that you can edit them in Premiere and send them back to Flix without having to reproduce the effect. Rendered Animations - the camera moves from Storyboard Pro are burned into a .mov clip so you can't adjust the keyframes.  Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Premiere. If you disable this control, you can only send .mov files to Premiere and the Panels with Camera Move control is disabled.
Mark In	Sets the frame on which to set the Mark In

Premiere Setting	Description
	point. The default is 120 (5s in for a 24 fps show).
Mark Out	Sets the frame on which to set the Mark Out point. The default is 360 (15s in for a 24fps show). When Use Flix Timing is enabled, mark out is automatically set based on each panel's duration.  Warning: Setting your Mark Out point to a value lower than the Mark In point may create an unreadable .xml file.
Enable Audio	Controls whether Flix audio is sent to Premiere or not.
New Clip Color	Sets the color of new clips in your sequence.
Marker Type	Choose whether Flix sends and receives Timeline or Clip markers to and from Premiere. Timeline markers apply to a particular timestamp in the sequence and appear on the timeline. Clip markers apply to a whole clip within a sequence and appear at the beginning of the clip.  Note: The Marker Type you choose means that Flix only reads that type. For example, if you have Clip selected and Editorial adds Timeline markers in Premiere, they are ignored by Flix.



Note: Make sure you have write permission to the publish directory. Contact your Systems Administrator for more information.



Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **\${HOME}** on macOS.

Setting up Flix with Avid Media Composer

1. To set Avid Media Composer as your default editorial tool, go to **File > Preferences > Third Party Apps > Editorial Tools**.
2. Click the **Editorial Tool** dropdown menu and select **Avid Media Composer**.
3. Use the controls on this page to set your preferences for exporting **.aaf** files to Media Composer. See the table below for details.
4. In the **Publish and Export** section, enter the folder destination for your publishes into the **Directory** path or click **Browse** and then select a folder. For example, `T:\flix_publishes[show_tracking_code][sequence_tracking_code]`.
5. If you want your published files to follow a specific naming convention, you can set that in the **Editorial Sequence Name** and **Editorial Clip Name** fields. Choose tags from the table below to construct a filename convention.



Note: Make sure you have write permission to the publish directory. Contact your Systems Administrator for more information.



Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **\${HOME}** on macOS.

Avid Setting	Description
Use Flix Timing	Toggle ON to send every panel as a clip retaining its duration set within Flix. This is useful to retain the timing established by a

Avid Setting	Description
	recorded pitch, saving you time and avoiding redoing work. See Preserve Sequence Timings from Flix for more details.  Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Avid. If you disable this control, you can only send .mov files to Avid and the Panels with Camera Move control is disabled.
Panels with Camera Move	Choose whether to deliver panels with camera moves from Storyboard Pro to Avid Media Composer as a .png file with the Motion Effect applied, or as a QuickTime. Images with Keyframes - this option preserves the camera move keyframes from Storyboard Pro so that you can edit them in Avid and send them back to Flix without having to reproduce the effect. Rendered Animations - the camera moves from Storyboard Pro are burned into a .mov clip, so you can't adjust the keyframes.

Avid Setting	Description
	 Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Avid. If you disable this control, you can only send .mov files to Avid and the Panels with Camera Move control is disabled.
Mark In	When Enable Mark In is ON, sets the frame on which to set the Mark In point. The default is 120 (5s in for a 24 fps show).
Mark Out	When Enable Mark Out is ON, sets the frame on which to set the Mark Out point. The default is 360 (15s in for a 24fps show).  Warning: Setting your Mark Out point to a value lower than the Mark In point may create an unreadable .aaf file.
Enable Audio	Controls whether audio is available in Avid Media Composer or not.
New Clip Color	Sets the color of new clips in your sequence.
Source Media Duration	Sets the maximum number of frames available for editing still media in Media Composer. Default is 480 frames.  Note: Mark In and Mark Out must fall within the frame range set here.

Avid Setting	Description
Color Range	Sets either Full or Legal range color for publishes to Avid.  Note: The default color range is set to Full.
Codec	Specifies the type of Avid video codec used for exporting to Media Composer: DNxHD 36 (DNxHD LB) DNxHD 45 (DNxHD LB) DNxHD 115 (DNxHD SQ) DNxHD 120 (DNxHD SQ) DNxHD 145 (DNxHD SQ)  Note: Higher data bit rates generally produce higher quality images, but the files exported require more storage space and take longer to transfer.

Preserve Sequence Timings from Flix

Flix integration with Storyboard Pro allows you to import camera moves and panel durations and pass them on to Editorial, saving you time and giving your editor a head start. Of course, some editors prefer to receive all panels with their default durations, so that's why in Flix you can control this behavior with a single preference.

- To send every panel as a clip retaining its duration set within Flix, go to **File > Preferences > Third Party Apps > Editorial Tools** and then enable the toggle for **Use Flix Timing**.
- Disable **Use Flix Timing** to send every panel with its default duration.

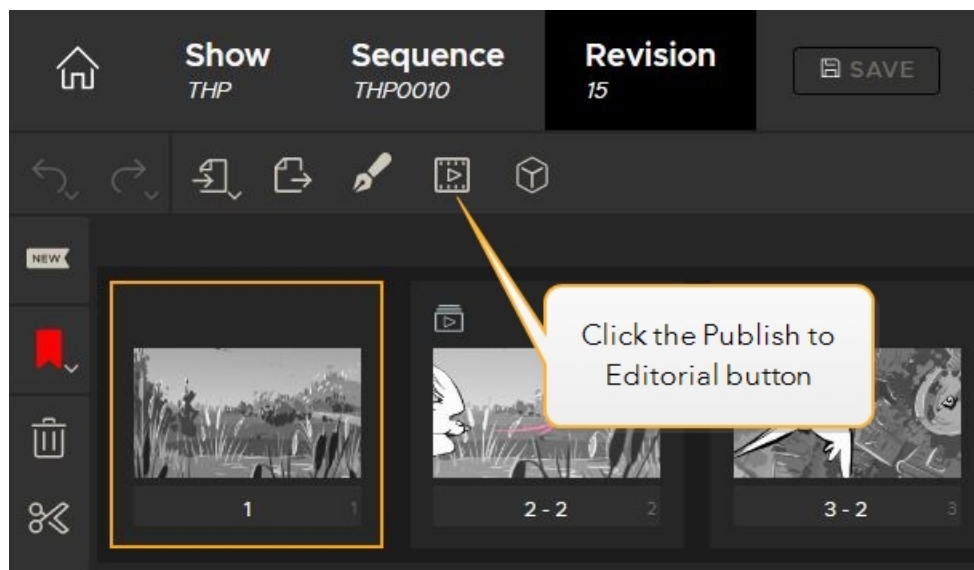


Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Editorial. If you disable this control, you can only send **.mov** files and the **Panels with Camera Move** control is disabled.

Publishing a Sequence to Editorial

The following steps apply to both Adobe Premiere and Avid Media Composer. Publishing to Premiere creates **.xml** files and publishing to Media Composer creates **.aaf** files.

1. On the main toolbar of your Revision, click the **Publish to Editorial** button.



The **Publish to Editorial** dialog appears.

Publish to Editorial

Publish Directory

F:\Flix 7

BROWSE

Publish with Camera Move

Rendered Animations

When "Use Flix Timing" is disabled, Panels with camera moves will be sent as rendered animations.

[More information](#)

Generate Contact Sheet

☒

Single Screen (Landscape)

Generate Dialogue File

☒

Comment

Publish Directory:

F:\Flix 7

CANCEL

PUBLISH



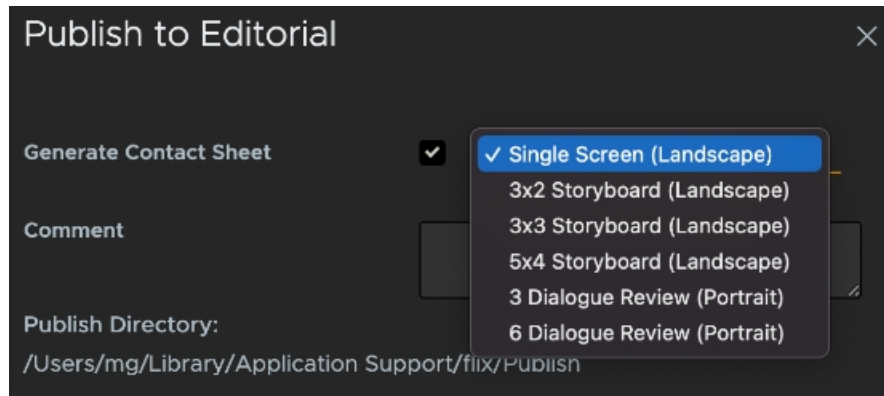
Note: If you attempt to publish to a directory for which you do not have sufficient permissions, a warning appears. You will need to ask a Systems Administrator to ensure the permissions on the publish directory to allow you to publish.

- Choose the directory in which you are saving your published file.
- Select how you would like Flix to export panels with camera moves.

Images with Keyframes - this option preserves the camera move keyframes from Storyboard Pro so that you can edit them in Avid and send them back to Flix without having to reproduce the effect.

Rendered Animations - the camera moves from Storyboard Pro are burned into a **.mov** clip, so you can't adjust the keyframes.

4. Specify if you want to generate a Contact Sheet of the panels in your Revision by checking **Generate Contact Sheet**. This is saved as a **.pdf** file.
5. Use the dropdown to the right to select a Contact Sheet template to use.



There are several defaults to choose from and administrators can create their own templates. See [Customizing Contact Sheet Templates](#) for more information.

6. Specify if you want to generate files of the dialogue in your Revision by checking on **Generate Dialogue File**. Your dialogue is saved as a **.txt** file for Media Composer and a **.srt** file for Premiere.
7. Enter a **Comment** if needed. This appears in the email notification.
8. Click **Publish**.

The directory to which your **.pdf** and **.xml/.aaf** files have been published opens.

<filename> **_all.xml/.aaf** - This file contains the panel information of your entire sequence as well as any new or existing audio.

<filename> **_new.pdf** - This file (Contact Sheet) contains the panel thumbnails of your entire sequence with **New** labels on panels that have been added or edited since the last publish. The Contact Sheet also contains dialogue. This is only created if you enabled **Generate Contact Sheet**.

<filename> **_new.xml/.aaf** - This file contains only information and audio of panels that have been added or edited since the last publish.



Note: If no new changes have been made to the sequence revision, clicking the Publish to Editorial button opens the directory to which your files have been published. If files are missing from the publish directory, Flix automatically downloads them from the server and stores them in the publish directory.



Note: If a Publish fails, it may be because Flix Server needs access to a font. Check [Running Flix Server](#) for more information.

Filtering a Sequence by Published/Unpublished Panels

To check which panels have previously been included or excluded from publishes to editorial:

1. Click on the  Filter button in the Panel Browser.

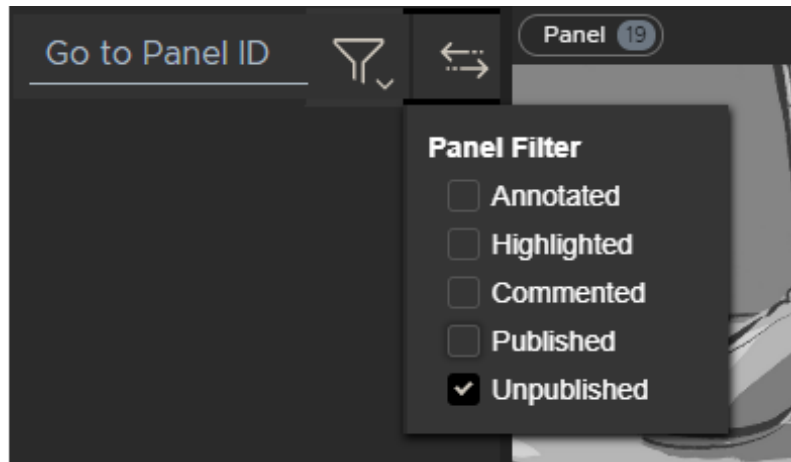
2. Either:

Check the **Published** option to see only panels that have been included in a previous publish to editorial.

OR

Check the **Unpublished** option to see only panels that have not been published to editorial

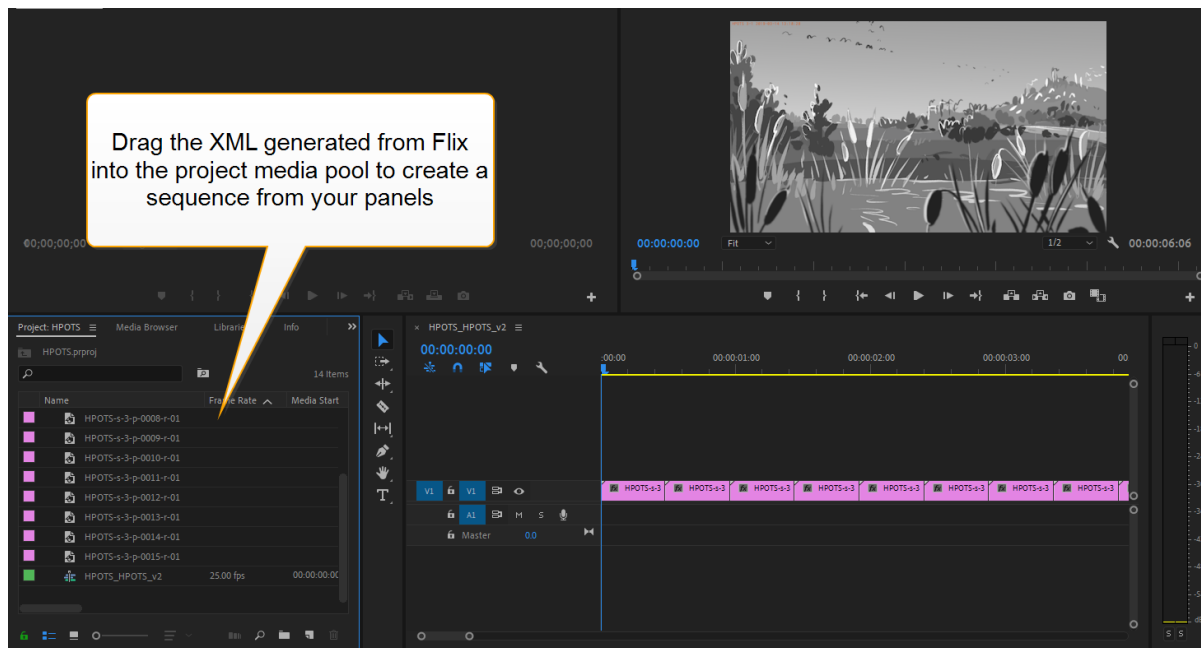
previously.



Importing Your Sequence to Adobe Premiere

1. Open an existing project or create a new one.
2. Drag the **.xml** file into your project window.

A sequence is automatically created using the imported clips. Any new panels are highlighted in your sequence if you have set the **Highlight New Panels** preference in **Preferences > Third Party Apps >**

Editorial Tools > Adobe Premiere.

Note: Recorded pitches embed shot duration metadata in the **.xml** file, so panels appear in the sequence with their recorded timings.

Importing Your Sequence to Avid Media Composer



Article: Have a look at Knowledge Base article [Q100650](#) for more detailed information on Storyboard Pro camera animations in Avid Media Composer.

1. Open an existing project or create a new one.



Note: If you're creating a new project, make sure your media creation settings match the AAF export setting in Flix. For example, DNxHD 36 (DNxHD LB).

2. Drag either the **all.aaf** or the **new.aaf** file into a bin.

If you choose the **all.aaf**, every panel is in the generated sequence. If you choose the **new.aaf** file, only new Flix panels will appear in your bin. New clips are highlighted (green is the default color but you can change this in [Flix Preferences](#)) and a sequence is automatically created using the imported clips.



Note: AAFs generated by Flix only contain new media. This is to avoid duplicate media files from previously imported AAFs. If you import an **all.aaf** of a sequence you have worked on previously, Media Composer displays an alert saying some of the embedded media failed to import. This is expected and only the new panels appear in your bin. The other clips appear as offline media which can be relinked using Media Composer's relink tool.



Tip: When importing a AAF from Flix, you can determine where new panels belong in an existing edit by using the **Flix Sort** and **Flix Sort per Revision** column options in your bin.

Importing Your Dialogue File as Captions in Avid Media Composer

Once you have exported your dialogue out of Flix as a **.txt** file, you can import that file to Avid as captions.



Tip: You can export your dialogue from your Flix revisions by using either **Publish to Editorial**, or the **Export from Flix** button.



Video: Watch [this video](#) to learn more about using Flix's dialogue text file for captions in Avid.

To import caption data in Avid:

1. Create a new **Video Track**.
2. Open the **Effect Palette** and click the **Filters** tab, then search for **SubCap (Generator)**.
3. Drag the **SubCap (Generator)** onto your new video track.
4. Open **Tools > Effect Editor**.
5. Under **Caption Files**, click **Import Caption Data**.

A file browser appears for you to select your caption data.

6. Use the file browser to select your exported dialogue file and click **Open**.

A clip is created on your video track for each dialogue item which can then be edited independently from panels in your edit.

Sending a Sequence Revision back to Flix from Editorial

Once you've finished editing a sequence and adding in external media such as Adobe Premiere compositions or a Maya playblast, you can re-import the sequence in Flix and it appears as a new Revision.

If you've created a composition from multiple panels in the edit, the related panels are also sent to Flix as REF panels. REF panels provide easy access to all the images used in editorial to allow Flix to re-create a composition in the original PSD, improving collaboration and cooperation between Story and Editorial.

When imported into Flix, REF panels are marked with an icon () in the import summary table.



Warning: Do not mix down the composition panels in the edit or Flix will not recreate the REF panels as expected.

See [View Related Panels from Editorial in Flix](#) for more information.

Exporting Your Sequence Revision from Adobe Premiere

Flix requires both a **.xml** file and a **.mov** reference movie file to update your Sequence with a new Revision incorporating the changes made by editorial.

To create the **.xml**:

1. Select your sequence and click **File > Export > Final Cut Pro XML**.
2. Choose a location to save your file.
3. Click **Save**.

To create the **.mov**:

1. Click **File > Export > Media**.
2. Select a codec that uses QuickTime, such as Apple ProRes.



Warning: Check that the framerate and aspect ratio of your export matches those of your Show in Flix, otherwise your sequence will not be imported.

3. Choose a location to save your file.
4. Click **Save**, then **Export**.



Note: Markers created in Adobe Premiere display in the Flix panel browser at the beginning of the corresponding shot.

Exporting Your Sequence Revision from Avid Media Composer



Article: Have a look at Knowledge Base article [Q100650](#) for more detailed information on Storyboard Pro camera animations in Avid Media Composer.

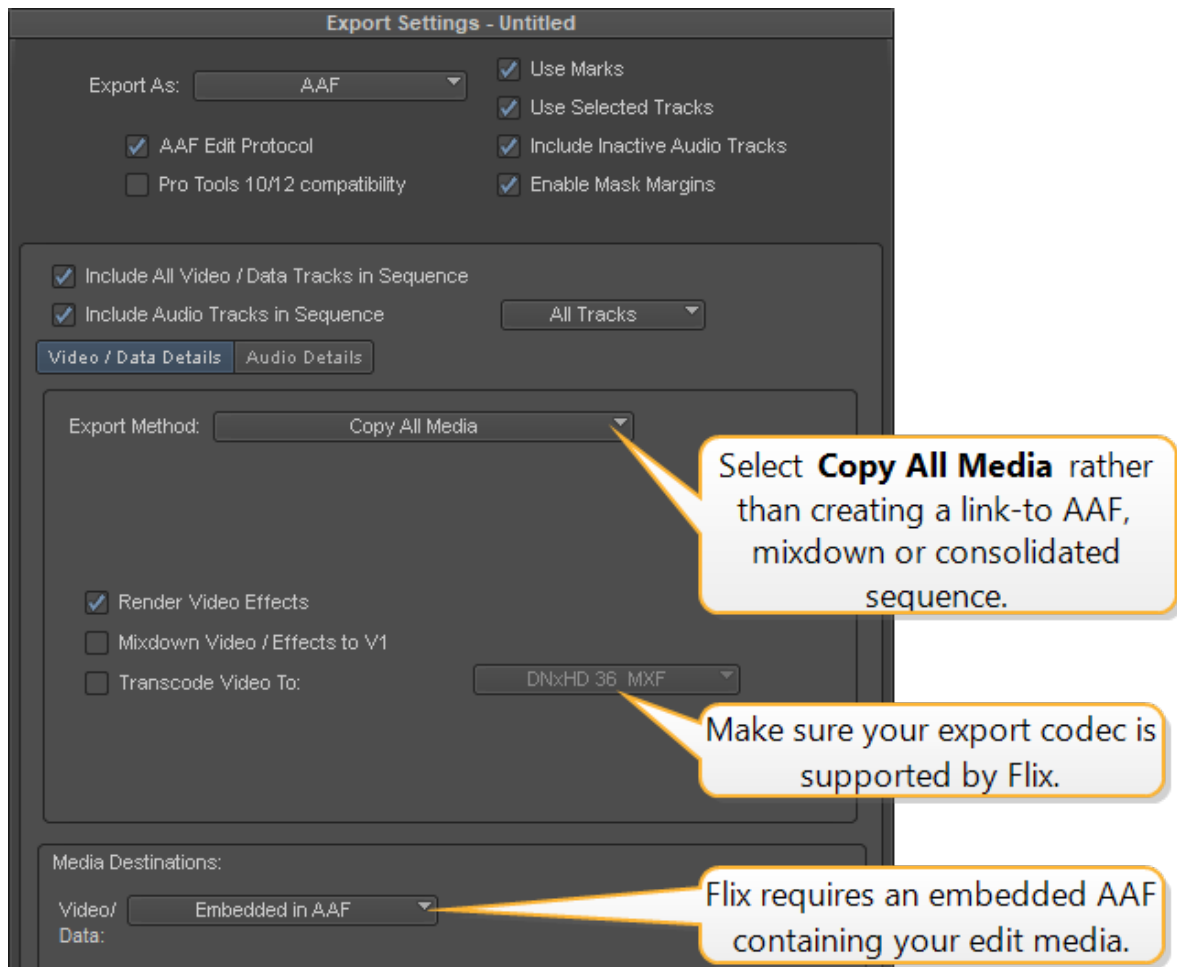
To export an **AAF** of your edit back to Flix:

1. Right-click on your sequence and select **Output > Export to File** or right-click on the Sequence Viewer and select **Export...**

A Save dialog opens.

2. Click **Options** to open the **Export Settings** window.

Set the **Video/Data Details** first.



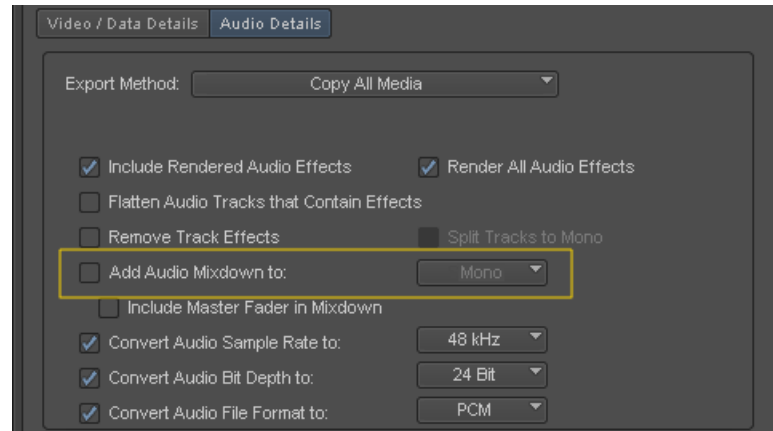
The **Audio Details** tab controls how audio is exported from Avid. Audio can severely impact the file size of AAFs exported from Avid to Flix.

You can:

- Disable **Add Audio Mixdown** (recommended) - manually mixdown the audio before export to AAF.



Note: We recommend manually mixing down audio in Avid before exporting the AAF back to Flix. See the [Audio Mixdown in Avid](#) Knowledge Base article (Q100581) for more information.

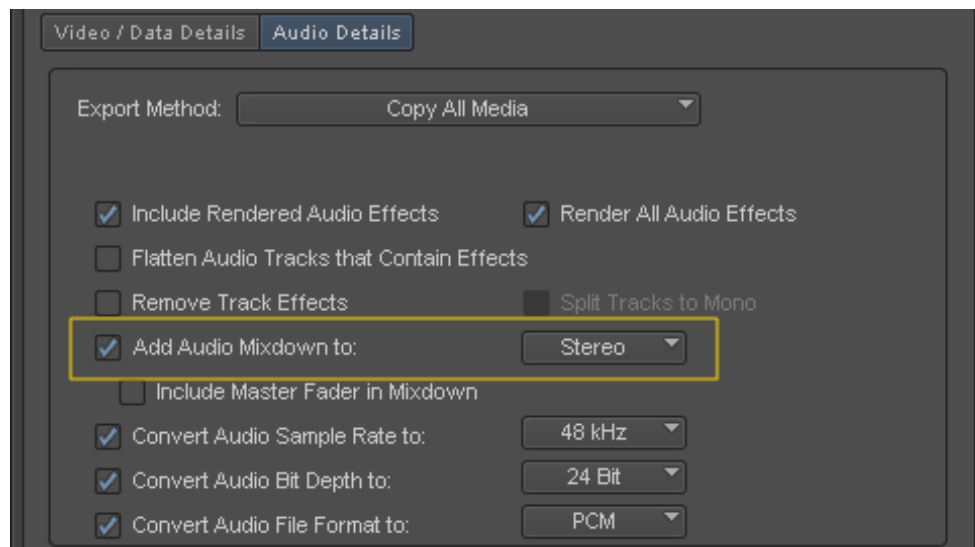


OR

- Enable **Add Audio Mixdown** - use Avid to mixdown the audio automatically before export to AAF.



Warning: Enabling **Add Audio Mixdown** includes all the source media for all audio clips in the edit, which can result in long processing times.



3. Click **Save** to finalize the video and audio export settings.
4. Choose a location to save your file and click **Save**.



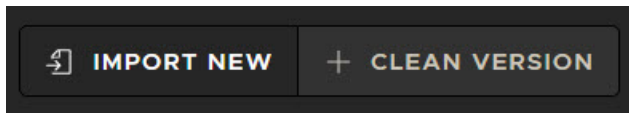
Note: Markers created in Avid Media Composer display in the Flix panel browser at the beginning of the corresponding shot.

Importing Your Sequence in Flix

1. In the breadcrumb, click **Sequence**.

This takes you to the **Revisions** level.

2. Click the **Import New** button and select the Sequence you want to import.



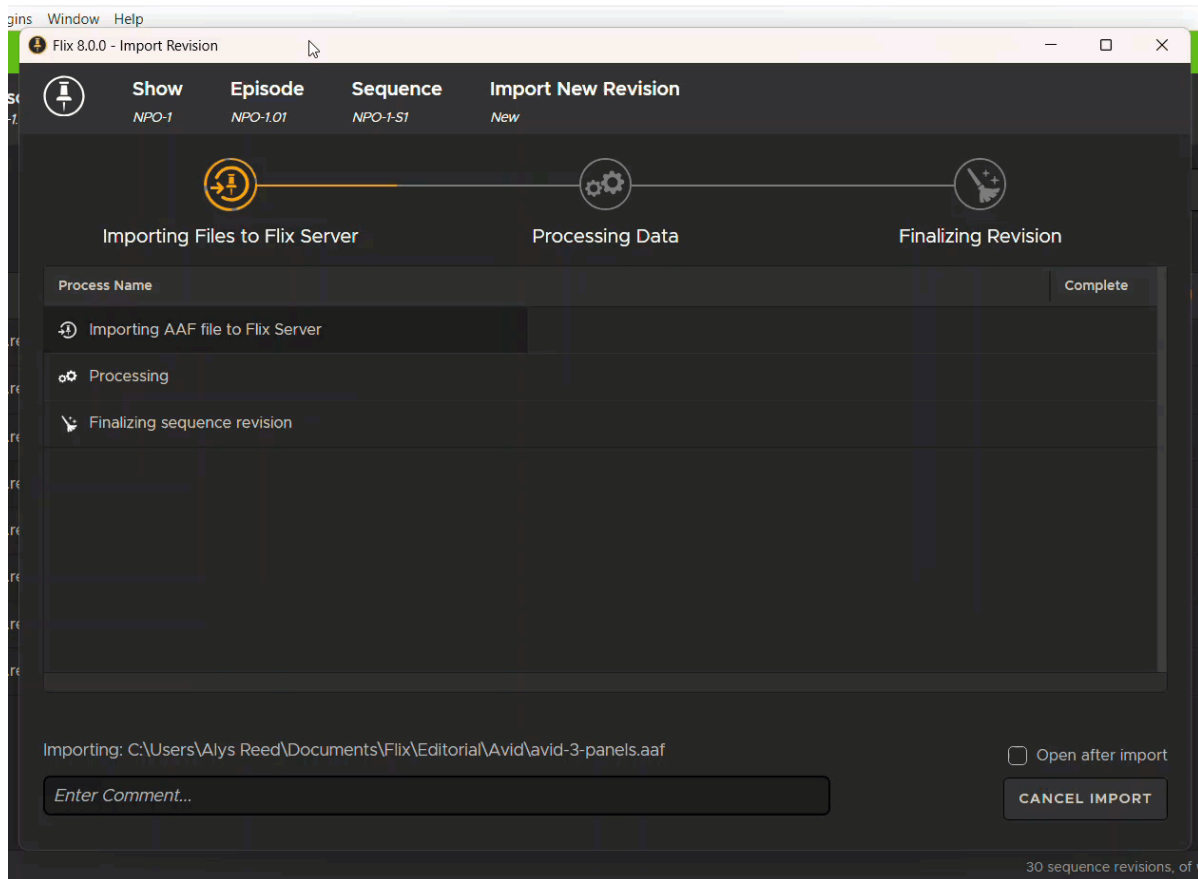
This opens the **Import** window.

3. If you're using Adobe Premiere:
 - **Browse** to your **.xml** and **.mov** files to populate the **Movie Path** and **XML Path**. You can select both files at the same time in your file browser.

If using Avid Media Composer:

- **Browse** to your **.aaf** file to enter the **AAF Path**.

4. Click **Import**.



The import revision window will then display a progress bar outlining the different stages of import.

The granular updates of the import will be displayed in the Process Step table of the window. Real time progress for each step is visible via the background bar of each row. The high-level progress bar will also be updated. When the step is successfully completed, there is a tick icon (✓) that appears in the Complete row.

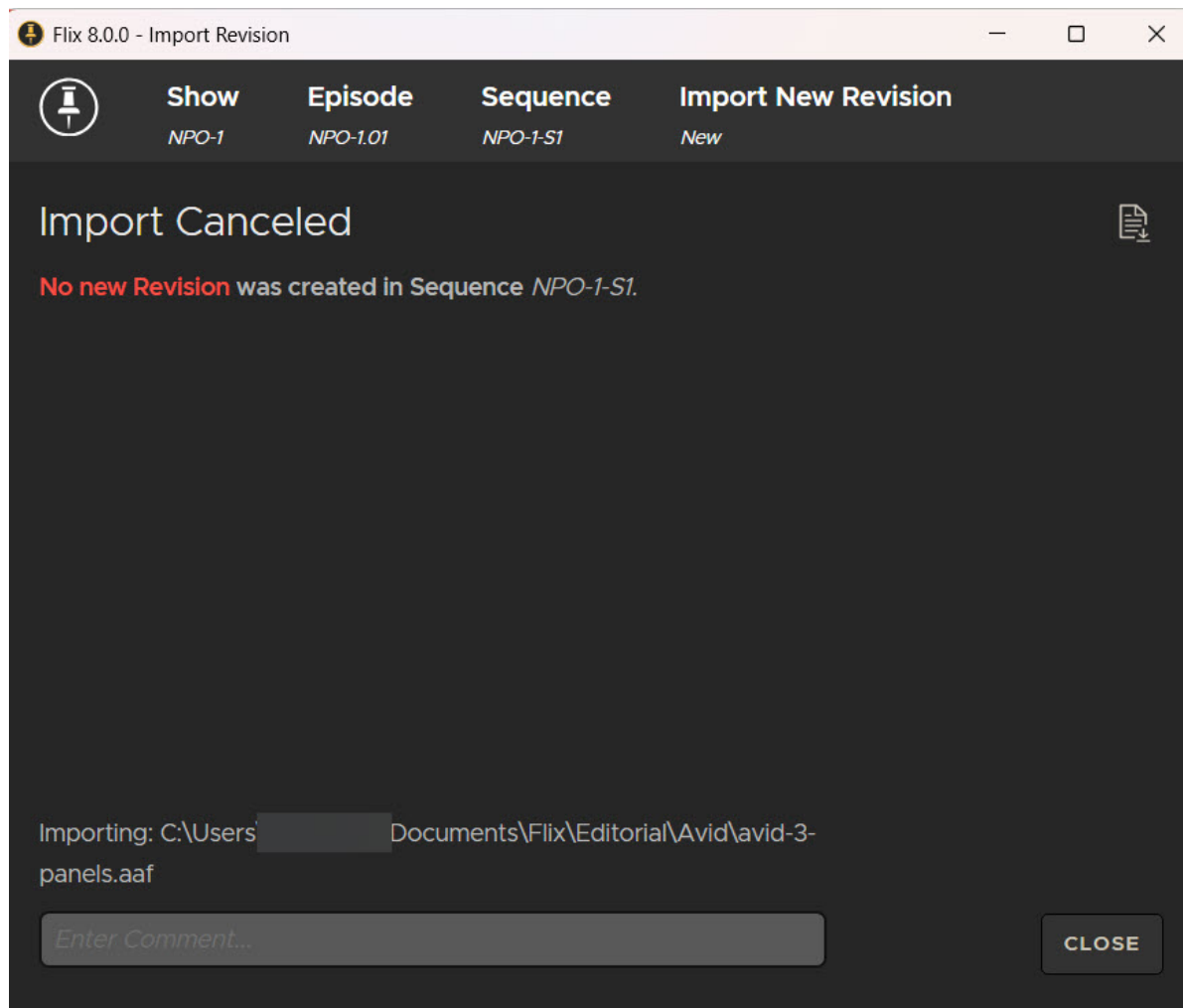
You can select whether to **Open after import** using the checkbox in the import window. If this option has been selected, the original window will automatically be taken to the newly created sequence revision, and the import window will close. When unchecked, the import window will remain open after the import has been completed.



Note: **Open after import** will only happen on a successful import.

You can add a comment to your sequence revision at any time during the import process, to give more context to the revision if desired. Enter the comment in the **Import New Revision** window. The comment will then appear on the sequence revision in the Revisions list once it has imported.

It is possible to cancel an import at any stage, and no partially created sequence will be left behind. You will still be able to Download an import log for canceled imports. Once the import has been completed, the cancel button can no longer be used.



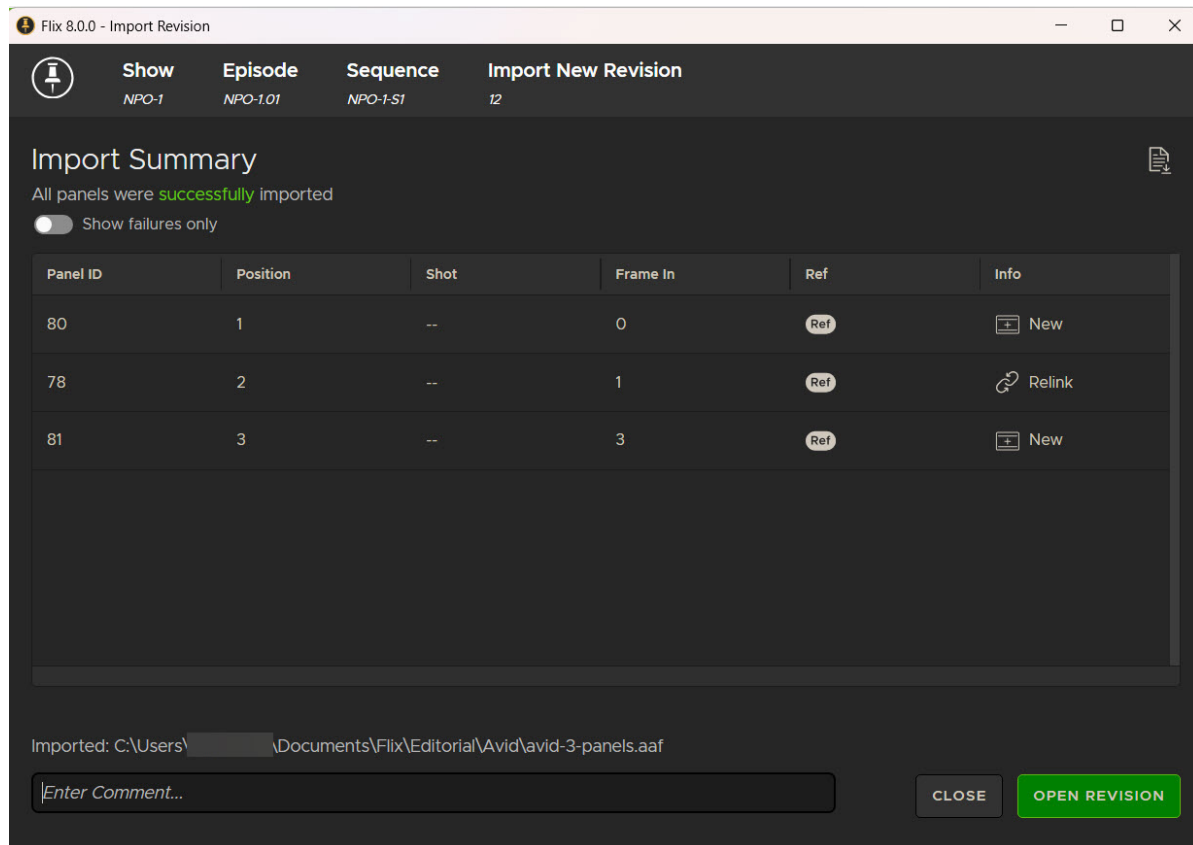
Note: Closing the import window during an import will cause the import to cancel. A warning window will pop up for confirmation of cancellation.

If an error occurs during import, an error notification banner will appear on the main Flix window. You can then inspect which section has caused the error. The imported sequence will also be marked as errored in the revision list.

If an error occurs for a panel artwork import or transcode, it will be displayed as a panel error in the final panel list.

Viewing Import Summary

Once the import has completed, the Import Revision window will display an Import Summary. This summary will contain the **Panel ID**, panel **Position**, **Shot**, **Frame in**, **Ref**, and **Info**. The panels are ordered by the panel position.

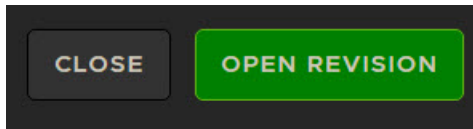


The **Info** column displays whether the panel is relinked directly to an existing panel (**Relink**), versioned-up an existing panel (**Version-up**), or imported as a new panel (**New**). There is also a tooltip that indicates whether the panel is brand new to flix, being reused from another sequence or is a duplicate of another panel in the sequence.

You can use the **Show failures only** toggle to display the failed panels.

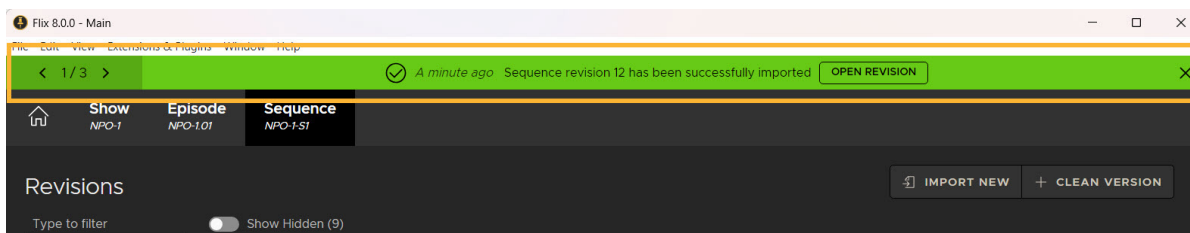
Opening New Revisions

Once the import completes, an **Open Revision** button will appear at the bottom right of the Import window.



Clicking the button will close the Import window and navigate to the new Revision in the main Flix panel browser. Clicking the **Close** button will close the Import window without opening the new Revision.

If you're not currently on the import window, an import success banner will appear on your current window. From this banner you can **Open Revision** (which will also close the import window) or **Show import window**.



Import Logs

You can get import logs for both successful and failed imports using the **Download Import Log** button (



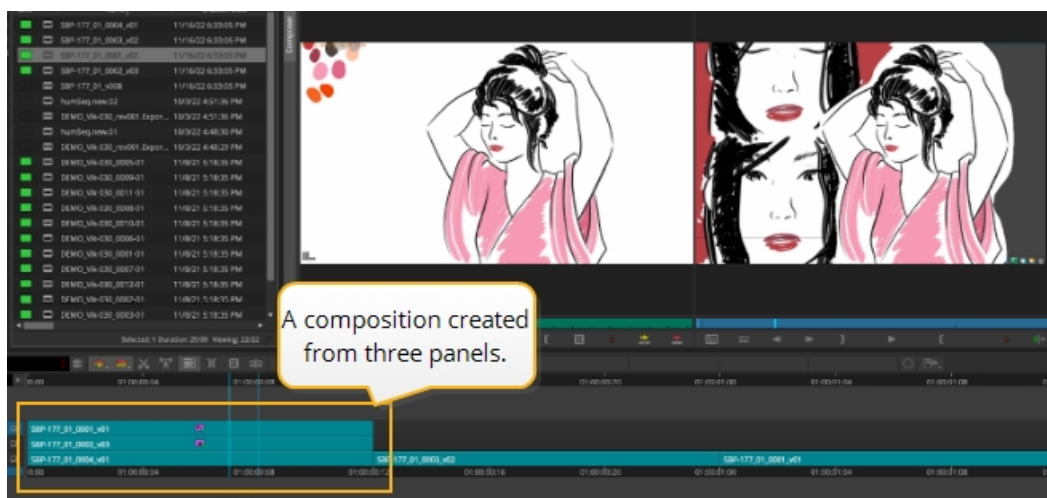
).

The import log includes:

- The Client logs
- The Server logs (for all servers)
- The TU logs
- A summary of the import results including:
 - Panel data in the table.
 - The start/end time of the import process.
 - The show, sequence and revision IDs.

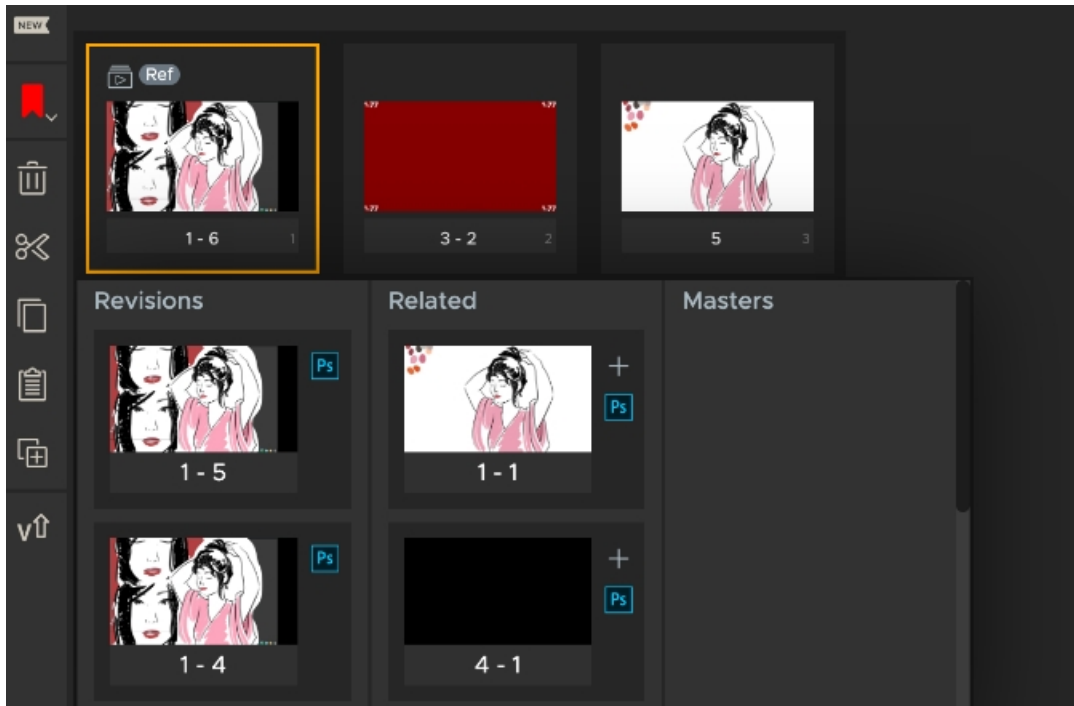
View Related Panels from Editorial in Flix

If you've created a composition from multiple panels used in the edit, the related panels are also sent to Flix as Ref panels. Reference panels provide easy access to all the images used in Editorial to allow Flix to re-create a composition in the original PSD, improving collaboration and cooperation between Story and Editorial.



Warning: Do not mix-down the composition panels in Editorial or Flix can not use its versioning system to create panel relations

Panels in Flix that include reference panels from Editorial are marked with the Ref tag. You can view the panels used in the composition in Flix by clicking on the panel ID and browsing through the **Related** section.



Note: You may find that the **Related** panels column includes one or more versions of the same versioned-up reference panel ID. This is to ensure that the link between related panels and their artwork is maintained if the reference panel is duplicated.

Access Editorial Publish Information from Your Inbox

Publishing from Editorial automatically notifies you with an easy to read email containing a breakdown of what's changed. No one can access the links without the correct Flix credentials, so your changes stay confidential and protected within your production.

The **Management Console** > **Permissions** tab controls who is notified by email about publishes using the **Shows** > **Receive Email** permission. The **Regular user** and **Show manager** roles are emailed by default, but you can add new roles to control who is notified by publishes. See [Creating Groups and Assigning Roles to Manage Permissions](#) for more information.



Show

EML

Sequence

ALD

New Revision

2

A new sequence revision has been published *from* editorial for *ALD*.

NEW	New Panels 60	Change by	Shot	ID	Index
	EML_ALD_0061_v01	KatiePM		61	1
	EML_ALD_0062_v01	KatiePM		62	2
	EML_ALD_0063_v01	KatiePM		63	3
	EML_ALD_0064_v01	KatiePM		64	4
	EML_ALD_0065_v01	KatiePM		65	5
	EML_ALD_0066_v01	KatiePM		66	6

Each publish email contains the following links and information:

- Clickable redirects to the relevant show, sequence, or revision in Flix.
- Sequence revision comments.
- Publish **to** Editorial only: A link to published files and Contact Sheet.



Note: The link redirects you to Flix Client and requires you to log in for security purposes before you can access the link.

- Publish **to** Editorial only: Panel comments added or resolved since the previous publish to Editorial.
- Information on:
 - New panels.
 - New panel revisions.
 - Deleted panels.

- Panel duration changes.
- Publish **from** Editorial only: Camera move changes.

Photoshop Extension Workflows

The following video provides an overview of how Flix and Photoshop work together.

Setting Up Flix to Work with Photoshop

To use Photoshop with Flix, you need to configure Photoshop as your sketching tool and install the Flix Photoshop plugin.



Note: Ensure Photoshop is closed while setting preferences in Flix.

Configure Flix to Use Photoshop

Open the Sketching Tools settings:

1. Navigate to **File > Preferences** on Windows or **Flix > Preferences** on macOS. Alternatively, use the **Ctrl/Cmd+,** (comma) keyboard shortcut.
2. Expand **Third Party Apps** and click **Sketching Tools**.

Set up Photoshop as the sketching tool:

1. Choose **Adobe Photoshop** as the **Sketching Tool**.
2. Click **BROWSE** to locate and select the Photoshop executable. This populates the **Executable** field.

Install the Flix Extension for Photoshop

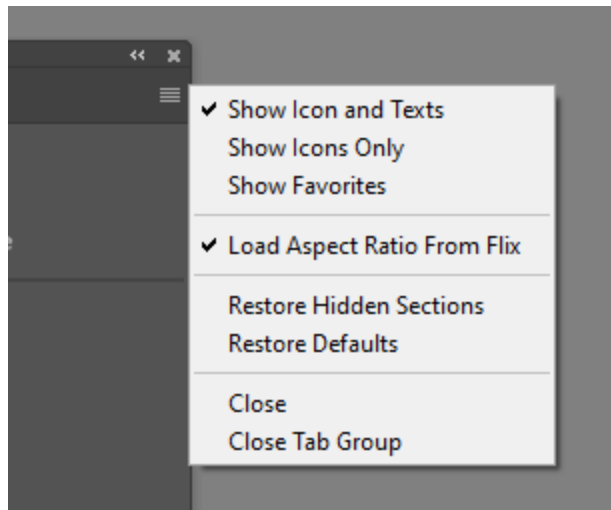
You must install the Photoshop Flix extension to use the Flix plugin in Photoshop. For instructions on how to do this see [Managing Extensions & Plugins](#).

Once the extension is installed, open it in Photoshop by selecting **Plugins > Flix > Flix**.

Customizing the Flix Plugin

The plugin can be customized to suit your workflow by hiding action categories and collecting favorites into a separate section.

Viewing options can be selected using the plugin hamburger menu.



Action Display Options

By default, the action includes an icon and a text label.

- Click on the hamburger menu and select **Show Icons Only** to remove the text label
- Click on the hamburger menu and select **Show Icon and Texts** to show both icon and text.

Hiding and Managing Favorite Actions

You can hide any category that you no longer need on the plugin panel.

- Right-click on the category name and select **Hide This Section**.

The plugin lets you choose favorite actions to show in the **Favorites** section.

- Click on the hamburger menu and select **Show Favorites** to show the **Favorites** section.
- Right-click on any action to add it to favorites. Once added, right-click to remove it.

Restoring the Default Layout

- To reverse any customization and reset to factory defaults, click on the hamburger menu and select **Restore Defaults**

Photoshop Workflow Options

The Flix plugin offers four ways of composing panels to suit your preferred workflow.

- **Images** - This is perhaps the most common workflow and involves simply sending the currently visible image as shown on the Photoshop canvas to Flix as a panel. You can also create a set of thumbnails in a single Photoshop file to design nine panels in a 3x3 grid. Each thumbnail in the grid corresponds to a single panel.
- **Layer Comps** - You can maintain alternate versions of images by taking a snapshot of layer visibility, position, and style as a 'layer comp'. You can then use the layer comps to generate Flix panels. Each layer comp corresponds to a single Flix panel.
- **Frame** - Photoshop can maintain a sequence of images as frames on a timeline. This is a neat way of previewing panels and playing through the sequence. Any of the frames can be sent to Flix as panels.
- **Layer** - Should you prefer, you can design panels using layers and layer groups in Photoshop. Each group or a non-grouped layer corresponds to a single panel in Flix.

About Source Files

Throughout this document, we refer to a 'Source File', which is the original file that was used to generate panels. In Flix, when you open panels in Photoshop, you have the option to open the source file or open a file that contains only the visible elements in the panel.



Note: Prior to Flix 7, the Source File was referred to as the Master Image.

- To open the source file in Photoshop, navigate to **Flix > Preferences > Third Party Apps > Sketching Tools** and activate **Always Open Source File**.

To open a source file, it must be stored on the Flix server.

- To send the source file from Photoshop to the Flix server when sending/replacing panels, check the **Always include source file checkbox** at the bottom of the Flix plugin.



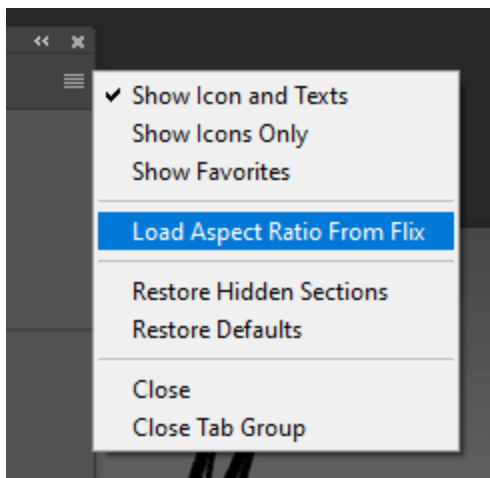
Note: Source files can inevitably take up more storage on the Flix server than files holding only visible layers. As a result, your organization may recommend best working practices when managing source files to avoid using excessive server storage.

Setting the Aspect Ratio

Make sure you use the correct aspect ratio for your project. By default, the Flix plugin loads the aspect ratio from Flix.

To set the aspect ratio manually:

1. Open the hamburger menu on the Flix plugin and uncheck **Load Aspect Ratio from Flix**.



2. Use the **Aspect Ratio** dropdown at the bottom of the plugin to select the required value.

Pushing Panels to Flix

All the workflows have both **Send** and **Replace** actions.

Send Actions

Send actions simply insert new panels into the timeline after the current panel selection in Flix.

Replace Actions

Replace actions are used to create a new versions of panels in Flix, based on the current panel selection.

When replacing multiple panels, panels are generated in the order of the items used to define the panels in Photoshop. For example, in the order of the **Layer Comp** list.

Multiple panels will overwrite the same number of panels in Flix. For example, if you have 5 layer comps, you can push them to Flix in order using a single action (**Replace Each Layer Comp**). This replaces 5 panels in parallel.

Sending New Panels and Thumbnails To Flix



Note: Transferring panels from Photoshop to Flix locks the target Flix revision and you cannot open a new panel in Photoshop or use any of the replace actions during the transfer process.

Create a Panel and Send it to Flix

Flix sends the panel to the edit (sequence revision). The panel is inserted after the currently selected panel

1. Click **New Template** to open a blank canvas with the correct aspect ratio.
2. Add your artwork.
3. When you're ready to send, click **Send Current Image** or a different send option if you're using another workflow (see [Photoshop Workflow Options](#)).

Creating Thumbnails and Sending them to Flix

Flix sends the thumbnails to the edit. They are inserted as nine new individual panels after the currently selected panel, if any.

1. Click **New Thumbnails**
2. Draw on the thumbnails.
3. Click **Send Thumbnails**.

Annotating a Panel Using Photoshop

This video demonstrates how to make annotations on Flix panels using Photoshop.

1. In Flix, select a panel in your edit and click the Open in Sketching App  button.

The panel opens in Photoshop.

2. In Photoshop, create a new layer and draw an annotation on that layer.
3. In the Flix toolbar, click **Send Selected Layers as Annotation**.

In Flix, click the Toggle annotations  button under the player to show the new annotation.

Working with Photoshop's Layer Comps

Layer Comps let you maintain different versions of your work in a single **.psd** according to visibility, style, and position of layers.

For more information on using layer comps, see [the official documentation](#).

Sending Layer Comps

The Flix plugin lets you send either **Selected** layer comps or every (**Each**) layer comp. One layer comp generates a single panel, and so the number of panels generated is the same as the number of layer comps you have in the file.

- Select **Send Each Layer Comp** or **Send Selected Layer Comps** to send the layer comps after the currently selected panel in Flix.



Note: Check **Always include source file** to send the source file to Flix. See [Photoshop Extension Workflows](#).

Making Revisions Using Layer Comps From Source Files

When revising panels, they may be generated using layer comps for which you have stored the source **.psd**. In this case, you can reopen the source file in Photoshop to apply your revisions.



Note: If a layer does not have a stored source file, see [Making Revisions Using Layer Comps Without Source Files](#).

1. Ensure the **File > Preferences > Third Party Apps > Sketching Tools > Always Open Source File** is enabled.
2. Select the panel in Flix you want to revise.
3. In the main toolbar, click the Open in Sketching app  button.
4. This opens the master image in Photoshop with all the information, including hidden layers and empty groups.
5. In Flix, this opens the Master file that the panel is linked to in Photoshop.
6. After you are done working on your master image, save the **.psd** file and select **Replace Each Layer Comp** from the Photoshop actions.

In Flix, this updates panels from in the order of the first layer comp to the last.

This video demonstrates how source images are created and where Flix stores them.

Making Revisions Using Layer Comps Without Source Files

Panels may not have a stored source file, but you may want to use layer comps as a way of storing and editing a collection of panels. In this case, you can send any number of panels to Photoshop and have them collected in a single file containing one layer comp for each panel.

1. To send panels to a Photoshop file using layer comps:
2. In Flix, navigate to **File > Preferences** and open **Third Party Apps > Sketching Tools**.
3. Set **Panel Open Behaviour** to **Open in Layer Comps**.
4. In your edit (sequence revision), select the required panels you want to open as layer comps.

5. In the main toolbar, click the Open in Sketching app  button to send your panels to Photoshop.
6. This opens one **.psd** file containing a layer comp for each selected panel, each layer comp containing the layers for the selected panels.
7. Edit and save the **.psd** file to automatically update the Flix panels.

Working with Photoshop's Frame Timeline

Photoshop's timeline feature is a useful tool to construct an animation or play through a series of stills.

You can capture the state of your layers and add the resulting image as a frame to the timeline. By manipulating layer visibility and properties, you can build a sequence of frames and play through them.

The Flix plugin lets you send frames from Photoshop and use them to generate panels in Flix.



Note: For more information on the timeline, see [the official documentation](#).

Send Each Frame To Flix

To send each frame as a separate panel, click **Send Each Frame**.



Note: Check **Always include source file** to send the source file to Flix. See [Photoshop Extension Workflows](#).

Making Revisions Using Frames and Source Files

You can revise Flix panels in Photoshop using existing source files that have already been set up using the timeline frame workflow.

To revise a panel using the timeline-based workflow:

1. In Flix, ensure **File > Preferences > Third Party Apps > Sketching Tools > Always Open Source File** is enabled.
2. Select the panel you want to work on.

3. In the main toolbar, click the Open in Sketching app  button.

This opens the source file in Photoshop with all the information, including hidden layers and empty groups.

4. Edit the frames in Photoshop.
5. To propagate the changes, save the **.psd** file and select **Replace Each Frame** from the Photoshop actions.

This updates panels in Flix from the first selected.

Making Revisions Using Frames Without Source Files

If you haven't yet used the timeline frame workflow or do not have the original source file, you can create timeline frames in Photoshop from existing Flix panels.



Note: When you are working in a Frame Timeline without any original source file, in the Frame Timeline, click the hamburger menu and disable **New Layers Visible in All Frames**. This displays all the frames in the Frame Timeline as separate frames, otherwise all the frames use the same images.

1. Navigate to **File > Preferences > Third Party Apps > Sketching Tools > Panel Open Behavior** and select **Open in Timeline**.
2. In your edit (sequence revision), select the required panels you want to open in a timeline.
3. In the main toolbar, click the Open in Sketching app  button to send your panels to Photoshop.

This opens a **.psd** file containing one frame for each selected panel, each frame containing the layers used for the selected panels.

4. To propagate the changes to Flix, save the **.psd** file.

Working with Photoshop's Layers/Groups

The Flix plugin lets you create panels from layers in Photoshop.

When working with your file in Photoshop, you have two ways of delivering panels to Flix:

- **Send** layers to create new panels in Flix
- **Replace** panels from the selected panel position in Flix using the Photoshop layers



Note: Make sure the layers or groups you want to send are visible, otherwise they will be ignored.

The Flix plugin has the following send actions and a **Replace** version for each action:

- **Send Each Layer** - Sends to Flix each visible layer in an image as a separate panel.
- **Send Each Layer + BG** - Sends to Flix each visible layer in an image as a separate panel but keeps the background the same for each one.
- **Send Each Layer + FG/BG** - Sends to Flix each visible layer in an image as a separate panel but keeps the background and foreground the same for each one.
- **Send Selected Layers** - Sends to Flix only the selected layers of the image as a new panel.



Note: Check **Always include source file** to send the source file to Flix. See [Photoshop Extension Workflows](#).

Making Revisions from Layers and Groups With Source Files

If you have the source file associated with your panel, you can send this to Photoshop to make revisions.

To open the panel's source file in Photoshop:

1. Ensure the **File > Preferences > Third Party Apps > Sketching Tools > Always Open Source File** is enabled.
2. Select the panel you want to work on.
3. In the main toolbar, click the Open in Sketching app  button.
4. This opens the source file in Photoshop with all information including hidden layers and empty groups.
5. Edit the file and save the **.psd**
6. Select a **Send** or **Replace** layer action (see above) from the Flix plugin actions.

This updates panels in Flix from the first selected starting from the first layer or group.

Making Revisions to Layers and Groups Without Source Files

If you don't have the source file for your panel(s), you can create a **.psd** with layers corresponding to the selected panels.

To create a **.psd** with layers using selected panels:

1. Navigate to **File > Preferences > Third Party Apps > Sketching Tools > Panel Open Behavior** and select **Open as separate PSD**.
2. In your edit (sequence revision), select the required panels you want to open as layers or groups.
3. In the main toolbar, click the Open in Sketching app  button to send your panels to Photoshop.

This opens a **.psd** file containing your layers or group for each selected panel.

4. To push your changes to Flix, edit the layers and save the **.psd** file

This automatically replaces the opened panels with the updated layers or groups.

Photoshop Actions

Flix offers a Photoshop plugin so you can edit your panels in Photoshop and send them back to Flix. The Flix plugin contains a number of actions for creating and sending panels.

For information on installing and running the Photoshop plugin, see [Setting Up Flix to Work with Photoshop](#).



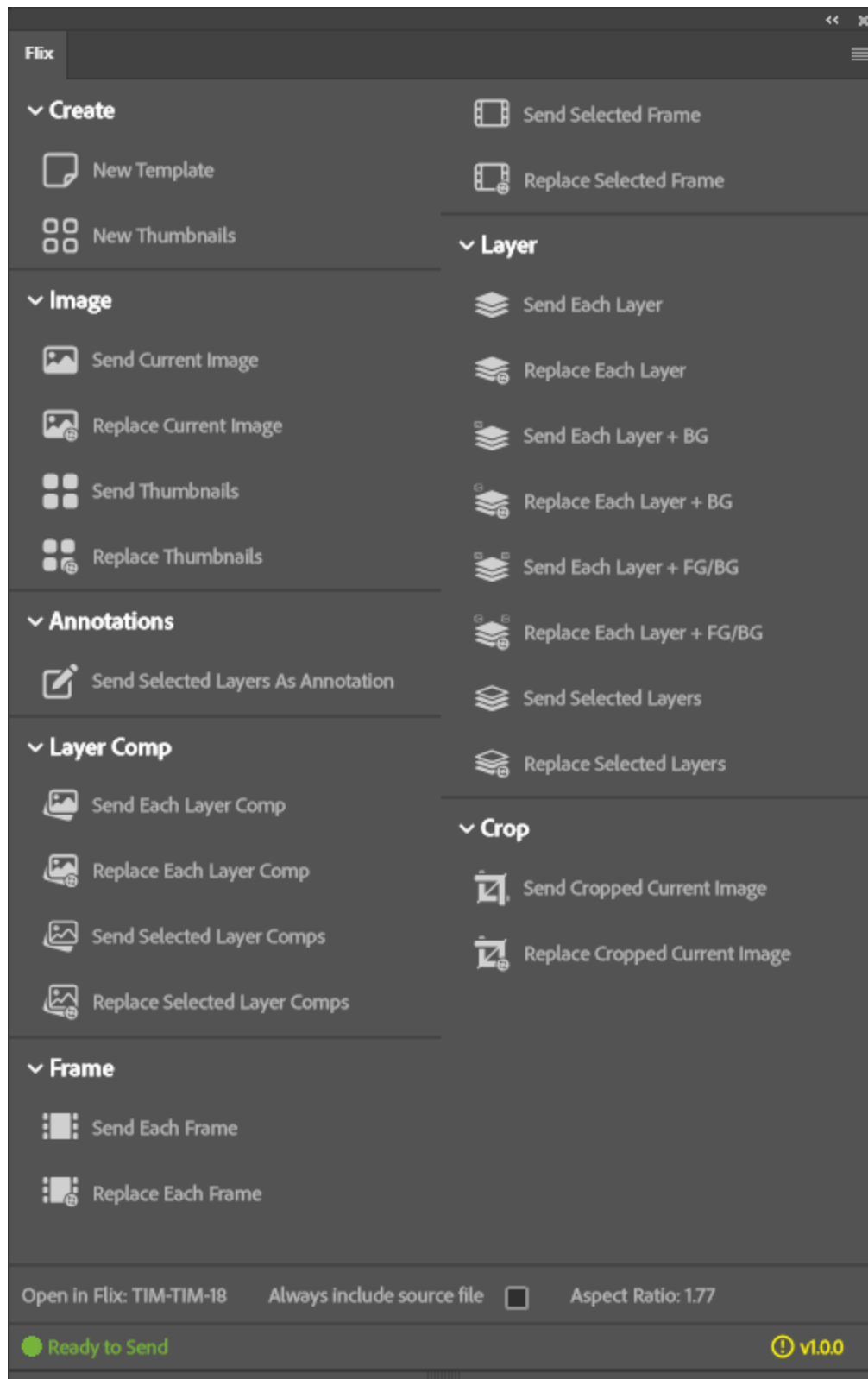
Note: Flix only supports certain versions of Adobe Photoshop, depending on the version of Flix you have installed. See [Third-Party Application Support](#) for more information.

Photoshop Plugin Overview

To open the Flix Photoshop plugin:

- In Photoshop, navigate to **Plugins > Flix > Flix**.

The plugin opens in a new panel, and you'll see the actions categorized by type.



The Flix Photoshop plugin.



Note: Both Photoshop and Flix need to be open for these actions to work.

Send vs Replace Actions

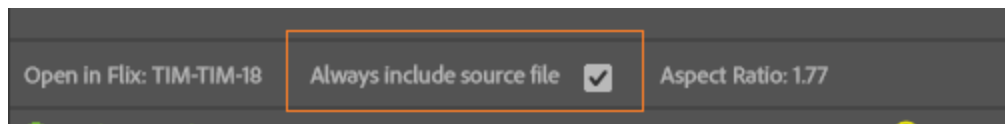
For many of the actions, there are two options, send and replace. These give you more control over how panels are added to your Flix sequence revision.

- Send inserts the panel(s) after the currently selected panel in Flix.
- Replace overwrites the current selection. When sending multiple panels, a replace action will also overwrite the subsequent panels.

Sending the Source File

You may want to store the complete source file, including hidden layers, on the Flix server for future edits.

- Check the **Always include source file** checkbox to maintain the complete source file on the server.



Note: Prior to Flix 7, the source file was referred to as the master file.

Create Actions

Create actions are used to start new panel designs or thumbnail sets.

Thumbnails are a quick way of creating panels and are useful for rapid development. A thumbnail sheet is essentially a collection of nine panels that can be sent to Flix in one batch.

Create Action	Description
New Template	Creates a new document with the aspect ratio specified by Aspect ratio at the bottom of the panel. For information on setting the aspect ratio, see Setting the Aspect Ratio .

Create Action	Description
New Thumbnails	This creates a new document that contains nine thumbnails, each with the aspect ratio specified by Aspect ratio .

Image Actions

Image actions send images and thumbnail sets to Flix.

For thumbnails, Flix converts the thumbnail sheet into panels, and you have two options for panel insertion:

- **Send** insert the thumbnails after the current selection.
- **Replace** overwrites the current selection and the eight subsequent panels.

Image Action	Description
Send Current Image	Sends the current image to Flix.
Replace Current Image	Replaces the currently selected Flix panel with the image.
Send Thumbnails	Sends thumbnails created by the New Thumbnails action to Flix as individual panels. The panels are inserted after the currently selected panel. No panels are overwritten.
Replace Thumbnails	Replaces the panels in Flix from the currently selected position with the thumbnails. All nine panels, including the currently selected one, are overwritten.

Annotation Actions

Annotation actions send a layer as a panel annotation to Flix.

Annotation Action	Description
Send Selected Layer as Annotation	Sends the currently selected layer to Flix as an annotation.

Layer Comp Actions

Layer Comp actions send layer comps as panels to Flix.

Layer Comp Action	Description
Send Each Layer Comp	Imports or re-imports each layer comp as a separate panel with all the visible layers from the individual layer comp.
Replace Each Layer Comp	Replaces Flix panels with each layer comp, starting from the selected Flix panel.
Selected Layer Comps	Re-imports all selected layer comps.
Replace Selected Layer Comps	Replaces Flix panels with each selected layer comp, starting from the selected panel in Flix.

Frame Actions

Frame actions are used when panels are defined using frames on the Photoshop timeline. The timeline gives you a neat way of organizing panels and playing through the sequence. Each frame represents an individual panel.



Note: These actions are only available if you are using the animation timeline workflow.

Frame Action	Description
Send Each Frame	Re-imports each frame in the Photoshop timeline as a separate panel.
Replace Each Frame	Replaces Flix panels with each frame in the timeline starting from the selected Flix panel. This is only available if you are using the frame timeline workflow.

Frame Action	Description
Send Selected Frame	Send the selected frame to Flix and insert it into the sequence after the current selection.
Replace Selected Frame	Send the selected frame to Flix and replace the selected panel.

Layer Actions

Layer actions are used when you're using layers to construct panels.

Layer Action	Description
Send Each Layer	Sends each visible layer in an image as a new panel.
Replace Each Layer	Sends each visible layer to replace the selected panel.
Send Each Layer + BG	Send each visible layer in an image as a new panel and keep the background the same for each one.
Replace Each Layer + BG	Send each visible layer to replace the selected panel and keep the background the same for each one.
Send Each Layer + FG/BG	Send each visible layer in an image as a new panel and keep the background and foreground the same for each one.
Replace Each Layer + FG/BG	Send each visible layer to replace the selected panel and keep the background and foreground the same for each one.
Selected Layers	Send only the selected layers of the image as a new panel.
Replace Selected Layers	Replace the selected Flix panel with only the selected layers of the image.

Crop Actions

Use crop actions to send a selection

Crop Action	Description
Cropped Current Image	Exports the image selection to a new Flix panel.
Replace Cropped Current Image	Replaces the currently selected panel in Flix with the image selection.

Maya Extension Workflows

Flix 7 extended the support for more third-party tools to aid panel design, 3D visualization, and story development by adding a Maya extension.

For action sequences, certain shots necessitate collaboration between storyboard and layout to capture complex camera movements and environmental interactions.

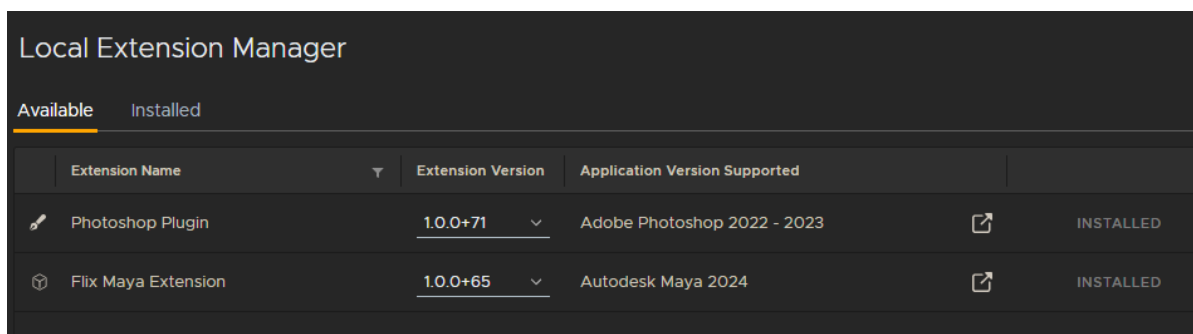
The Maya extension improves collaboration by helping you capture detailed environments and highly dynamic camera moves in the storyboarding phase, as layout artists can send snapshots or playblasts of their 3D scene to Flix as panels.

Story handoff to 3D artists is streamlined as they can load Flix panels directly into their Maya scene.

Setting up Flix to Work With Maya

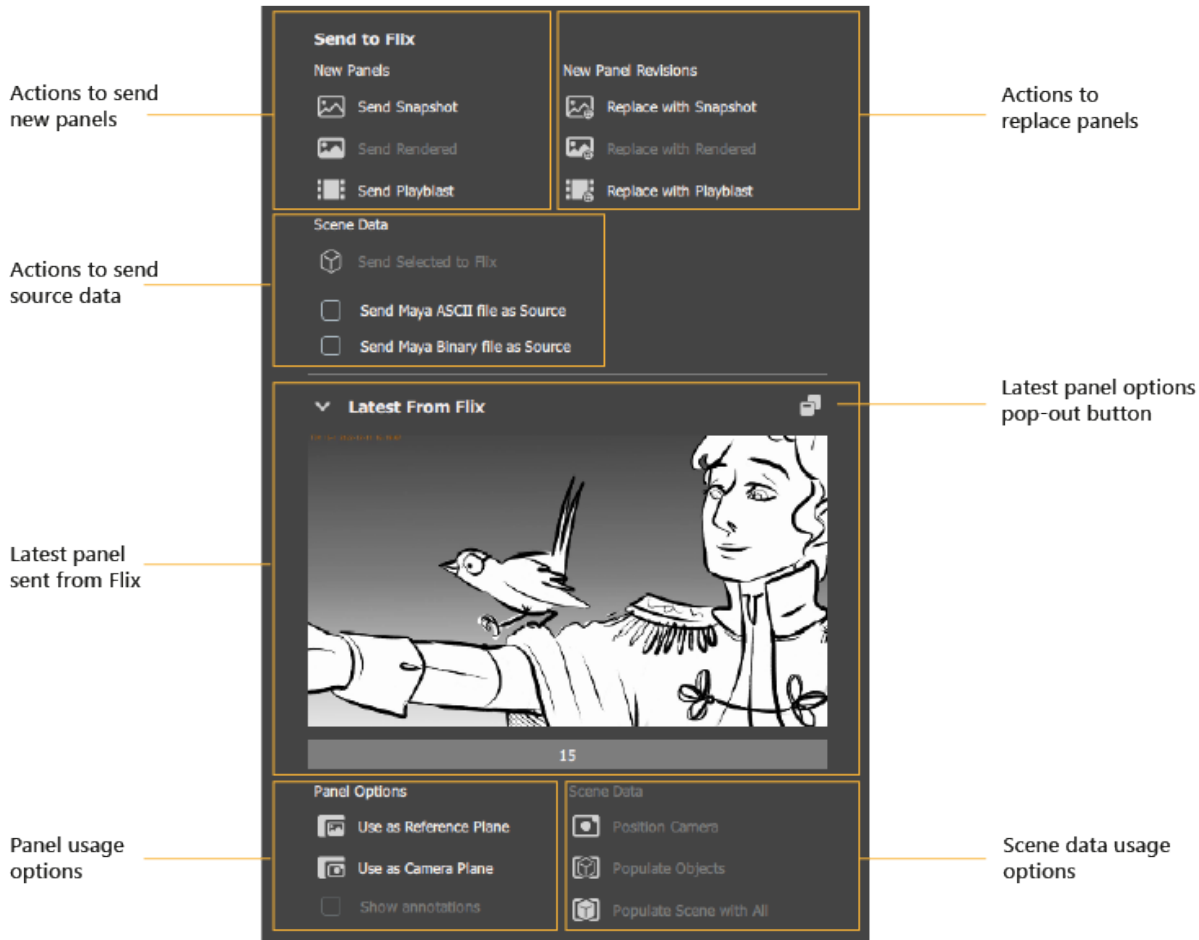
Install the Maya Extension

- Use the Local Extension Manager to install or update the Maya extension. For full instructions, see [Managing Extensions & Plugins](#).



- Once installed, open Maya and navigate to **Flix > Flix Plugin** to open the Maya Flix Plugin.

Maya Flix Plugin Overview



The Maya Flix plugin.

The plugin has two main sections:

- Send to Flix** - The upper part of the plugin contains the actions to send or replace panels in Flix using snapshots or playblasts.
- Latest From Flix** - The lower part of the plugin contains the image and controls for the most recent panel sent from Flix.

Sending New Panels and Panel Revisions from Maya to Flix

There are currently two types of Maya visuals that can be sent to Flix:

- **Snapshot** - Send a snapshot of the 3D scene in the Maya viewport. A snapshot provides an unrendered static image for a panel.

If you have an animation set up in Maya, you can step through the keyframes and take snapshots to generate a sequence of panels.

- **Playblast** - Send an animated playblast as an animated panel to Flix. A playblast is a quick preview animation you can create to get a sense of the animation without having to render it in full detail. Maya has a dedicated tool for creating playblasts.



Note: Sending a **Rendered** scene is currently unavailable and will be included in a future release.

Sending New Panels From Maya

- Construct a scene (with optional animation) or a playblast, and click **Send Snapshot** or **Send Playblast**, respectively.
- If your scene is animated, you can step through the animation and click **Send Snapshot** at each keyframe to generate a sequence of panels.
- If you are using a playblast, the resulting panel in Flix is an animated panel.

The panel is inserted in Flix after the current selection position.

Revising Existing Panels Using Maya

The workflow for panel revision is similar to sending new panels, but the selected panel in Flix is replaced by the addition of a new version.

- Click **Replace with Snapshot** to replace the existing Flix panel selection with a static snapshot of Maya's 3D viewer.
- Click **Replace with Playblast** to replace the existing Flix panel selection with an animated panel that copies the playblast.

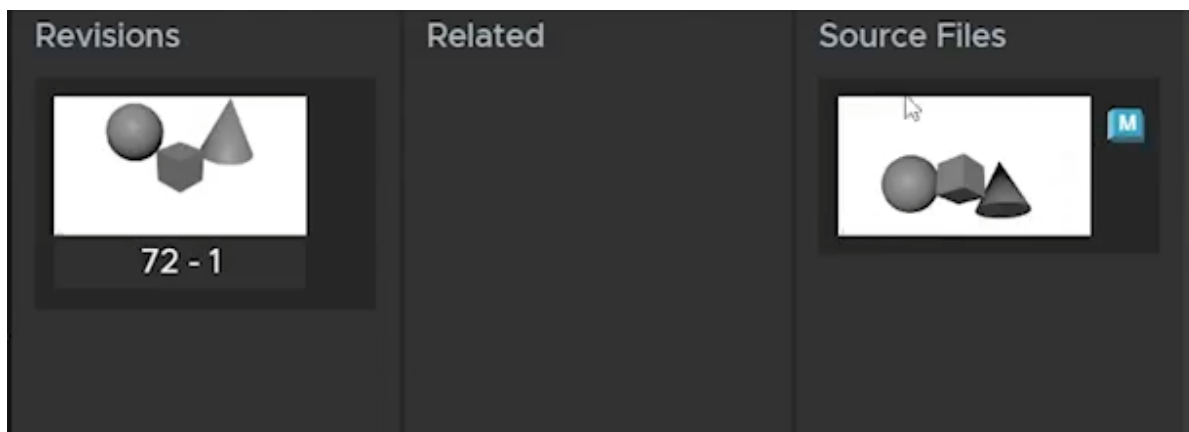
Sending Scene Data to Flix

You can include the Maya source data when sending new panels or revising panels.. By including the data, you make it available to other artists who may need to work on the same scene.

You can choose to send the source as an **.ma** (Maya ASCII) file, or an **.mb** (Maya binary) file.

- Check **Send Maya ASCII file as Source** or **Send Maya Binary file as Source** to send the source in the preferred format.

In Flix, you can see that the source has been sent if it can be seen in the **Source Files** column of the panel revisions pop-up in Flix. The file will also be available in the Source File Library. See [Browsing Source Files in the Source File Library](#).



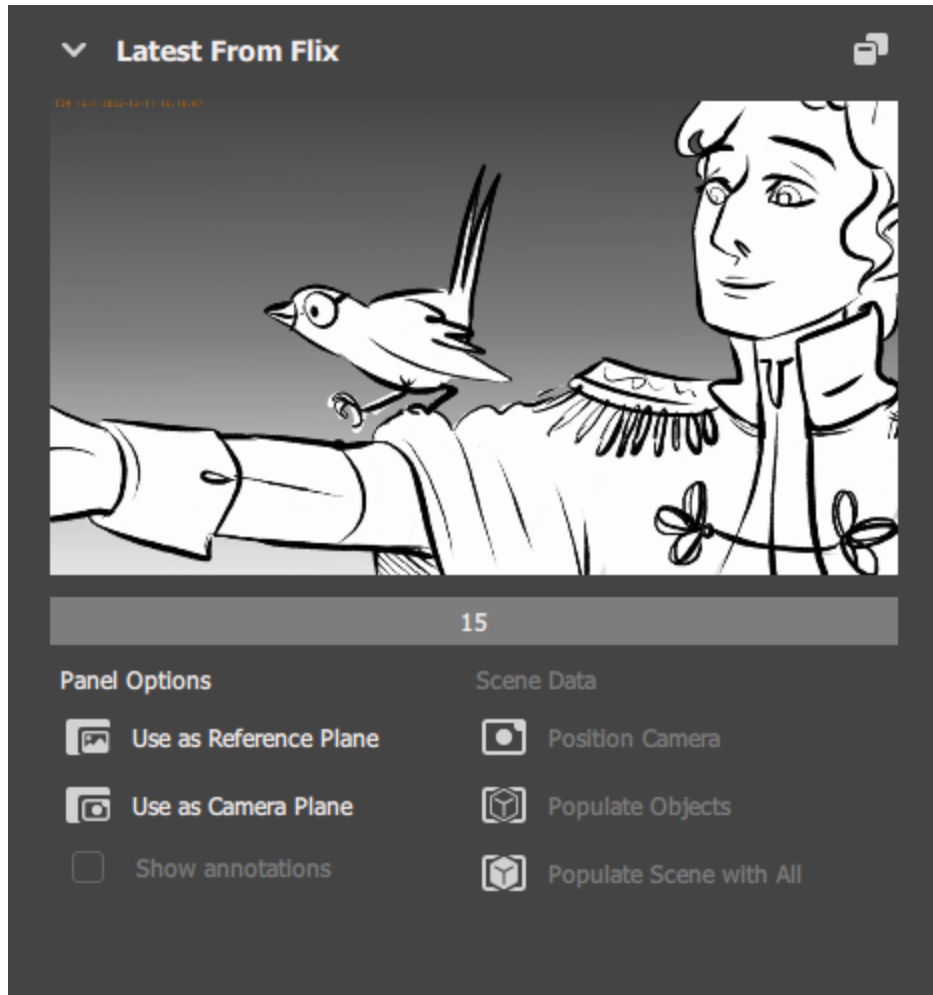
In Flix, click the panel revision number under the panel thumbnail to see corresponding source files.

Sending Panels to Maya

You can send a panel to Maya for use in the 3D scene to aid in constructing or modifying panels that can then be sent back to Flix.

- In Flix, select any panel and click the Send to 3D  button in the toolbar to send the panel to Maya.

Once the panel is received, Maya populates the **Latest Fom Flix** section of the plugin with the panel image.

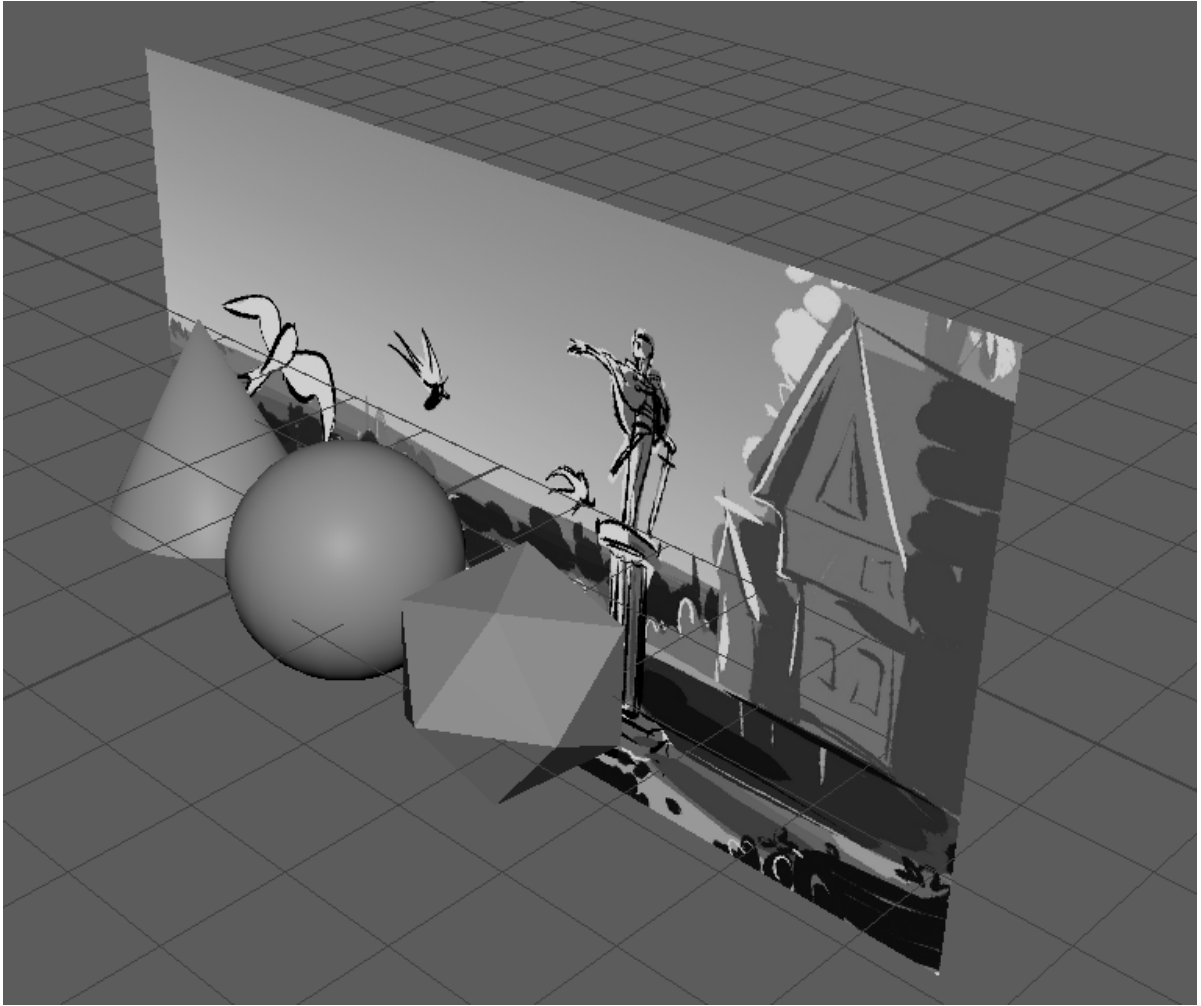


Maya shows the latest panel that is sent from Flix. The panel image can then be used in the 3D scene.

Using Panels in Maya

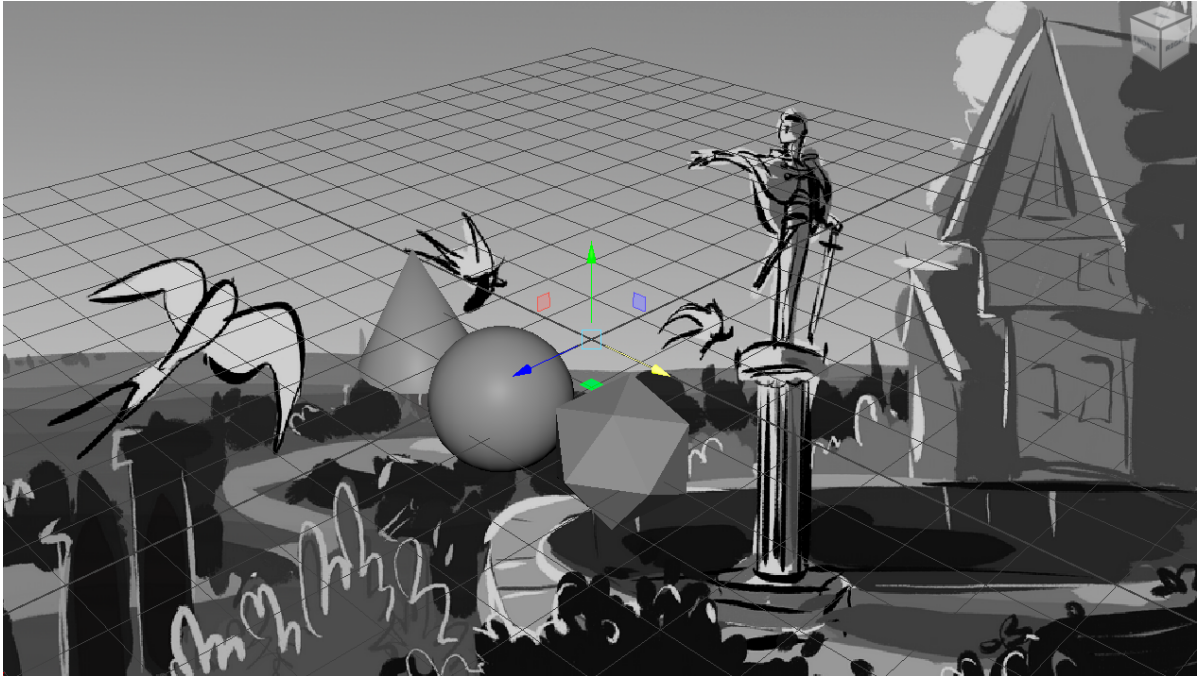
Maya can receive panels from Flix to be used in two ways:

- Click **Use as Reference Plane** to add the panel as a reference plane to the scene on the xy plane. The reference plane can be positioned anywhere in the 3D space and used as reference for building the 3D scene.



A panel inserted as a reference plane. The 3D primitives show how the panel is inserted in the scene.

- Click **Use as Camera Plane** to fix the Flix panel image to the viewing plane of selected viewport's camera. You can then add and manipulate 3D objects over the backdrop.



A panel inserted as a camera plane.



Note: It is possible to use both methods.

Storyboard Pro & Flix

Setting Up Flix for Storyboard Pro

To set Storyboard Pro as your Sketching Tool:

1. Navigate to **File > Preferences > Third Party Apps > Sketching Tools**.
2. Select **Storyboard Pro** from the dropdown menu.
3. Set the **Executable** preference by browsing to the **StoryboardPro** (.exe on Windows, .app on Mac) executable file on your computer. The file path should look something like this:

`C:\Program Files (x86)\Toon Boom Animation\Toon Boom Storyboard Pro
<version>\win64\bin\StoryboardPro.exe`

/Applications/Toon Boom Storyboard Pro <version>/Storyboard Pro <version>.app

4. Choose the default **Store Source Files** behavior during imports to Flix from Storyboard Pro:

Enabled (default) - save all **.sbpz** source files on the Flix Server during import. Saving the Storyboard Pro **.sbpz** project to the server allows all artists to access the main source file and assets from one central location so that you always know you're working on the latest version.

Disabled - do not save **.sbpz** source files on the Flix Server during import. If you don't save the source files to the Flix Server, you may find some artists are working on older versions of the Storyboard Pro source files that they have saved locally.

5. Under the **Export** section, choose the behavior for exporting Flix sequences to Storyboard Pro.

Export Path - Set the path for your exported Flix sequence.

Installing the Flix Extension for Storyboard Pro

You must install the Storyboard Pro Flix extension to use the Flix plugin in Storyboard Pro. For instructions on how to do this see [Managing Extensions & Plugins](#).

Once the extension is installed, open it in Storyboard Pro by selecting **Windows > Flix**. This will open the Flix plugin in the Storyboard Pro UI.

To learn more about how to use the Flix extension, go to the [Storyboard Pro Extension Workflows](#) section of this page.

Importing Storyboard Pro Projects into Flix

You can create a new sequence revision by importing your Storyboard Pro project directly into Flix.



Article: Have a look at Knowledge Base article [Q100650](#) for more detailed information on Storyboard Pro camera animation.

To import your Storyboard Pro project into Flix:



Warning: Windows only: From Flix 7.1, ingests from Storyboard Pro will only work if the OpenH264 codec is installed on your machine.

1. Save your Storyboard Pro project.
2. In **User Preferences**, enable **Store Source File** to save **.sbpz** source files on the Flix Server. Saving the Storyboard Pro **.sbpz** project to the server allows all artists to access the main source file and assets from one central location so that you always know you're working on the latest version.

Sketching Tools

Sketching Tool: Storyboard Pro

Executable: C:\Program Files (x86)\Toon Boom Animation\Toon B... **BROWSE**

Store Source File ☒

Export Path: C:\Users\...AppData\Roaming\Flix\SBP **BROWSE**

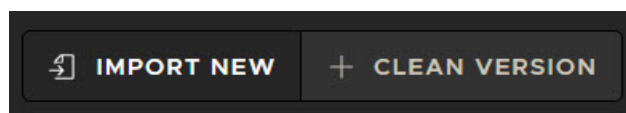


Tip: The **Store Source Files** option is enabled by default in the **Preferences** under **Third-Party Apps** > **Sketching Tools (Storyboard Pro selected)** > **Store Source File**.

Once an **.sbpz** file is stored on a Flix Server, artists can download the source files quickly and easily by clicking the download button at the **Sequence** level.

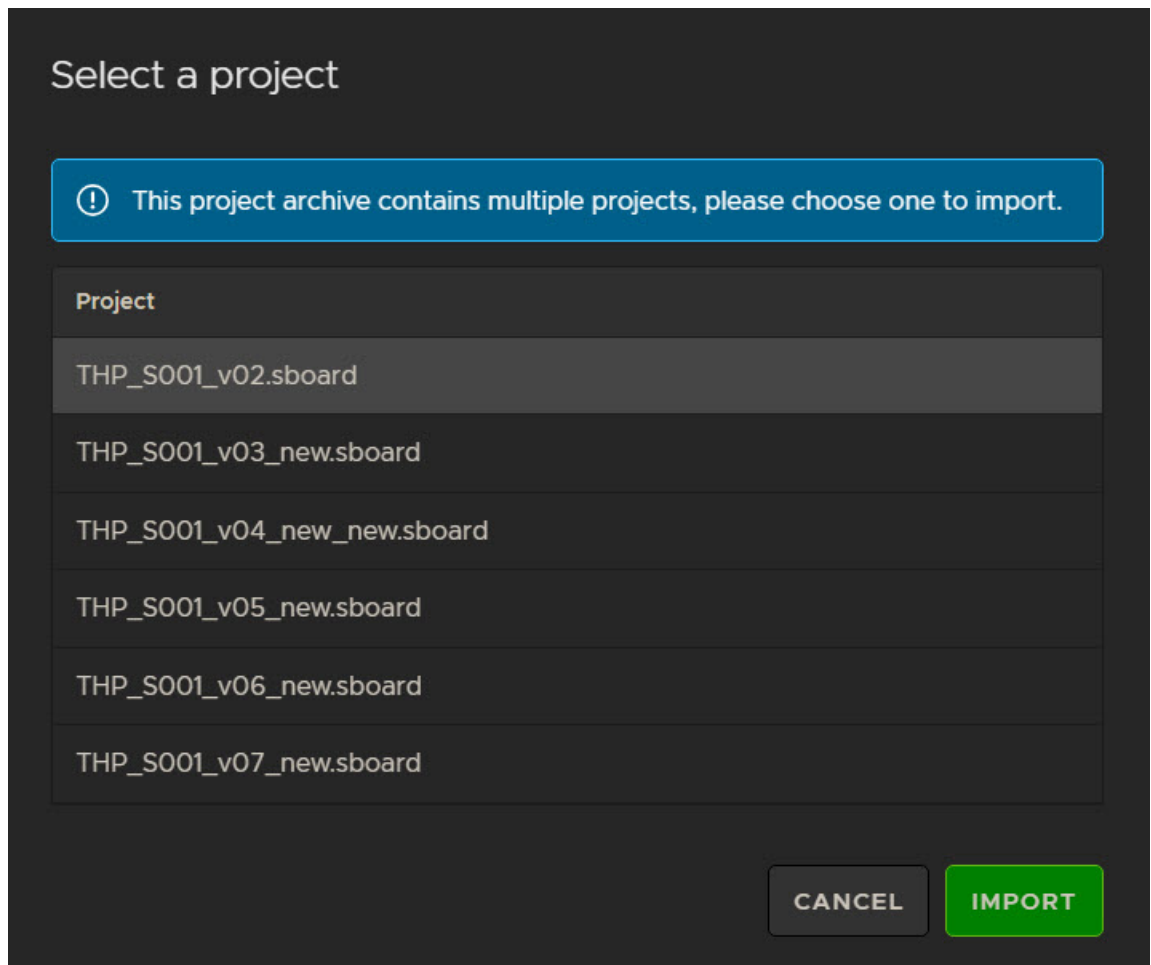
	#	User	Date	Comment	Source Files	To Editorial	From Editorial
⋮	14		Aug 4, 2025, 12:31:06 PM	--			
⋮	13		Jul 31, 2025, 11:58:07 AM	--			✓

3. In Flix, navigate to the Revisions page, either by clicking on your sequence or creating a new one.



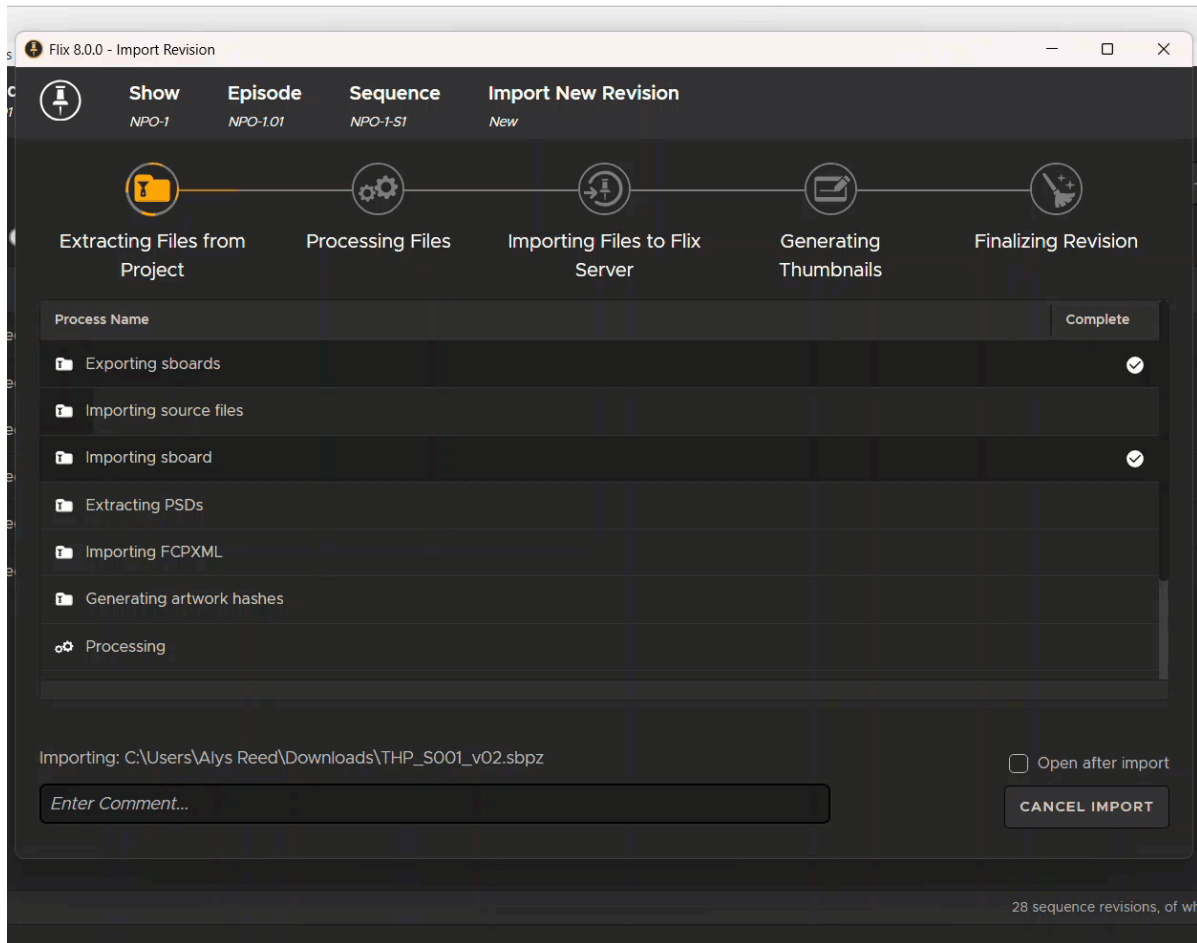
4. Click on the **Import New** button and select the Storyboard Pro project you want to import.
5. The Import page opens in a new window.
6. Click **Browse** and navigate to the Storyboard Pro project (**.sboard** or **.sbpz**) file you want to import. Select the file and click **Open**.

Importing **.sbpz** files provides an optimized experience using less storage and reducing backup issues or loss of data. On import, you can choose which **.sboard** to load from the available boards packaged within the **.sbpz** file:



The path to the project file appears in the **Project Path** when you've chosen the required file.

7. Click **Import**.



The import revision window will then display a progress bar outlining the different stages of import.

The granular updates of the import will be displayed in the Process Step table of the window. Real time progress for each step is visible via the background bar of each row. The high-level progress bar will also be updated. When the step is successfully completed, there is a tick icon (✓) that appears in the **Complete** row.

You can add a comment to your sequence revision at any time throughout the import process, to give more context to the revision if desired.

You can select whether to **Open after import** using the checkbox in the import window. If this option has been selected, the original window will automatically be taken to the newly created sequence revision, and the import window will close. When unchecked, the import window will remain open after the import has been completed.

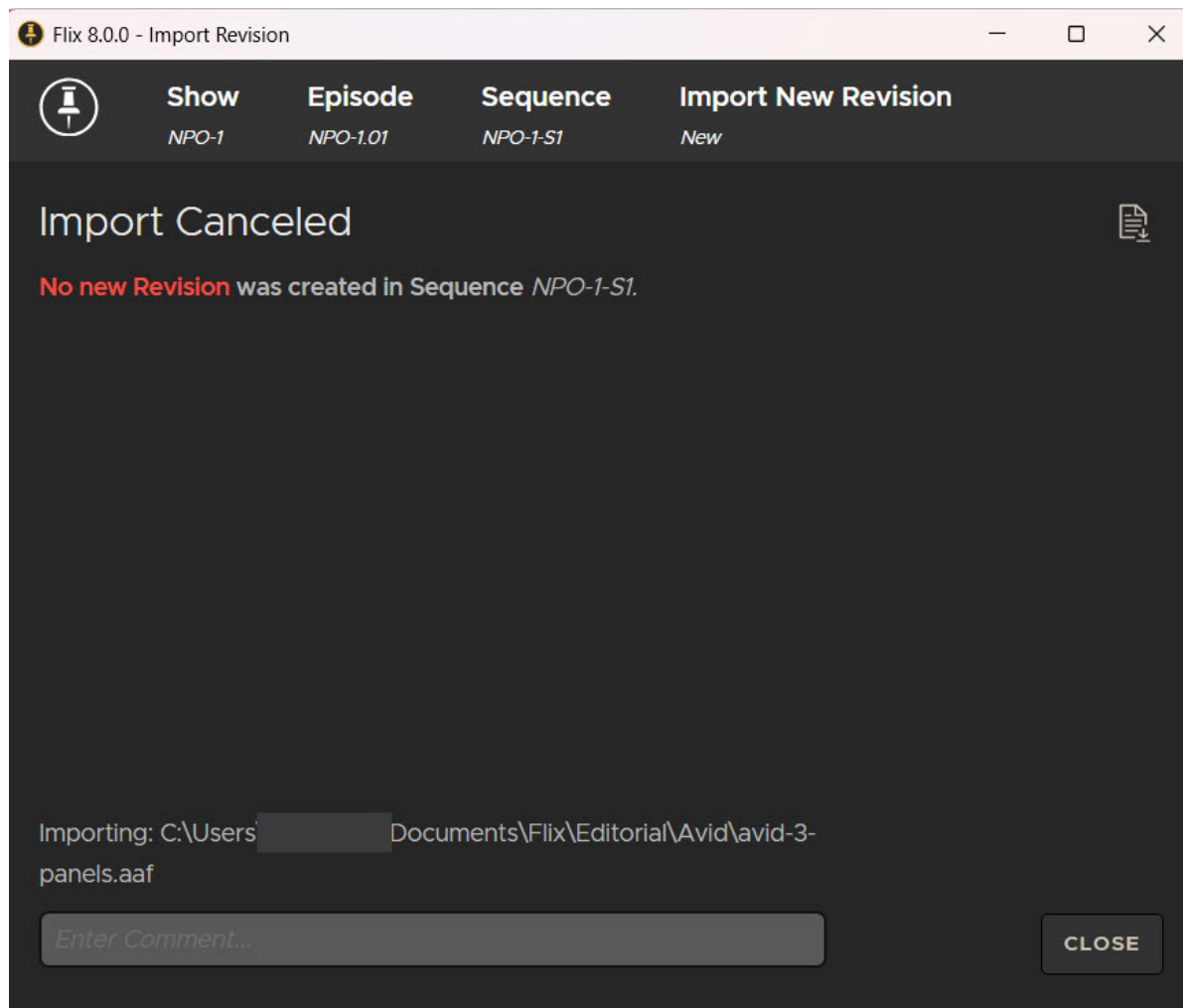


Note: **Open after import** will only happen on a successful import.

It is possible to cancel an import at any stage. You will still be able to Download an import log for canceled imports. Once the import has been completed, the cancel button can no longer be used.



Note: If you cancel an import after the **Processing Files** stage, a partially created sequence may exist in your revisions. However it will be marked with a **Canceled** comment and hidden from your revisions list.



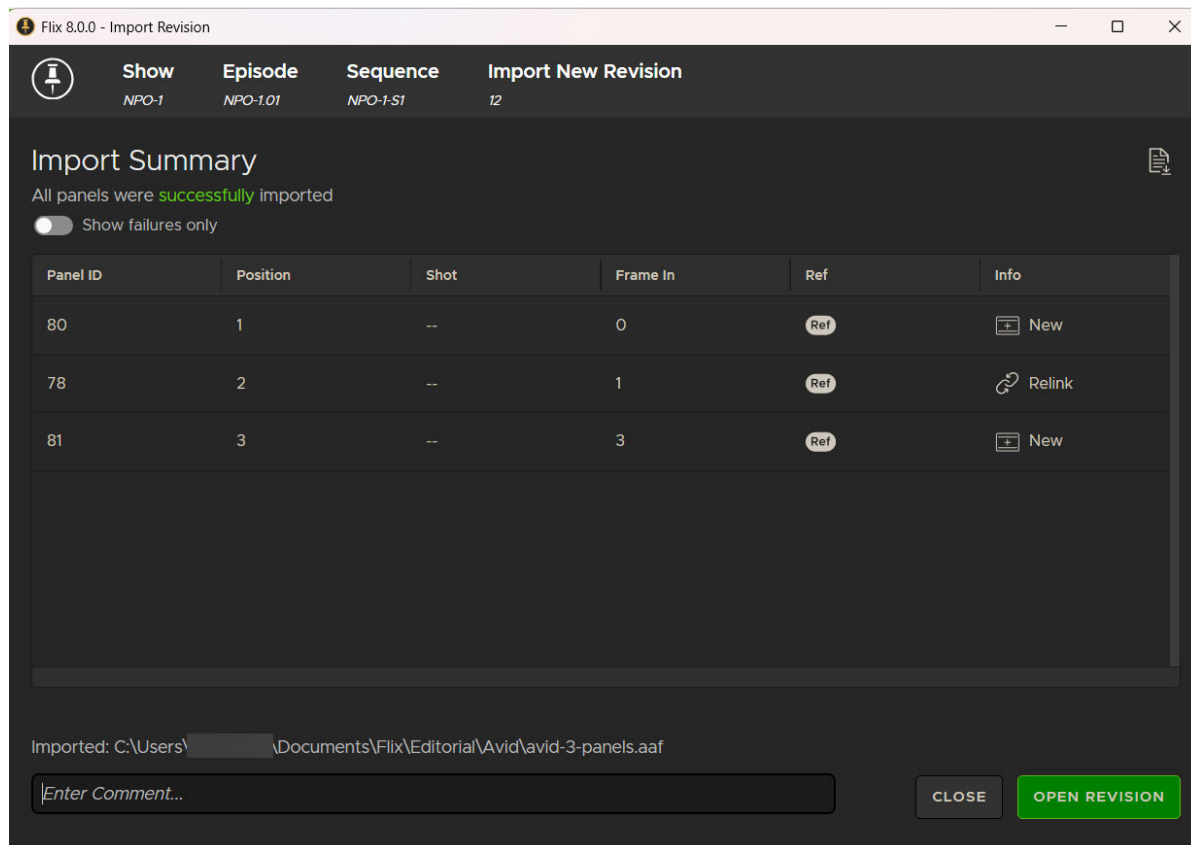
Note: Closing the import window during an import will cause the import to cancel. A warning window will pop up for confirmation of cancellation.

If an error occurs during import, an error notification banner will appear on the screen. You can then inspect which section has caused the error. The imported sequence will also be marked as errored in the revision list.

If an error occurs for a panel artwork import or transcode, it will be displayed as a panel error in the final panel list.

Viewing Import Summary

Once the import has completed, the Import Revision window will display an Import Summary. This summary will contain the **Panel ID**, panel **Position**, **Shot**, **Frame in**, **Ref**, and **Info**. The panels are ordered by the panel position.

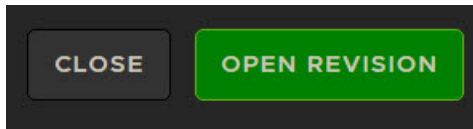


The **Info** column displays whether the panel is relinked directly to an existing panel (**Relink**), versioned-up an existing panel (**Version-up**), or imported as a new panel (**New**). There is also a tooltip that indicates whether the panel is brand new to flix, being reused from another sequence or is a duplicate of another panel in the sequence.

You can use the **Show failures only** toggle to display the failed panels.

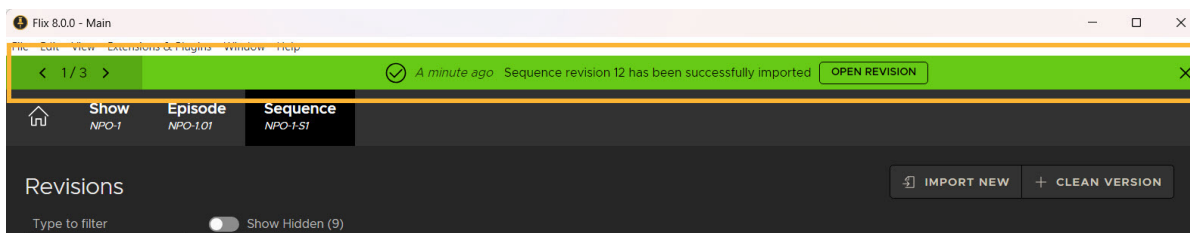
Opening New Revisions

Once the import completes, an **Open Revision** button will appear at the bottom right of the Import window.



Clicking the button will close the Import window and navigate to the new Revision in the main Flix panel browser. Clicking the **Close** button will close the Import window without opening the new Revision.

If you're not currently on the import window, an import success banner will appear on your current window. From this banner you can **Open Revision** (which will also close the import window) or **Show import window**.



Import Logs

You can get import logs for both successful and failed imports using the **Download Import Log** button () found at the Sequence breadcrumb level.

The import log includes:

- The Client logs
- The Server logs (for all servers)
- The TU logs
- A summary of the import results including:
 - Panel data in the table.
 - The start/end time of the import process.
 - The show, sequence and revision IDs.



Tip: You can choose whether to download the full log bundle or just the summary file at **File > Preferences > Upload Full Log Bundle After Imports**.

Once the Storyboard Pro import is completed, a new sequence revision is created in Flix from this import. Scene numbers show as Shots on sequences imported from Storyboard Pro into Flix.

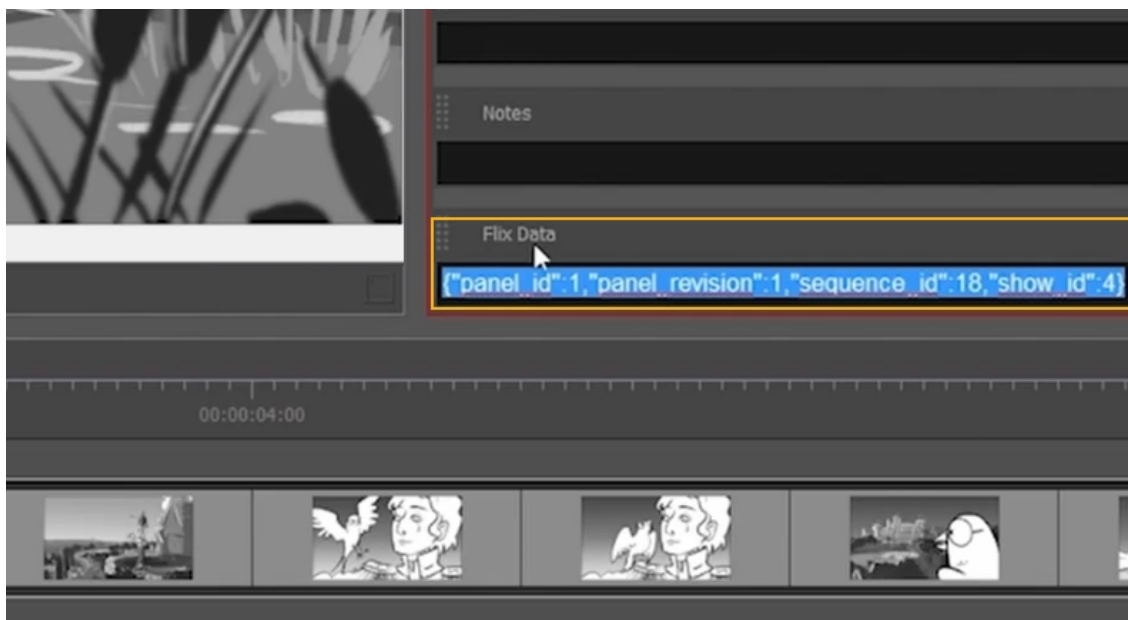


Note: Transitions do not affect panel duration, but the marker denoting the start of the second panel is not displayed in the Flix UI, so you'll only see one marker at the start of the first panel with the transition. Marker and transition data is still available in editorial and Storyboard Pro when you export from Flix, but Flix does not currently support transitions natively.

Smart Panel Relinking Between Flix and Storyboard Pro

Flix's smart panel relinking system means panels can originate from anywhere. Flix sends its own panel metadata to Storyboard Pro to assess which panels need to relink, which ones need to version up, and which ones are completely new, and Flix recognizes them when re-importing those panels from Storyboard Pro.

Exporting panels from Flix writes metadata into each panel containing its ID, revision, sequence, and show information. You can see that metadata in Storyboard Pro in the **Flix Data** field associated with each panel.



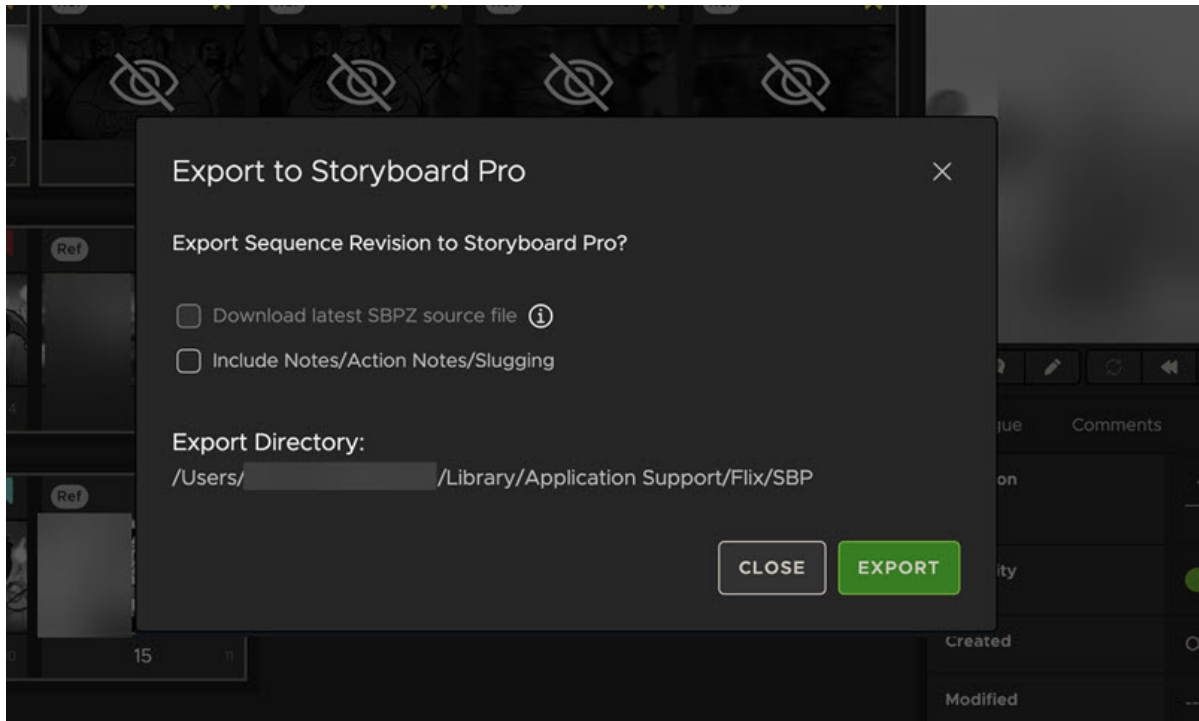
The same is true of panels in Flix. Each panel has an **Origin** in the panel **Properties**.

Dialogue	Comments	Properties	Dialogue	Comments	Properties	Dialogue	Comments	Properties
Duration		12	Trim Animation			Duration		39
Artist		joel.admin	Artist		admin	Artist		roy.yang
Created		May 5, 2022, 3:51	Created		May 12, 2022, 1:00	Created		Jun 2, 2022, 2:00
Modified		--	Modified		--	Modified		Jun 2, 2022, 2:00
Origin		Manual Import	Origin		Storyboard Pro	Origin		Photoshop
Published		×	Published		×	Published		×

Exporting Flix Sequences to Storyboard Pro

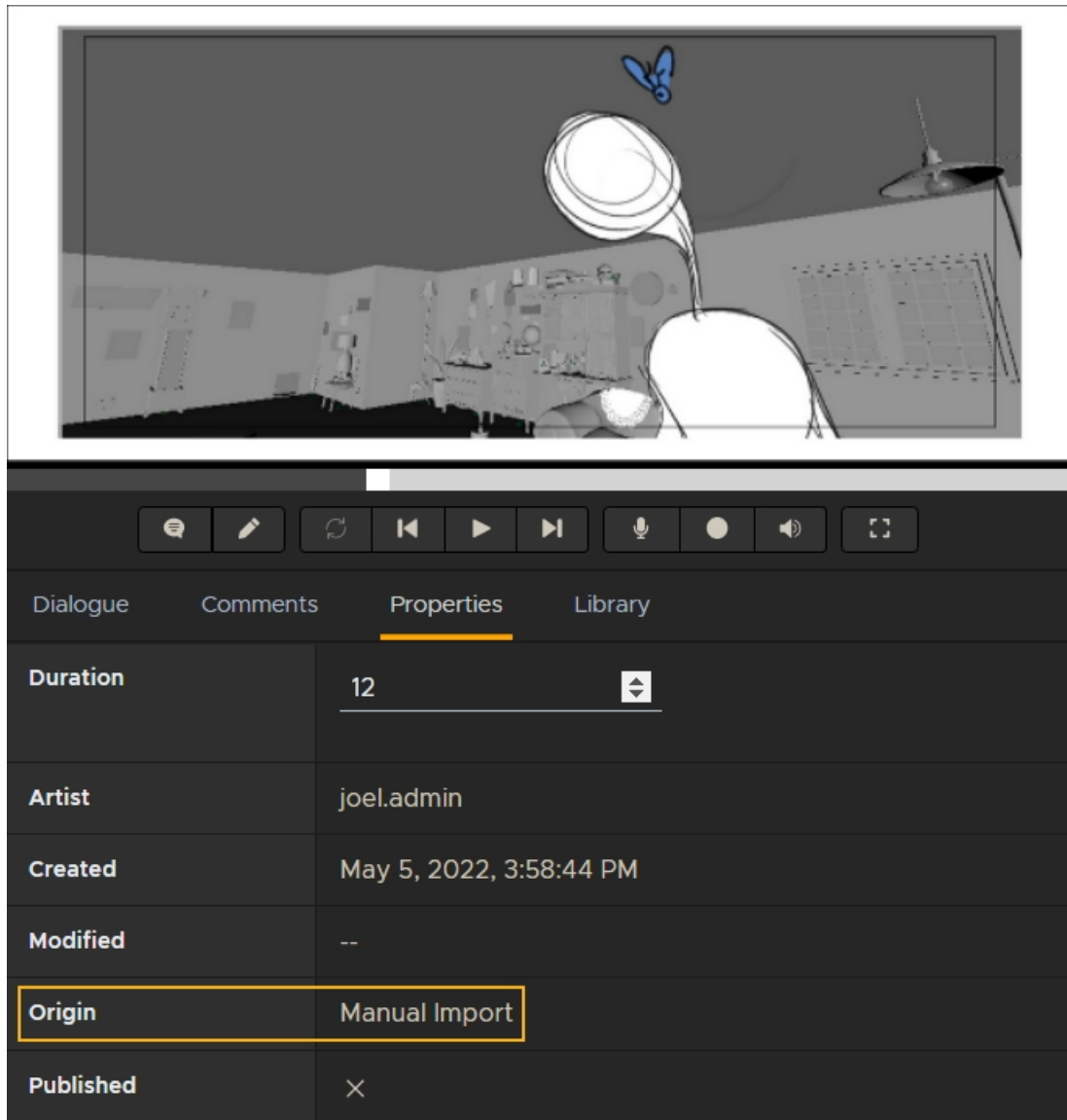
To open your Flix sequence in Storyboard Pro:

1. Ensure Storyboard Pro is the default sketching tool. To check this, please read [Setting Up Flix for Storyboard Pro](#).
2. Open the sequence revision you want to send to Storyboard Pro.
3. Click on the **Open in Sketching App** button, or double-click on a panel.
4. Select the **Download latest SBPZ source file** option if you want to download the latest available SBPZ based on the current revision. If ticked, the file will be exported to the location set in User Preferences, and will open at the end of export.
5. Select whether to **Include Notes/ Action Notes/ Slugging**.

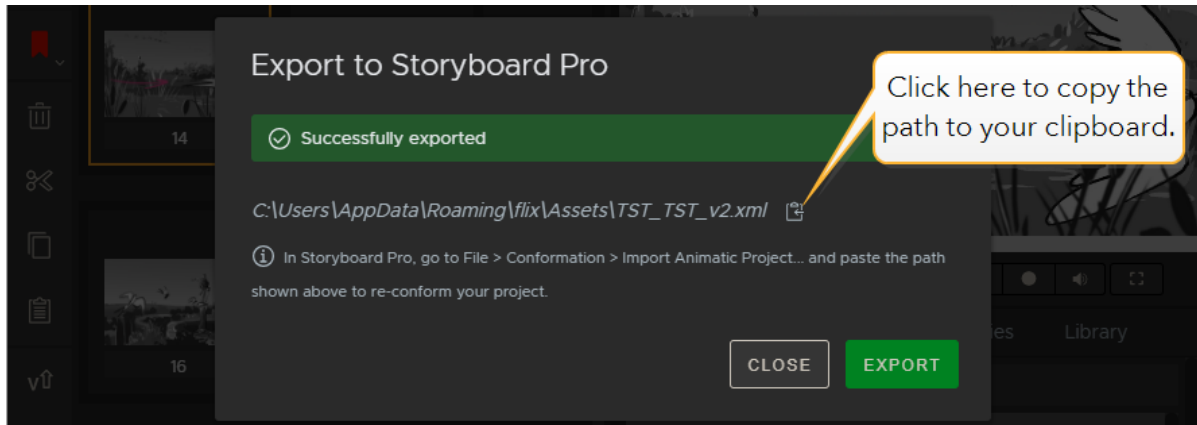


6. Click **Export**.
7. Flix creates a **.xml** file and saves it in the location nominated in your exports path.

The **.xml** Flix generates for the export includes Flix metadata for each panel in the revision, including new panels that Storyboard Pro has never seen. When these new panels are sent back to Flix from Storyboard Pro, Flix looks for that metadata to version them up instead of creating new panels with new IDs, which helps avoid confusion resulting from the ID change. The metadata is displayed in the panel **Properties** tab under **Origin**.



Note: New panels from Flix containing metadata are only versioned up automatically once, the first time that they are reconformed and sent back to Flix from Storyboard Pro. See [Smart Panel Relinking Between Flix and Storyboard Pro](#) for more information on panel relinking.



Note: To change the default path for exports, go to **File > Preferences > Third Party Apps > Storyboard Pro > Export**.



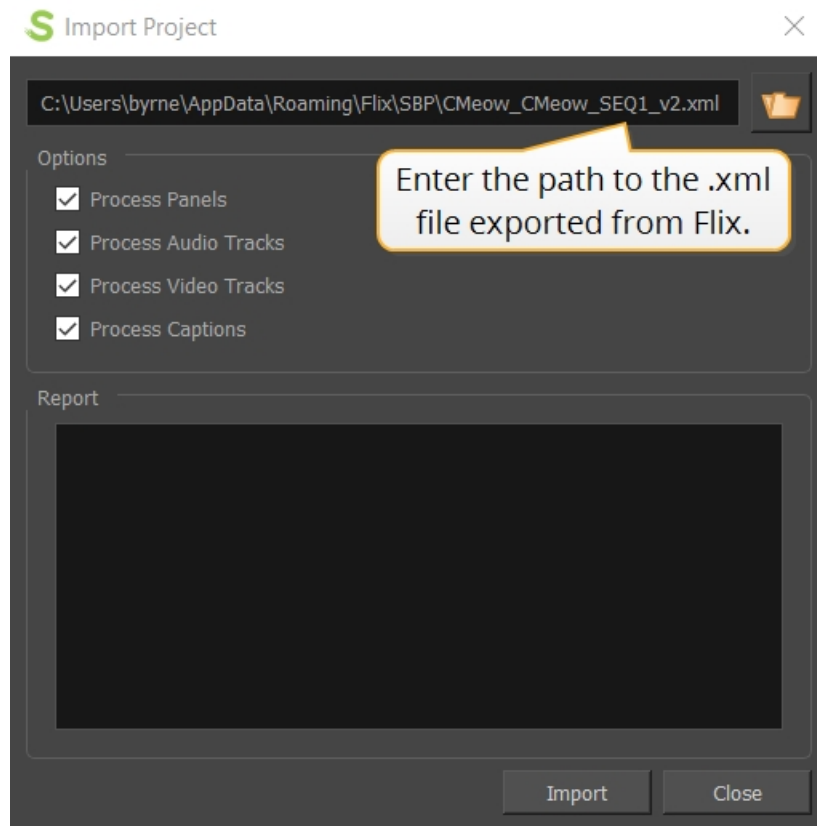
Tip: If you only need to send specific panels to Storyboard Pro (instead of entire sequences), you can use the [Storyboard Pro Extension Workflows](#).

Re-Conforming Your Project in Storyboard Pro

After creating an **.xml** file from your project in Flix, you can update your original project in Storyboard Pro and continue working when changes come back from editorial and Flix. This process is called reconforming, which aligns the original Storyboard Pro project with your changes from Flix and editorial.

To reconform your original Storyboard Pro project, follow these steps:

1. In Storyboard Pro, go to **File > Conformation > Import Animatic Project**.
2. The **Import Project** dialog opens.



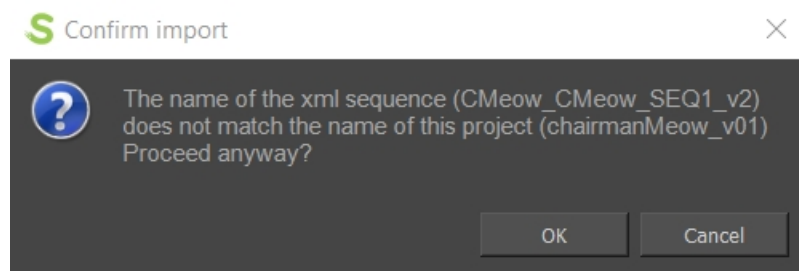
3. Enter the path to the **.xml** file exported from Flix, or click the  icon and browse for the file.



Tip: You can copy the path from the **Export to Storyboard Pro** menu in Flix by clicking the  button and pasting the path in the **Import Project** dialog. See [Exporting Flix Sequences to Storyboard Pro](#) for more information.

4. Click **Import**.

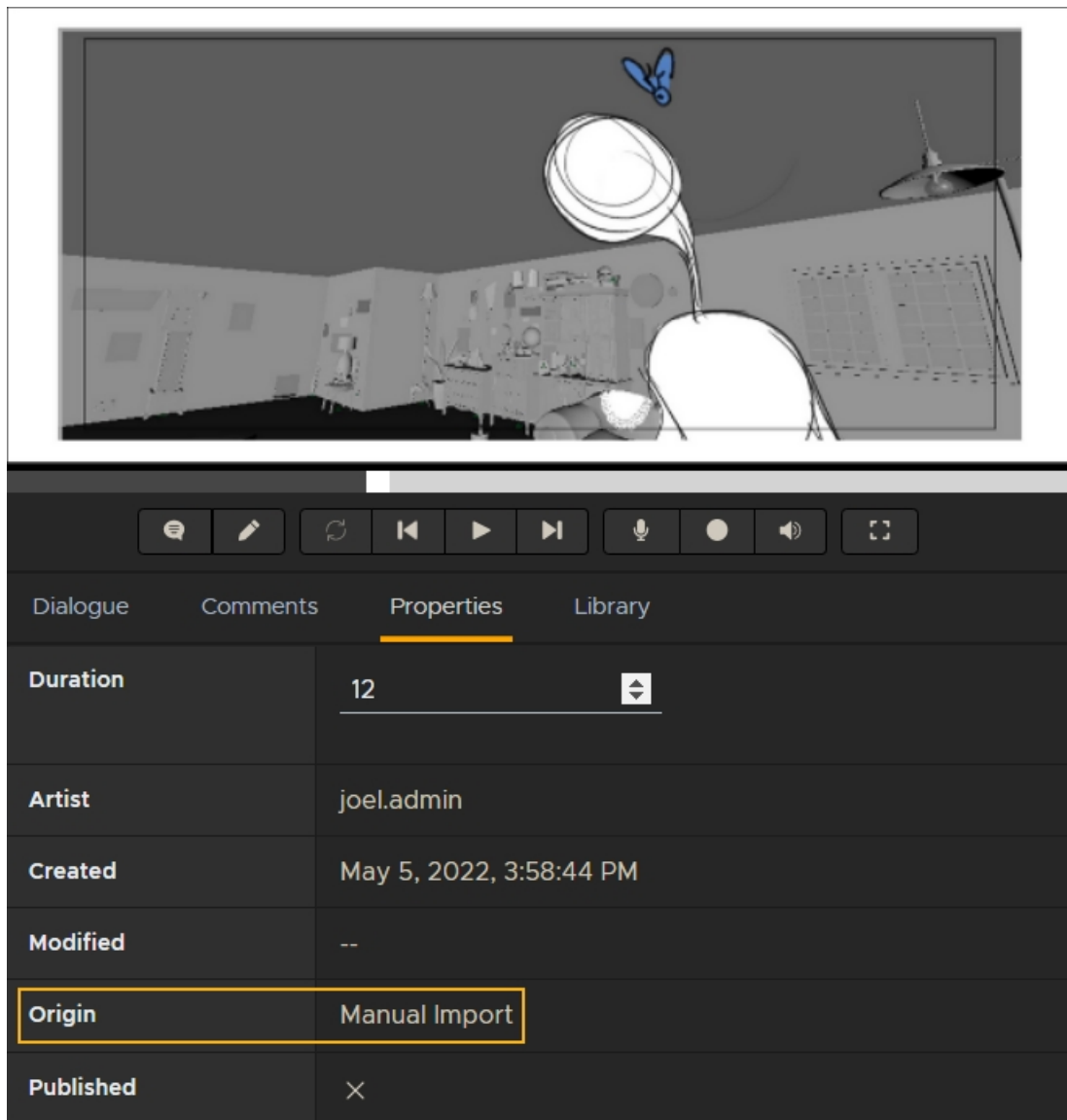
Storyboard Pro asks for confirmation of the sequence you are importing. You may see a warning that the project name is different to the file being imported. This is expected, as Flix is likely to have been set up with different export naming conventions.



5. Click **OK** to proceed.

Your Storyboard Pro sequence updates with any changes that came from Flix. From here you can make any changes to panels as you work and then re-export your work back to Flix as normal.

The **.xml** Flix generates for the export includes Flix metadata for each panel in the revision, including new panels that Storyboard Pro has never seen. When these new panels are sent back to Flix from Storyboard Pro, Flix looks for that metadata to version them up instead of creating new panels with new IDs, which helps avoid confusion resulting from the ID change. The metadata is displayed in the panel **Properties** tab under **Origin**.



- 6.



Note: New panels from Flix containing metadata are only versioned up automatically once, the first time that they are reconformed and sent back to Flix from Storyboard Pro. See [Smart Panel Relinking Between Flix and Storyboard Pro](#) for more information on panel relinking.

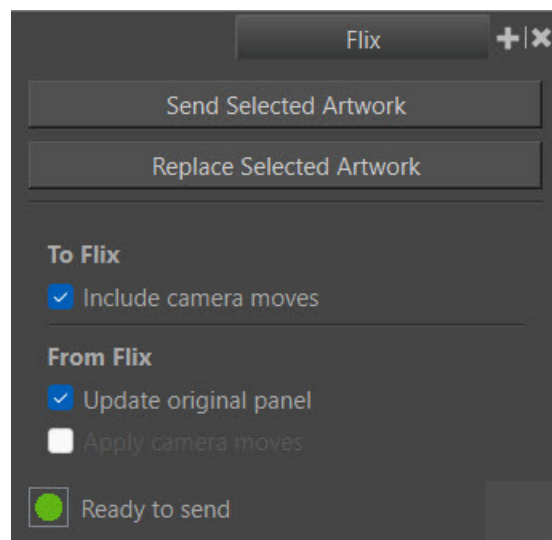
Storyboard Pro Extension Workflows

Checking your Flix/ Storyboard Pro Connection

In Storyboard Pro, the Flix extension window has a traffic light style for the connection status between the programs. This can be found at the bottom left corner of the Flix plugin window.

- Green - Connected and ready to send
- Yellow - Connected, but not ready to send
- Red - Offline and won't send

The **Send Selected Artwork** and **Replace Selected Artwork** buttons are disabled when the panels are not ready to send. The Replace button is disabled when multiple panels or shots are selected.



Sending Panels between Storyboard Pro and Flix using the Extension

You can Send and Replace individual panels between Storyboard Pro and Flix.

Panel relinking is maintained throughout the Send and Replace workflows with Storyboard Pro. For example, when you send a single panel to Flix, and then import the entire Storyboard Pro project into Flix, the already-sent panel will relink automatically.

Storyboard Pro > Flix

- **Send Selected Artwork** inserts the panel(s) after the currently selected panel in Flix.
- **Replace Selected Artwork** overwrites the currently selected panel in Flix with the image from Storyboard Pro. This option will always version-up the panel in Flix.

Send

Sending a panel always creates a new panel ID.

Storyboard Pro panels with layer transforms are always sent as artwork+MOV. The rendered Flix thumbnails match the layer transform. When panels with layer transforms are published to Editorial, they will be sent as prerendered MOVs.

Check the **Include camera moves** button to send camera move information (and its keyframe data) along with the artwork when sending a panel from SBP to Flix. This will mean the camera move can be viewed and edited in editorial.

When you send or replace panels from Storyboard Pro to Flix, any updated dialogue will also be sent over, with line breaks preserved. Note that the Dialogue field in Storyboard Pro must be set to **Default**.

Replace

Replace creates a new version of the selected panel in Storyboard Pro, and overwrites the currently-selected panel in Flix. The Flix panel will be versioned-up, and you will be able to access previous versions by clicking on the panel and viewing the **Revisions**.



Note: You can only replace individual panels. The Replace button will be disabled if multiple panels are selected in Flix.

Flix > Storyboard Pro

To send or replace panels from Flix into Storyboard Pro:

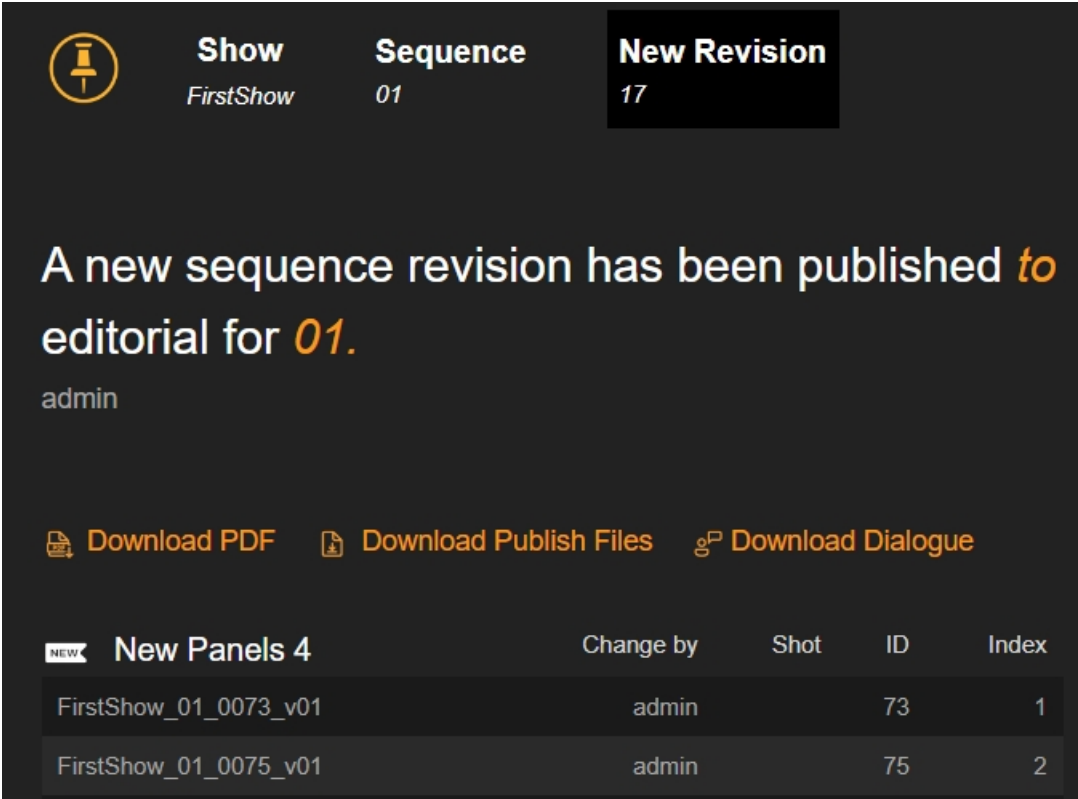
1. Select the Panel(s) you want to send over to Storyboard Pro.
2. Click the **Send to Storyboard Pro** icon (.
3. A transfer bar will appear at the bottom-right corner of Flix to indicate transfer progress.
4. If you have **Update Original Panel** selected in the Flix extension, any panels that already exist in Storyboard Pro will be replaced with the new version you have sent over.

- Any new panels will be sent to Storyboard Pro and added after the currently-selected panel. This will be the case for any existing panels as well if **Update Original Panel** is unticked.

Access Editorial Publish Information from Your Inbox

Publishing to Editorial automatically notifies you with an easy to read email containing a breakdown of what's changed, including links that take you straight to Flix so you can see the new revision immediately. No one can access the links without the correct Flix credentials, so your changes stay confidential and protected within your production.

The **Management Console** > **Permissions** tab controls who is notified by email about publishes using the **Shows** > **Receive Email** permission. The **Regular user** and **Show manager** roles are emailed by default, but you can add new roles to control who is notified by publishes. See [Creating Groups and Assigning Roles to Manage Permissions](#) for more information.



Change by	Shot	ID	Index
admin		73	1
admin		75	2

Each publish email contains the following links and information:

- Clickable redirects to the relevant show, sequence, or revision in Flix.
- Sequence revision comments.
- Publish **to** Editorial only: A link to published files and Contact Sheet.



Note: The link redirects you to Flix Client and requires you to log in for security purposes before you can access the link.

- Publish **to** Editorial only: Panel comments added or resolved since the previous publish to Editorial.
- Information on:
 - New panels.
 - New panel revisions.
 - Deleted panels.
 - Panel duration changes.
- Publish **from** Editorial only: Camera move changes.

Flix Reference Guide

Flix Server Options

Copyright © 2025 The Foundry Visionmongers Ltd.

This is the full list of options for the **config.yml** server configuration file.

Required Options

The following options must be specified to run a Flix server.

hostname - IP address or fqdn for the Flix Server to run on, e.g. flix001.mycompany.com or localhost



Note: You will need to make sure that the **hostname** option is set to a publicly available hostname or IP address. We recommend a fully qualified domain name and unique hostname for each server.

http_port - The port number for the Flix Server HTTP API to run on. By default, this port is set to **8080**.



Note: Ensure that Flix Server is accessible through the server's firewall to connect to it. Refer to the [Opening ports on your firewall for Flix communication](#) Knowledge Base article for more information on how to open ports through a firewall.



Tip: We recommend pinging the Flix server hostname via the command line to test for accessibility to the server.

client_transfer_port - The port number the server listens on for RPC transfers from the Flix Client transfer utility. By default, this port is set to **9091**. You only need to add this option if you want to change the default port.

mysql_hostname - Address of the db server, IP or fqdn.

mysql_username - MySQL username with access to the Flix schema

mysql_password - MySQL password

MySQL

`mysql_port` - Port number for the MySQL database. Default: 3306.

`mysql_database` - Name of the database for Flix. Default: `flix`.

`mysql_dial_timeout` - The maximum amount of time to wait when attempting to connect to a database. Default: 10s.

`mysql_read_timeout` - The maximum amount of time to wait when reading incoming data. Default: 5m.

`mysql_write_timeout` - The maximum amount of time to wait when writing outgoing data. Default: 10m.

`mysql_transaction_timeout` - The maximum amount of time to wait before aborting a stalled database transaction. Do not change this unless necessary Default: 10m.

`db_backup_directory` - Provides an alternate path to where the database backup file is created. For example: `/var/flix/db_backups`.

`mysql_max_connections` - Maximum number of active connections allowed to the MySQL database per server. Default: 70.

Asset Storage

`asset_directory` - Path to where assets should be stored. Defaults to the Flix Server install directory

`shared_storage` - Configures server to use shared storage for Flix assets.



Note: If switching from local to shared storage, you will also need to migrate your assets for Flix Server to pick them up with the new configuration. Refer to [Migrating Assets When Switching to a New Assets Directory](#) for more information.

Licensing

`floating_license_hostname` - Address of the license server (if using floating licensing)

`floating_license_port` - Port number for the license server to run on (if using floating licensing)

`licence_directory` - Path to the license (if using node-locked licensing)

Authentication

LDAP (Lightweight Directory Access Protocol)

For guidance on formatting, please refer to the example at the end of this section.

The following options must be specified to run a Flix server.

`use_ldap` (optional) - This turns on or off the LDAP authentication method for this server. Values: **true** or **false**.

`base` - The base dn is the point from where a server searches for users in your LDAP/AD. You must supply at least the Domain Component (DC).

`host` - The hostname or IP address of your LDAP/AD server.

`port` (optional) - The port number to be used when connecting to the LDAP/AD server.

`use_ssl` (optional) - This indicates whether or not to use SSL/TLS when connecting to your LDAP/AD server. Values: **true** or **false**.

`bind_user` (optional) - This is an account that binds to the LDAP server and performs user and group searches. It can be a read-only account. Make sure the bind user you want to use has permissions to search through the desired paths. The value of this setting can be in one of the following formats:

`username`

`cn=username,dc=domain,dc=com`

`username@domain`

`bind_pass` (optional) - The password for the name provided in `bind_user`. If you don't use `bind_user`, or if it does not require a password, you don't need to set this.

`self_auth` (optional) - If this is set, `bind_user` and `bind_pass` are ignored. Instead, Flix attempts to use the username and password from the user logging in to bind.

User Search

dn (optional) - DN from where to start the search from. If this value is not set the **base** will be used.

filter (optional) - Filter to apply when searching the directory. Specify the **objectClass** for your users. The default value is: **(objectClass=organizationalPerson)**

user_attr - The attribute to use for the username matching for the authentication. On most AD servers, the default setting is - **sAMAccountName**.

name_attr (optional) - The attribute used to return the user's full name. On most AD servers, the default setting is - **displayName**.

email_attr - Defines a custom attribute for the user email address to be retrieved from, other than the default 'mail' attribute. This might be useful in cases when the mail field is used for personal email addresses and the cn field for company email addresses.



Note: It is not currently possible to specify which users/groups should be notified upon Editorial publishes. However, the email_attr option does make it possible to retrieve a different mail attribute for users, which can remain blank in LDAP for those who don't wish to receive notifications for Editorial publishes.

Group Search

dn (optional) - DN from where to start the search from. If this value is not set the 'base' will be used.

filter (optional) - Filter to apply when searching the directory. The default value is empty.

user_attr - The name of the attribute from the user search which can be found in a group attribute such as **member**. Common values are **distinguishedName**, **uid**, **sAMAccountName**.

group_attr - The group attribute that has the same value as the user attribute set above. On most AD servers the default setting is 'member'.

name_attr (optional) - The name of the group. On most AD servers the default setting is either 'name', 'cn' or even 'description'.

group_prefix (optional) - Only groups that start with this string will be added to Flix when a user logs in.

group_suffix (optional) - Only groups that end with this string will be added to Flix when a user logs in.

Auto-assign Group-Role Pairs

LDAP groups can be used to automatically assign group-role pairs. This is done by a new YAML section within the LDAP section roles. See [Example LDAP Flix Config File](#) for more information.

Additionally, four optional fields have been added:

allowed_users - an explicit list of LDAP usernames which are permitted to log into Flix (cannot be set if blocked_users is set).

blocked_users - an explicit list of LDAP usernames which are not permitted to log into Flix (cannot be set if allowed_users is set).

required_groups - a list of groups users must have to be permitted to log into Flix, users must have all groups in this list. Cannot be set if forbidden_groups is set.

forbidden_groups - a list of groups which if a user has any of, that user will not be permitted to log into Flix. Cannot be set if required_groups is set.

Example LDAP Flix Config File



Note: This example is for illustration purposes. The entry preceding the ':' is a key that Flix reads, which needs to be named as in the example, but the entry following the ':' follows the exact naming of the attribute name in your AD.

```
ldap:
  use_ldap: true
  base: dc=flix,dc=ad
  host: 10.10.10.10
  port: 385
  use_ssl: false
  self_auth: false
  bind_user: CN=Flix,OU=Flix-Users,DC=flix,DC=ad
  bind_pass: PASSWORD
  user_search:
```

```

dn: OU=Flix-Users,DC=flix,DC=ad
filter: (objectClass=organizationalPerson)
user_attr: sAMAccountName
name_attr: displayName
email_attr: description
group_search:
  dn: OU=Groups,DC=flix,DC=ad
  filter: (objectClass=group)
  user_attr: distinguishedName
  name_attr: name
  group_attr: member
  group_prefix: flix-
  group_suffix: -flix
roles:
  - role: Show manager
    group_suffix: "Manager"
  - role: View only
    group_suffix: "Readonly"
  - role: Regular user
    group_prefix: "Flix"
  - role: View only # fallback role

```

Email

smtp_hostname - The hostname e.g., smtp.gmail.com

smtp_port - Port number e.g., 465

smtp_username (optional) - SMTP email address e.g., example@email.com

smtp_password (optional) - SMTP password e.g., MyP@ssword

smtp_send_from - Sets the email address Flix uses for notifications, e.g., flix_publishes@mycompany.com



Note: If the smtp_username and smtp_password config options are not set, Flix Server attempts to connect to the smtp server without authenticating when sending notification emails.

HTTPS

ca_file - Add this option and the path to a CA certificate file if using self-signed certificates.

cert - Add this option and the path to a TLS certificate file (public key).

key - Add this option and the path to the TLS key file (private key).

webui:

enabled: true - Allows your server to accept connections from a web browser. This is for running Flix Client through a web browser.

Proxy

host - The hostname of the reverse proxy server. This is the URL that Flix Client uses to connect to the server HTTP API.

port - The HTTP(s) port the reverse proxy server is listening on.

transfer_port - The grpcs port the reverse proxy server is listening on for file transfers.

Security

disable_hostname_check boolean [default: false] - If set to true, Flix Server does not check that the host header requested by the Flix Client matches the hostname of the server. This can be used for reverse proxy setups where the reverse proxy's public URL does not match the Flix Server's hostname.

skip_transfer_tls boolean [default: false] - If set to true, Flix Server listens for insecure (non-TLS) RPC connections. This can be used for reverse proxy setups where the reverse proxy terminates the TLS connection (grpcs://) from the Flix Client and uses an insecure connection upstream to the Flix Server.



Note: Flix Clients always communicate using gRPCs, never using insecure (non-TLS) RPC connections.

Logs

max-log-mb - Sets default log file size to 5 MB. After this data limit is reached, the log file rotates and the older data is split off and stored in archived logs.

log_file - Sets the filename and location for server logs. The CLI flag will make the directory if it doesn't exist. For example: /var/flix/logs/serverlogs.log.

log_thread_info - Determines what information is included in the logs using 'true' to include and 'false' to exclude. All values are true by default.

- user - the user who started that 'thread' of function calls.

- `client_id` - the Flix client ID.
- `server` - the originating server.

Other

`font_directory` - Path to access fonts

`flix5_compatible_imports` - Use in Flix 6 when reconforming Avid AAFs created in Flix 5 to relink correct panels.

`webhooks` - Custom event-based API triggers that automate standard processes can be enabled or disabled on a per-server basis using the `enabled: true` or `false` options.

See <http://docs.flix-dev.com/#tag/Webhook> and <https://pypi.org/project/flix-sdk/> for more details on Flix's API and the **flix-sdk** package.

Flix Preferences

To access Flix's Preferences, navigate to **File > Preferences** on Windows and **Flix 8.0 > Settings** on macOS.



Note: These preferences can be enforced at the Studio and Show levels, see [Settings & Preferences](#)

General

Setting	Description
Show Splash Screen	Enables or disables Flix's splash screen at start-up.
Enable Autosave	Enables or disables autosaving of sequence revisions.
Automatically Hide Autosaves	Toggle on to hide autosaves for all users.
Autosave Frequency	Set Flix's autosave to 5, 10 or 30 minute intervals.
Allow Save Comments	Enables or disables the option to save comments for revisions.
Maximum Banner	Choose how many notifications stack until the oldest is dismissed

Setting	Description
Notifications	automatically. Default is 3.
Default Flix Panel Duration	Change the default duration of panels. Default is 12 frames.
Upload Full Log Bundle After Imports	Whether Flix should upload a full log bundle of client, TU, and server logs after a sequence revision import, or just the import summary file. Default is enabled.
Reset Preferences	Resets all preferences to their default values.

Third-Party Apps

Sketching Tools

Setting	Description
Sketching Tool	Specifies the external application used to edit panels, Adobe Photoshop or Storyboard Pro.

Storyboard Pro

Setting	Description
Executable	Specifies the file path to the version of Storyboard Pro you want to use.

Setting	Description
	 Note: The Executable directory setting for both Windows and Mac is available at the studio or show level, for cases where multiple users might be on different operating systems.
Store Source File	Whether to store the Storyboard Pro Source File. Default is enabled.
Export Path	Set the path for your exported Flix sequence.

Adobe Photoshop

Setting	Description
Executable	Specifies the file path to the version of Storyboard Pro you want to use.  Note: The Executable directory setting for both Windows and Mac is available at the studio or show level, for cases where multiple users might be on different operating systems.
Store Source File	Whether to store the Storyboard Pro Source File. Default is enabled.
Panel Open Behavior	Specifies how to open panels in Photoshop: • Open as Separate PSD - opens the selected panels in Photoshop as separate .psd files.

Setting	Description
	• Open in Layer Comps - opens the selected panels in Photoshop as layer comps in a single file. • Open in Timeline - opens the selected panels in Photoshop in the Timeline, as frames in a single file.
Always Open Source File	When enabled, opening a panel in Photoshop opens the Source File, if one exists. Flix opens the .psd file with all information including hidden layers and empty groups.
Send Annotation as Layer	When enabled, annotated panels open in Photoshop with their annotations shown on an additional layer.

Editorial Tools

Setting	Description
Editorial Tool	Specifies the external application used for editorial roundtripping, Avid Media Composer or Adobe Premiere.

Adobe Premiere

Setting	Description
Use Flix Timing	Toggle ON to send every panel as a clip retaining its duration set within Flix. This is

Setting	Description
	useful to retain the timing established by a recorded pitch, saving you time and avoiding redoing work. See Preserve Sequence Timings from Flix for more information.  Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Avid. If you disable this control, you can only send .mov files to Avid and the Panels with Camera Move control is disabled.
Send / Receive Camera Moves As	Set how you send camera moves with panels. Choose between Images with Keyframes or Rendered Animations. When Use Flix Timing is disabled, panels with camera moves will be sent as Rendered Animations.
Mark In	When Enable Mark In is ON, sets the frame on which to set the Mark In point. The default is 120 (5s in for a 24 fps show).
Mark Out	When Enable Mark Out is ON, sets the frame on which to set the Mark Out point. The default is 360 (15s in for a 24fps show). When Use Flix Timing is enabled, Mark Out is automatically set based on each panel's duration.

Setting	Description
	 Warning: Setting your Mark Out point to a value lower than the Mark In point may create an unreadable .aaf file.
New Clip Color	Sets the color of new clips in your sequence.
Enable Audio	Controls whether audio is available in Avid Media Composer or not.
Send / Receive Shots	When enabled, shots will be included when publishing from and exporting to editorial
Send / Receive Shots As	How shots are sent to and received from editorial. Choose between Markers and New Track.
Shot Name Sent / Received As	Define whether the Marker name or Marker comment should be taken as the Shot name.
Marker Type	Choose whether Flix sends Timeline or Clip markers to Premiere. • Timeline markers apply to a particular timestamp in the sequence and appear on the timeline. • Clip markers apply to a whole clip within a sequence and appear at the beginning of the clip.

Setting	Description
	 Note: The Marker Type you choose means that Flix only reads that type. For example, if you have Clip selected and Editorial add Timeline markers in Premiere, they are ignored by Flix.

Avid Media Composer

Setting	Description
Use Flix Timing	Toggle ON to send every panel as a clip retaining its duration set within Flix. This is useful to retain the timing established by a recorded pitch, saving you time and avoiding redoing work. See Preserve Sequence Timings from Flix for more information.  Warning: Use Flix Timings must be enabled if you want to send camera move keyframes from Storyboard Pro to Avid. If you disable this control, you can only send .mov files to Avid and the Panels with Camera Move control is disabled.
Send / Receive Camera Moves As	Set how you send camera moves with panels. Choose between Images with Keyframes, Rendered Animations, or New Track.

Setting	Description
	When Use Flix Timing is disabled, panels with camera moves will be sent as Rendered Animations .
Mark In	When Enable Mark In is ON, sets the frame on which to set the Mark In point. The default is 120 (5s in for a 24 fps show).
Mark Out	When Enable Mark Out is ON, sets the frame on which to set the Mark Out point. The default is 360 (15s in for a 24fps show). When Use Flix Timing is enabled, Mark Out is automatically set based on each panel's duration.  Warning: Setting your Mark Out point to a value lower than the Mark In point may create an unreadable .aaf file.
New Clip Color	Sets the color of new clips in your sequence.
Source Media Duration	The maximum number of frames available for editing still media in Media Composer. Ensure the Mark In and Mark Out fall within this range.
Color Range	Sets either Full or Legal range color for publishes to Avid.  Note: The default color range is set to Full.
Codec	Specifies the type of Avid video codec used for exporting to Media Composer: • DNxHD 36 (DNxHD LB)

Setting	Description
	• DNxHD 45 (DNxHD LB) • DNxHD 115 (DNxHD SQ) • DNxHD 120 (DNxHD SQ) • DNxHD 145 (DNxHD SQ)  Note: Higher data bit rates generally produce higher quality images, but the files exported require more storage space and take longer to transfer.
Enable Audio	Controls whether audio is available in Avid Media Composer or not.
Send / Receive Shots	When enabled, shots will be included when publishing from and exporting to editorial
Send / Receive Shots As	How shots are sent to and received from editorial. Choose between Markers and New Track.
Shot Name Sent / Received As	Define whether the Marker name or Marker comment should be taken as the Shot name.

Publish and Export

Publish



Note: The chips, located at the bottom of the preferences panel show what information type you can include in the filename format. Click each field to reveal which information type is allowed. Invalid info types are grayed out. See [Setting Naming Conventions](#) for more information.

Setting	Description
Directory	Specify where Flix will deliver your editorial publishes on Windows. Environment variables can be used to construct these paths (e.g. S{HOME}). Click on the chips available below to construct paths using project metadata (e.g. [show_tracking_code])
Editorial Sequence Name	Name of the sequence as shown in Editorial.
Editorial Clip Name	Name of the clip as shown in Editorial.

Export



Note: The chips, located at the bottom of the preferences panel show what information type you can include in the filename format. Click each field to reveal which information type is allowed. Invalid info types are grayed out. See [Setting Naming Conventions](#) for more information.

Setting	Description
Directory	Specify where Flix will deliver your editorial publishes on Windows. Environment variables

Setting	Description
	can be used to construct these paths (e.g. S{HOME}). Click on the chips available below to construct paths using project metadata (e.g. [show_tracking_code]). Select whether to set the Same as the Publish Directory .
Sequence Name	Name of the sequence as shown when exported. Select whether to set as Same as the Publish Editorial Sequence Name .
Panel/Clip Name	Name of the Panel/clip. Select whether to set as Same as the Publish Editorial Clip Name .

QuickTime Export

Setting	Description
Include Dialogue	Toggle ON to export dialogue as subtitles when exporting from Flix to QuickTime.  Note: Ensure your video player has subtitles or closed captions enabled to display the exported dialogue.

Audio

Setting	Description
Input Device	Specifies the audio input device to use when recording audio.
Output Device	Specifies the audio output device to use when playing audio.

Panel Browser

Setting	Description
Swap Panel ID with Index	Swaps the position of each panel's unique ID number with the Panel Index (a panel's position in the sequence).
Animated Panels	Sets the default preview behavior for animated panels: • Preview all frames - Animated panels show all available frames in the panel viewer. • Preview first frame - Animated panels only show the first frame in the panel viewer and the Trim Animation control the Properties is disabled.  Note: You can override this preference on individual panels by clicking the icon in the top left to select  Preview all frames or  Preview first frame. Changing this preference does not affect individual panels that you have set manually.
Display Hidden Panel Placeholders	Select whether to display hidden panel placeholders - default is enabled.

Shot Settings

Setting	Description
Automatically Delete Panels when Deleting a Shot	Select whether to automatically delete the panels that make up a shot when you delete the shot.
Shot Name Format	The name of the shot.

Setting	Description
	 Note: The chips, located underneath this preference, show what information type you can include in the filename format. Click each field to reveal which information type is allowed. Invalid info types are grayed out. See Setting Naming Conventions for more information.
Minimum Length	Sets the minimum character length for your shots.
Increment	Sets the value of increments between shots. For example, if the shot number format is set to increments of 10, your first shot is labeled 0010, your second is 0020, 0030 etc.

Notifications

You can select which of your assigned shows trigger notifications when updates are published to and from Editorial.

Setting	Description
Publish to Editorial	Check the box per show to receive email notifications when panels are published to Editorial for the show. Toggle the checkmark in the box next to Publish to Editorial to activate or deactivate all notifications.
Publish from Editorial	Check the box per show to receive email notifications when panels are published from Editorial for the show. Toggle the checkmark in the box next to Publish from Editorial to activate or deactivate all notifications.

Advanced

Setting	Description
Temp Directory	Specifies where Flix stores temporary files.
Asset Cache Directory	Specifies where Flix caches your assets temporarily to improve responsiveness.
Clear Local Asset Cache	Click to remove the temporary files stored in the local asset cache directory.
Disable Panel Reuse	In the normal course of operation, Flix's intelligent versioning system automatically attempts to reuse an existing panel revision instead of creating a new panel ID or new panel revision. For troubleshooting and test purposes, you may want to disable versioning by enabling this preference. When enabled, Flix always creates a new panel ID or new panel revision instead of reusing an existing panel revision.

Flix Logs

There are four components to Flix that generate logs.

- **Flix Server** - Backend Processing
- **Flix Client** - User Interface
- **Flix Transfer Utility** - File Transfer
- **Flix Extension** - Plugins

Each uses its own logging system, which generates its own log files. System Administrators can use the following section to check what each log file records and where to find them.

Server Logs

Flix Server logs actions taken at various different levels, such as errors, warnings and information levels. This is designed to help you identify issues and to assist with troubleshooting bugs or configuration problems. Actions that are executed on the server side vary from the Client. The server logs creation of shows, sequences, panels etc and updates. The server also logs interactions with the MySQL database, so you can expect information in the server logs if you are experiencing issues with your database. Any interactions with the File System, namely your Assets directory, are also logged by the server. A general rule-of-thumb is that when Flix is computing, or persisting data, these operations occur on the server side.

Flix Server logs can be found on the server machine, at the location specified by the **log_location** option. If this option has not been set in your server config file, Flix Server saves log files in the directory from which it is being executed.

An example log line is shown here:

```
time="16 Sep 20 15:11 BST" level=debug msg="moving media object file"
func="logging.logrusIntegrate.Debug:" Destination=/home/jimmy/flix/flix-
server/assets/f22a1072-f675-4baa-b737-77edcd81f8fd/3/549_462809382.png
Source=/tmp/Flix762493943/152544291/462809382.png ThreadClientID=638745ba-df66-
4547-b83c-2ab90ae216c5 ThreadServer="Server {f23a6794-b675-4bcd-b327-
77aaed81f8fd | 192.168.1.67}" ThreadUser="User{jimmy.flix Admin user}"
```

By default, log lines contain the user from whom the request came, **ThreadUser**, the client, **ThreadClientID**, and the originating server, **ThreadServer**.

To change this behavior, set the 'user', 'client_id', and 'server' options to 'false' in your **config.yml**. For example, to display the User and Client ID, but not the Server, set the following options:

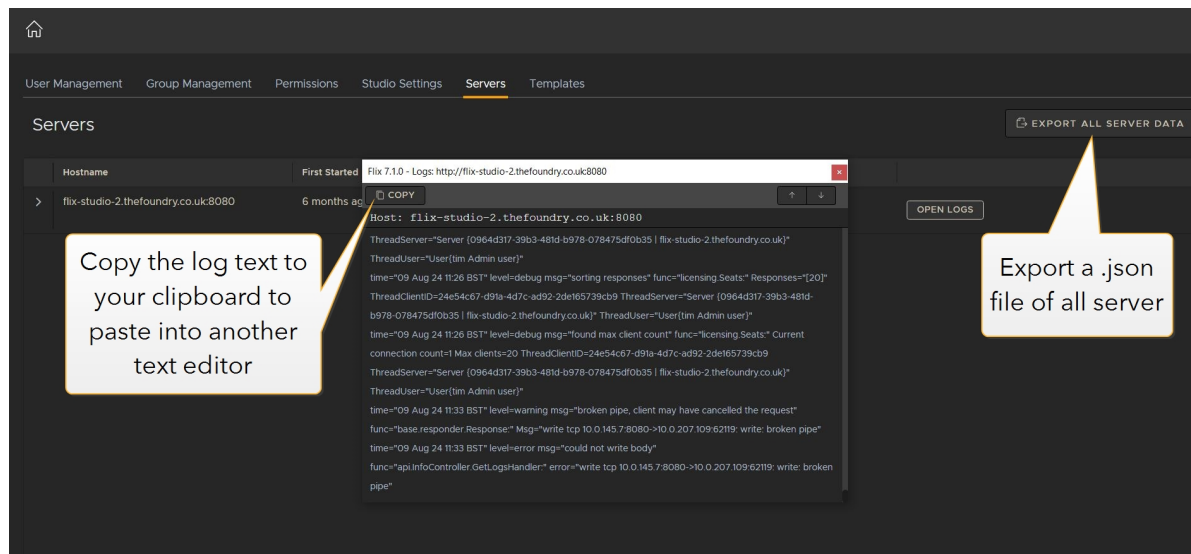
```
log_thread_info:
  user: true
  client_id: true
  server: false
```

Accessing Server Logs in Flix Client

Go to **File > Management Console > Servers**.

A list of all your running servers is displayed.

To access the log file for each server, click the **Open Logs** button against each server.



Client Logs

The client logs record actions performed by the Flix Client during runtime. The contents of the log file show differing levels of logging, such as Errors, Debugs, and Warnings. Errors are logged when something in Flix Client fails, such as an import, or a publish. Debugs and Notices are useful as an indication of actions being performed. Warnings indicate potential issues with Flix Client and could flag potential problems. It's prudent to understand what is causing a Warning to appear in the logs, so you know whether it needs addressing or not.

Client logs can be found here:

Mac: **~/Library/Logs/Flix-Client/flix-client.log**

Windows: **%APPDATA%\Flix-Client\flix-client.log**

Transfer Logs

The Flix Transfer Utility manages file uploads and downloads between Flix Client and Flix Server. Any information regarding potential issues occurring while transferring files can be found in this log file.

The Transfer logs can be found on end users' machines at the following location:

Mac: **~/Library/Logs/Flix-Client/flix-client-transfer-util.log**

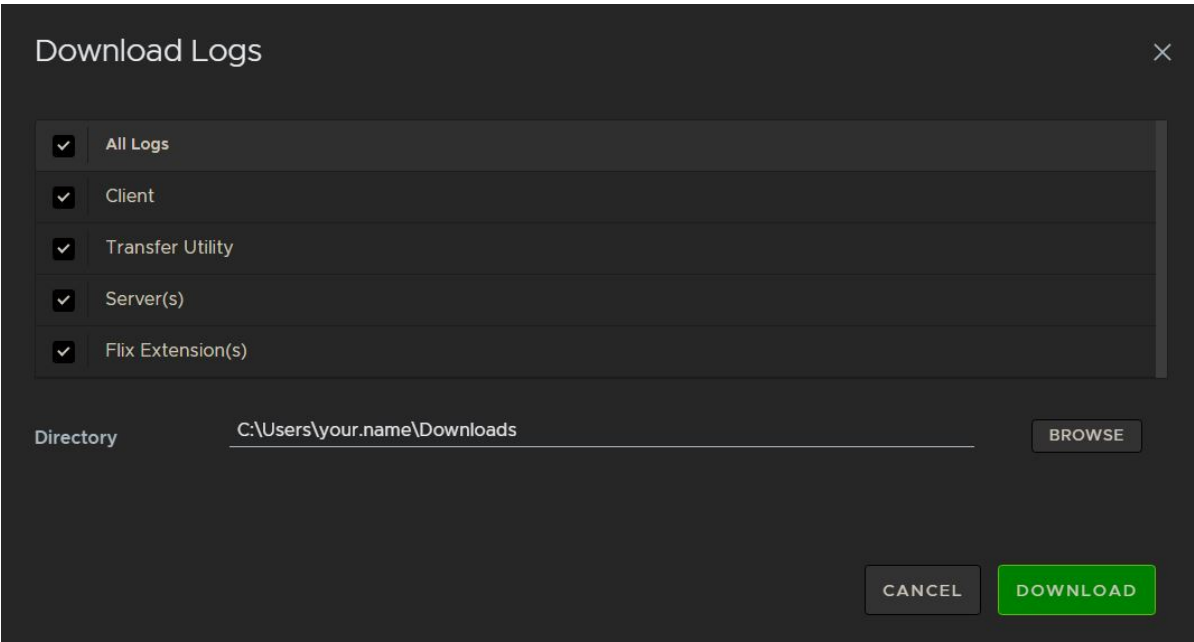
Windows: **%APPDATA%\Flix-Client\flix-client-transfer-util.log**

Download Logs

Admin users can choose to download specific logs or all logs in one go from the **Download Logs** menu.

To download logs:

1. Go to **Help > Download Logs**.
Select the logs you want to download or leave **All Logs** selected.
2. Choose which directory to download the file to and click **Download**.



Keyboard Shortcuts

Project

Command	Shortcut
Open the Preferences	Ctrl/Cmd +, (comma)
Open the Management Console	Ctrl/Cmd +. (period)
Save	Ctrl/Cmd +S

General Edit Actions

Command	Shortcut
Undo	Ctrl/Cmd+Z
Redo	Ctrl/Cmd+Shift+Z
Delete	Backspace
Select All	Ctrl/Cmd+A
Copy	Ctrl/Cmd+C
Paste	Ctrl/Cmd+V
Cut	Ctrl/Cmd+X
Save	Ctrl/Cmd+S
Publish sequence revision	Ctrl/Cmd+P

Player

Command	Shortcut
Add an Audience to the Viewer	A
Exit Full Screen Mode	Esc

Panels and Shots

Command	Shortcut
Go to above Panel	Up arrow
Go to next Panel	Right arrow

Command	Shortcut
Go to below Panel	Down arrow
Go to previous Panel	Left arrow
Add panel above to selection	Shift+Arrow Up
Add panel down to selection	Shift+Arrow Down
Add panel left to selection	Shift+Arrow left
Add panel right to selection	Shift+Arrow right
Create new panels	Ctrl/Cmd+N
Open selected panels in Photoshop / Open Export to SBP Modal	Ctrl/Cmd+Enter
Cut Panels	Ctrl/Cmd+X
Copy Panels	Ctrl/Cmd+C
Paste Panels	Ctrl/Cmd+V
Select All panels	Ctrl/Cmd+A
Go to Panel	Ctrl/Cmd+G
New Shot	Ctrl/Cmd+Shift+N
Select shots from panels	Ctrl/Cmd+Arrow Up
Select panels from shots	Ctrl/Cmd+Arrow Down
Select previous shot	Ctrl/Cmd+Arrow Left
Select next shot	Ctrl/Cmd+Arrow Right
Add previous shot to selection	Ctrl/Cmd+Shift+Arrow Left
Add next shot to selection	Ctrl/Cmd+Shift+Arrow Right
Edit shot name	Enter (when a shot is selected)
Save shot name	Enter (when editing shot name)

Command	Shortcut
Cancel editing shot name	Escape (when editing shot name)
Add selected panels to shot	Enter (when a shot and some additional panels are selected)
Delete Panels/ Shots	Backspace OR Delete
Opens shot sequential rename modal	F2

Window

Command	Shortcut
Full Screen	F11
Zoom In	Ctrl/Cmd+Shift+= (equal) Do not use the = key from the numeric keypad
Zoom Out	Ctrl/Cmd+- (minus) Do not use the = key from the numeric keypad
Reload	Ctrl/Cmd+Shift+R
Minimize	Ctrl/Cmd+M
Source File library	Ctrl/Cmd+L
Extension window	Ctrl/Cmd +/
Management Console	Ctrl/Cmd + .
User Preferences	Ctrl/Cmd + ,

Flix Resources

Copyright © 2025 The Foundry Visionmongers Ltd.

Flix Top Five Things to Onboard New Story Artists

Flix is designed to speed up and manage the various workflows feeding into the creation of a story. Storyboards, Dialogue, Notes from the director, annotations, every version sent and received from Editorial; it's all fed in and managed within Flix. As a Story Artist, you'll be working primarily in your sketching app, like Storyboard Pro. Flix makes your life easier as it tracks every version of a story sequence, acting as the hub that brings together everyone working on the story.

Here are the top five things that will help Story Artists get up and running quickly, so they can get back to focusing on creative tasks.

In the video:

1. [Importing Artwork into Flix](#)
2. [Panel ID vs Panel Index](#)
3. [Editing Existing Panels](#)
4. [Adding Dialogue Text](#)
5. [Adding Audio to a Pitch](#)

1. Importing Artwork into Flix

The first thing you need to do is make sure your boards are in Flix. This makes sure they are backed up securely and that everyone in the production can see them. Once the boards are in Flix, they can be arranged into a sequence and the director can make notes, add annotations or send the sequence on to Editorial.

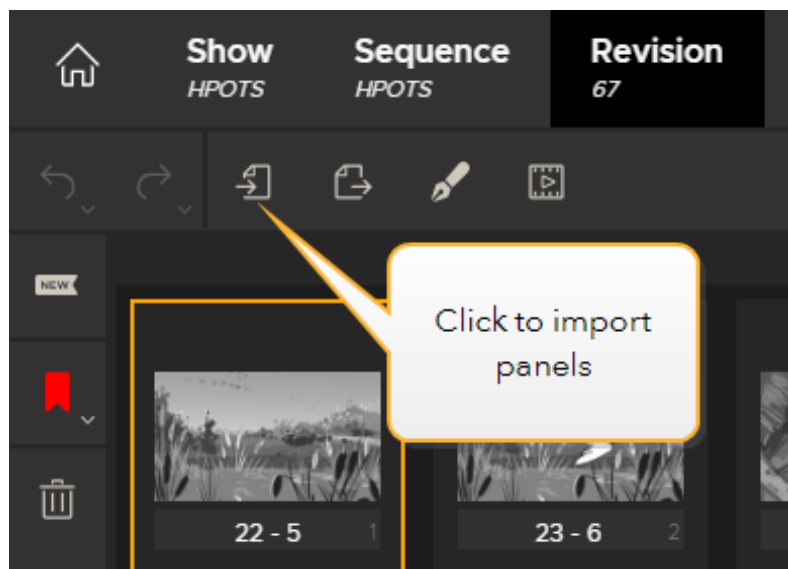
Flix supports multiple image formats, so artists can bring boards directly into Flix to construct shots and sequences. Imports can be flattened **JPGs**, **PNGs** and **TIFFs** you've already created, and **PSDs** from Photoshop or **.sboard** and **.sbpz** (packaged project) files from Storyboard Pro, if your artwork has layers you'd like to retain.



Note: Importing **.sbpz** files provides an optimized experience using less storage and reducing backup issues or loss of data.

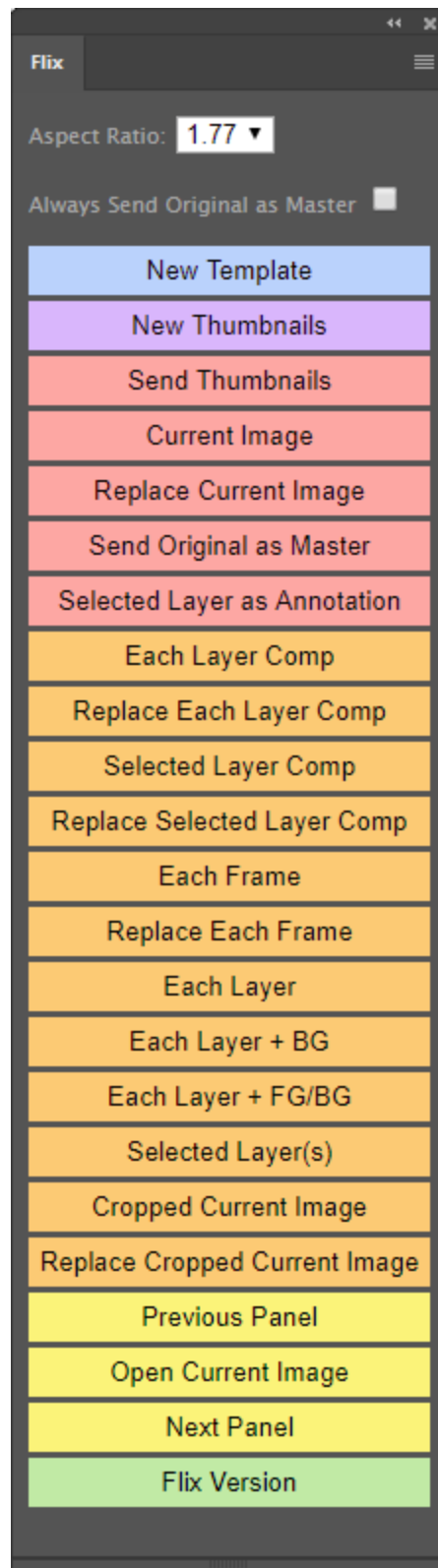
Importing Image Files

To import image files, simply drag and drop them directly into Flix's panel browser, or use the Import to Flix button at the top of the panel browser.



Importing PSD Files

You can import a **PSD** using the drag/drop method or the Import to Flix button, however, we recommend using the Flix interactive toolset in Photoshop to send the **PSD** file to Flix. This toolset can send layers, layer comps, thumbnails and frames directly from Photoshop's canvas into Flix's panel browser. See [Flix for Story](#) for more information or check out the Flix & Photoshop [online video course](#).



Flix/Photoshop toolset

Importing Storyboard Pro Files

You can easily import an entire project file from Storyboard Pro using the **SBP Import** button to browse for the desired **.sboard** or **.sbpz** (packaged project) file. This creates a new sequence revision which contains only panels from the Storyboard Pro file. Panels from other sequence revisions can be found in Flix's **Library** tab, and drag/dropped into this new revision. See [Flix for Story](#) for more information or check out the Flix & Storyboard Pro [online video course](#).



Note: **.sboards** and **.sbpz** (packaged project) are the only project file types that Flix supports from Storyboard Pro. Importing **.sbpz** files provides an optimized experience using less storage and reducing backup issues or loss of data.

For more information on importing panels and Flix workflows with Photoshop and Storyboard Pro, see [Flix for Story](#). For video tutorials aimed at Story Artists, click [here](#).

2. Panel ID vs Panel Index

What is the difference between a **Panel ID** and a **Panel Index**?

Each panel in Flix is assigned a unique **Panel ID**. This number never changes, so everyone on a production can trust they are always referring to the exact same panel.



Default positions of **Panel ID** and **Panel Index**.

The **Panel Index** is simply the number that indicates the order of panels. It's always sequential. In other words, if you rearrange the order of panels, the indexes will change but the IDs will not.

If you prefer, you can swap the position of the **Panel ID** with the **Panel Index**.

Go to **File > Preferences > Panel Browser > Swap Panel ID with Index** and toggle the switch ON.

The **Panel Index** now appears in the center of the panel and the **Panel ID** is shown on the right.



Panel ID and **Panel Index** in swapped positions.

3. Editing Existing Panels

Throughout the storyboarding process, your director and colleagues will add annotations and comments in Flix to let you know that a panel or sequence revision needs changes. You can just open up the board straight from the sequence in Flix, rather than digging through folders on your local machine to find the original.

To edit the original Photoshop file:

1. Make sure Photoshop is your default sketching app. To check this, go to **File > Preferences > Third-Party Apps** and check the **Sketching Tool** is set to Photoshop.
2. In Flix's panel browser, select the panel you would like to edit in Photoshop.
3. Double-click the panel or use the **Open in Sketching App** button.

The master Photoshop document opens in Photoshop.



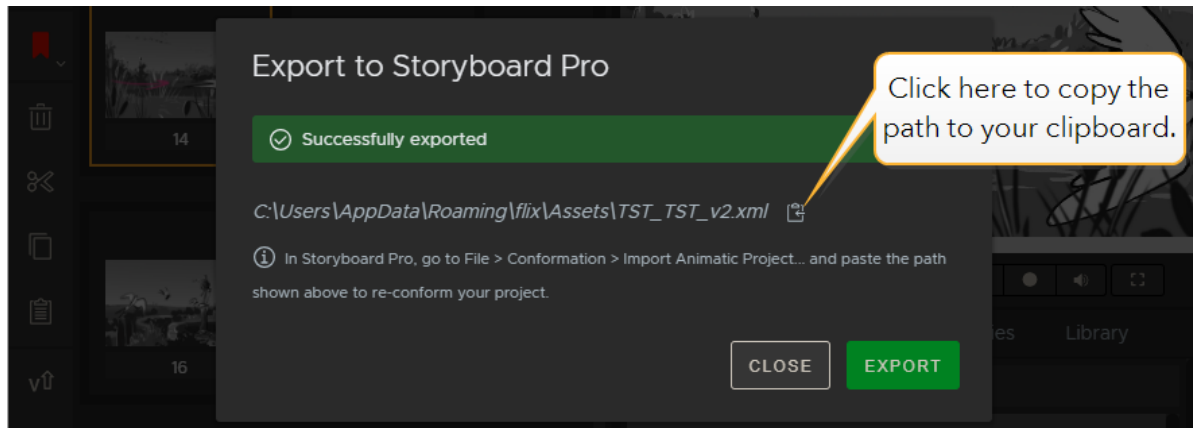
Note: The **Current Image** option in the Flix/Photoshop toolset sends a flattened **PNG** to Flix, so layers and layer comps won't have been retained when that panel is reopened in Photoshop.

To send existing panels to Storyboard Pro for further editing:

1. Make sure Storyboard Pro is your default sketching app. To check this, go to **File > Preferences > Third-Party Apps** and check the **Sketching Tool** is set to Storyboard Pro.
2. Select the panel in Flix.
3. Double-click the panel or use the **Open in Sketching App** button.

Flix prompts you to export the full sequence revision as an **XML** file.

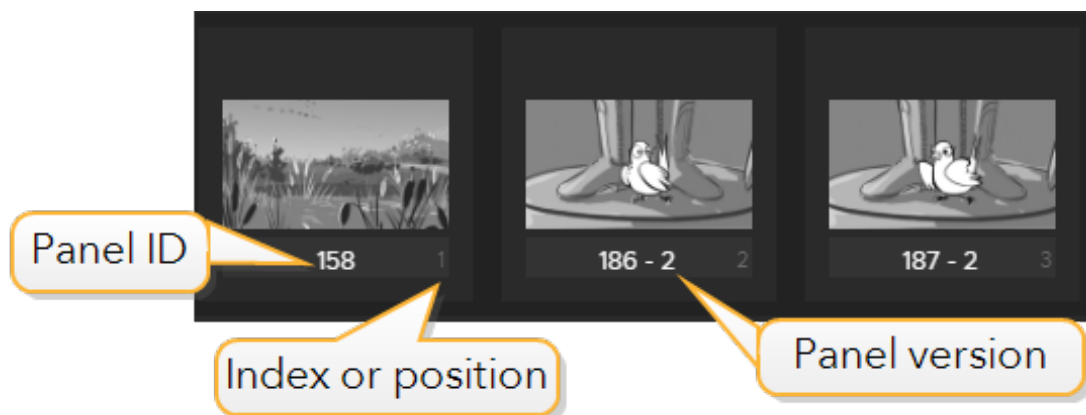
4. Click **Export**.



You can manually import the **XML** into Storyboard Pro, which reconfirms the sequence with all the latest changes you made in Flix.

Switching Panel Versions

Each time you make changes to a panel within Flix, your previous Flix edits are still available to you as panel versions. If it turns out that your team prefers the previous iteration of a panel, just click on the space at the bottom of the panel and select the desired version from the **Revisions** list.

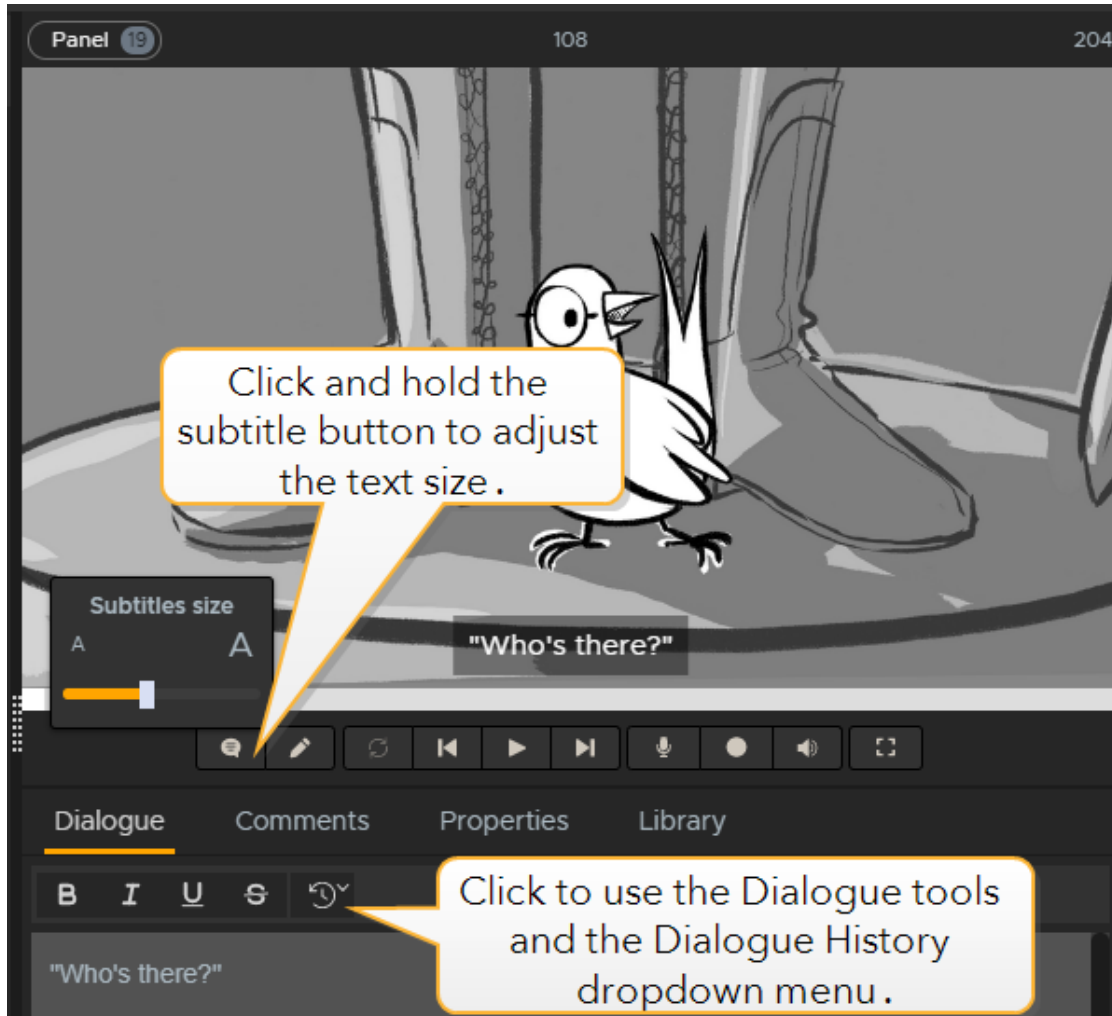


4. Adding Dialogue Text

If you receive script changes or decide to add different dialogue options to shots, you can use Flix to add dialogue to panels.

1. Click on a panel you'd like to add dialogue or other text to.
2. Type your dialogue in the text field in the **Dialogue** tab.

The text appears as subtitles in the Viewer.



Note: You can also select multiple panels and enter text in the Dialogue tab to have that text applied across all selected panels.



Tip: You can switch to use the **Dialogue** workspace if you prefer. Click on the **Switch Workspace** button at the top right of the panel browser.

Each time you revise a panel's dialogue, the panel's previous text is saved in Flix's **Dialogue History**. If you'd like to revert to an earlier version of a panel's dialogue, click on the **Dialogue History** button above the text field, and select the desired iteration.

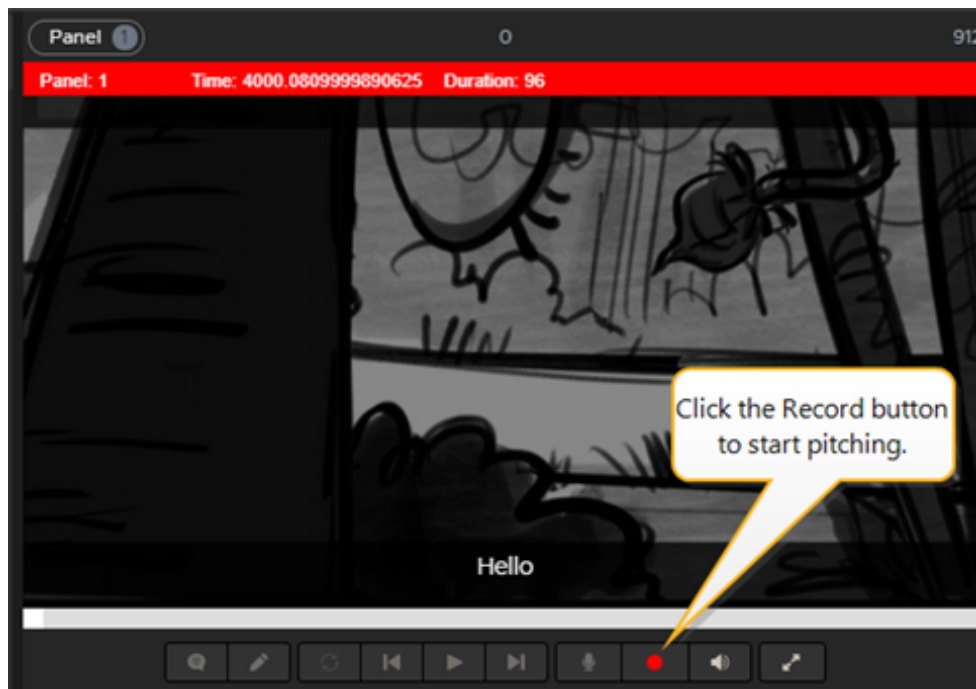
For more information on working with dialogue in Flix, see [Adding or Editing Dialogue](#).

5. Adding Audio to a Pitch

Flix allows artists to add vocals and sound effects to a sequence revision. This means you can get the sequence as close as possible to a worthy representation of the story, so decisions are made without any guesswork. Playing out your sequence in Flix can help artists determine if the pacing for each panel is right, or if the duration of certain panels should be adjusted to fit the overall timing of the sequence. You can even add the silhouette of an audience at the bottom of the viewer, so you're always reminded of the end goal.

1. Press the **Record** button underneath the **Viewer**.
2. Flix plays through the sequence.

Record your audio in time with the sequence revision.





Note: You can use pre-recorded audio to add background music or other sound effects by importing **MP3** or **WAV** files, which are added in time with the panels.



Tip: When playing the sequence, press **A** to add an audience to the bottom of the viewer.

For more information on recording audio, see [Recording a Pitch](#) or the Flix for Story [online video course](#).

Pitching the Final Sequence Revision

You can really only know how it plays when you screen it to a group. So, once your sequence is developed to a point where you're ready to show, you can play it to your team in Fullscreen/Pitch Mode. While in Pitch mode, annotations are not available, but you can toggle dialogue on or off, loop the playback, record additional audio, and adjust the volume.

If you prefer to pitch the sequence the traditional way, you can also simply step forward or backward through the sequence, using either your keyboard arrows or the Go-to buttons.

Flix Knowledge Base Articles

Foundry maintains a growing library of articles in our knowledge base. These knowledge base (KB) articles offer extra information on specific issues or tasks not covered in this documentation.

To see the current set of KB articles, see the [Flix Knowledge Base](#).

Flix Example Scripts

Foundry provides a Git repository of example scripts, templates, and external tools for Flix. This is a useful resource from which you can build your own customizations.

Click the script name to access the folder at the [Foundry Flix Github repository](#).

- **Blender** - Send a scene from Blender to Flix
- **Clipboard** - Send screen captures to Flix
- **Flixdy** - Flixdy - A fully-featured Python SDK for interacting with Flix which includes, amongst other examples,

- Create a Flix Show from ShotGrid project data
- Automate updates to ShotGrid based on actions in Flix
- **Hiero** - Share sequences between Flix and Hiero/HieroPlayer
- **License management** - Retrieve information on seat availability and usage
- **Move shows** - Move a sequence from one show to another
- **Panels media object** - Fetch panels with media objects



Warning: These scripts are not fully supported and may not be compatible with every version of Flix.

Flix Server Technical Overview

This technical overview aims to give Systems Administrators and Tech Teams deploying Flix in a studio environment some context as to what the requirements are, how it works under the hood, and to provide a high level understanding of Flix's overall architecture.

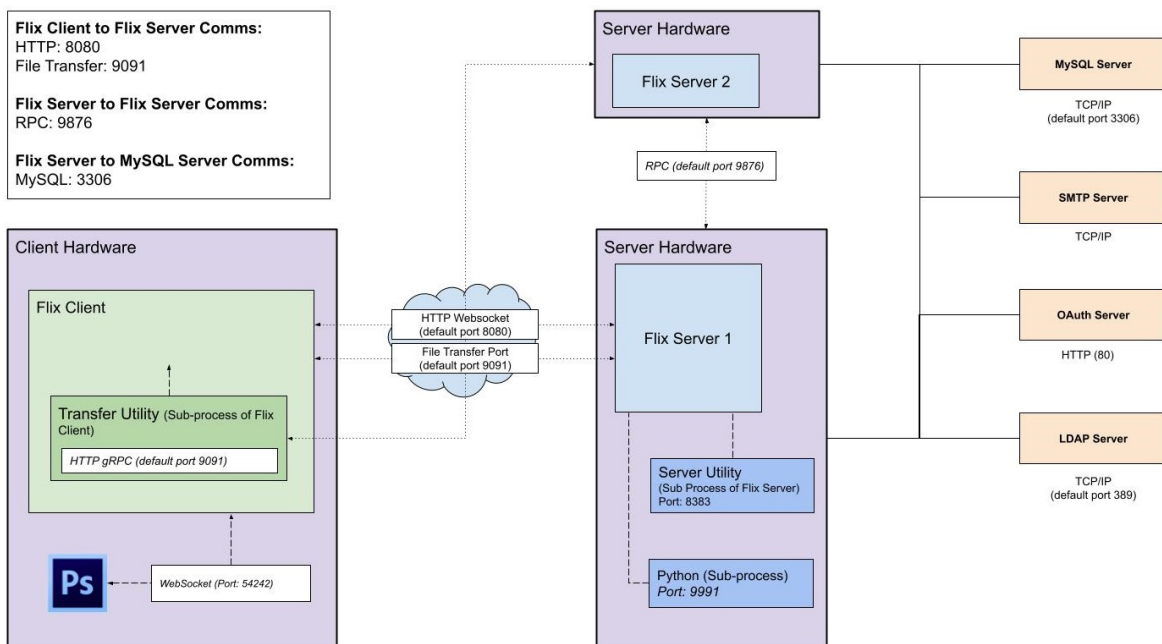
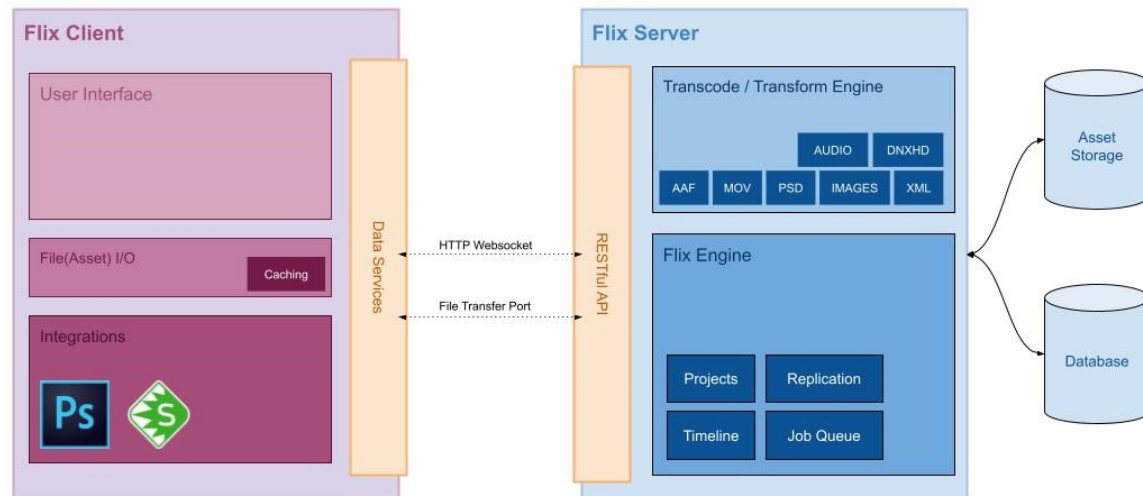
This overview assumes some familiarity with systems administration and client/server architecture.

Architecture

Flix is a client / server architecture application. The client (Flix Client) is the user interface for interacting with Flix, creating boards, viewing sequences, and interaction with third-party tools like Photoshop. The server side (Flix Server) is where the bulk of Flix's processing takes place. Flix Server provides a RESTful HTTP API for Flix Client, or custom scripts, to consume. Communication between Flix Client and Flix Server is done over HTTP(S) and Websockets.

The server utilizes a MySQL database for data persistence of shows, episodes, sequences, panels, dialogue, and so on. Although the database holds image data, such as panel thumbnails or artwork, it doesn't store the actual files or assets. These are stored in a separate asset location. Each Flix Server requires access to the same MySQL server to operate.

Flix Server handles all image transcoding, manipulation, and storage. Asset storage on the server can be local, or on a network share. Flix can be configured to use either option. Network shares must be presented to Flix as a file system directory.



Server Requirements

Flix Server can be installed on a physical server or virtual machine, and it's recommended to have multiple servers set up in your Flix deployment to provide scalability and redundancy. Flix spreads requests across servers equally, to ensure all servers resources are used to their full extent.

A single-server setup can be enough for a small deployment (3-5 users), but a multi-server setup is better suited to a larger user base, especially for workflows which involve greater involvement with Editorial. Servers can be added after the initial setup to scale up when needed.



Note: Flix handles its own load-balancing, so we do not recommend adding another load-balancer to your deployment, such as Varnish.

Hardware Specifications

The recommended hardware specs for Flix Server are highly dependent on usage. The main areas that require considerable resources are image processing and file transfers. Image processing in Flix is mostly handled on the server side, so the server requires resources to do these tasks. These tasks can be CPU intensive, especially the creation of DNxHD for Avid Media Composer. More powerful CPUs process images faster, resulting in faster editorial round-trips and panel creation.



Note: Flix Server does not require a GPU.

Flix Server architecture is built on the basis of concurrency and requires multiple CPU cores to operate effectively. Flix Server requires a CPU with a minimum of 2 cores, but we recommend 4 cores, or more, to allow Flix to multitask more effectively.

The number of tasks Flix Server can perform also depends on the amount of available RAM. More concurrent tasks may require more memory, so having enough RAM available is essential for good performance. A minimum of 4 GB of RAM is required, but we recommend 8 GB or 16 GB for larger installations.

Flix Server can be scaled in two different ways, horizontally and vertically. Horizontal scaling refers to the number of nodes, or servers, in your cluster. Vertical scaling refers to the amount of resources available to each of your nodes, such as CPU cores and RAM. Scaling in each direction provides a distinct advantage to your Flix installation:

Horizontal Scaling

- ⊕ Increases API throughput and allows more Flix Client connections
- ⊕ Reduces load on each node in the cluster (if you have high resource utilization on your Flix Servers, you may want to scale horizontally)
- ⊕ Adds redundancy and prevents downtime if a server outage occurs

- ⊖ Adds pressure to your database server

Vertical Scaling

- ⊕ Improves rendering speed
- ⊕ Reduces time for panel creation
- ⊕ Improves DNxHD rendering speeds
- ⊕ Each server can handle more tasks concurrently

Storage

Flix Server stores and manages all of the assets imported in Flix as panels, audio files, AAFs, and so on, and therefore requires access to a file share with enough free storage. The storage requirements vary heavily depending on the size of the production, its duration, and the type of usage it sees from Flix Clients.

1 TB storage for a feature production is a good recommendation as a starting point, as long as this can be expanded on as needed.

Flix Server stores the assets on the machine it's running on by default, but we highly recommend pointing Flix Server to an external file share. The external file share can then be accessed by multiple Flix Servers to avoid data duplication and to centralize all of your assets. If you configure Flix Server to use a shared network mount, we recommend you use the 'shared storage' feature, to allow any Flix Server to serve assets. Using this feature decreases unnecessary data duplication in your assets directories.

Flix stores all assets in the **Asset** directory, which is configured on the server. This directory is managed by the Flix Servers, and should not be directly accessed by sysadmins or artists.



Warning: We strongly advise against renaming, deleting, or tampering with files from the **Asset** directory, as it may cause failures in Flix or result in missing media.

Flix supports any file system available on the operating system. Flix expects a mounted file system directory with full read and write permissions to function correctly. SMB or NFS are recommended as they are widely available, though NFS setups perform better in some instances.

Assets are partitioned by **Show** in the **Asset** directory for the purposes of archiving, if required. Once a show becomes old, and no longer in use, its assets can be moved out of the **Asset** directory for backup purposes to free up storage availability. However, any access to assets requested for those shows is no longer available and appears offline within Flix.

Database

Flix uses the relational database MySQL for data persistence. MySQL has many features making data storage efficient and scalable and it is a tried and tested database server, ensuring we have a reliable data storage layer.



Warning: Flix MySQL server now requires the **innodb_autoinc_lock_mode** global variable to be set to 1, consecutive mode. If MySQL is configured to use interleaved mode, the server will not start.

All persistent data in Flix is stored in the database, except for the assets, which are stored on the file system separately. However, metadata for the assets, including location and naming, is stored in the database. This ensures we don't have to access the file system when querying asset data, which can be incredibly fast if the data is in the database's caches. This way, Flix doesn't rely on continuous reads that can put a lot of pressure on file system storage.



Article: If you want to learn more about what info is stored in the database and how this is connected to the assets and Flix, take a look at this [Knowledge Base Article](#).

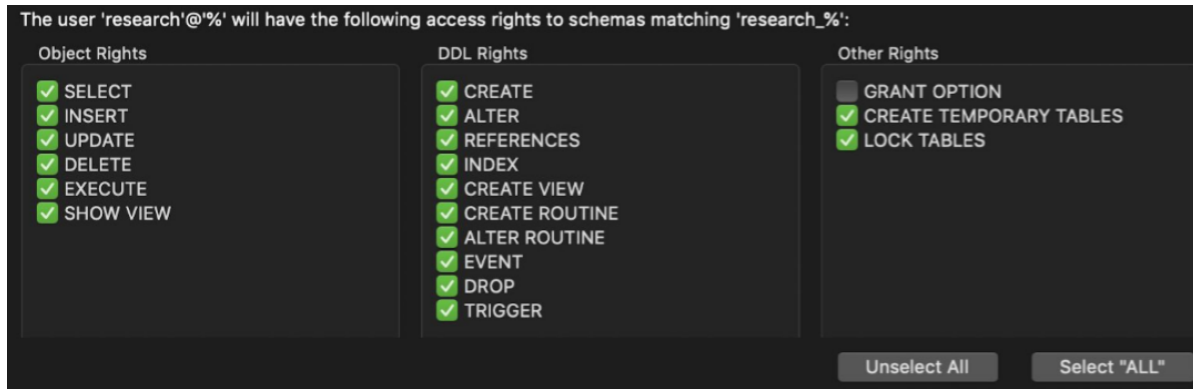
It is imperative that you back up the Flix databases, along with your **Asset** directories regularly to prevent data loss. Backups of the database can be completed using Flix Server and the **--db-backup** command. Backups can also be completed using MySQL directly for more advanced users.



Article: For more information on backing up your Flix database, take a look at this [Knowledge Base Article](#).

Flix installs all the relevant MySQL tables on startup, when required. You are prompted to backup your databases whenever Flix needs to make schema changes. Flix makes full use of database normalization to ensure data integrity and improved performance. We do not recommend making direct data changes to the database as this could cause unexpected data inconsistencies.

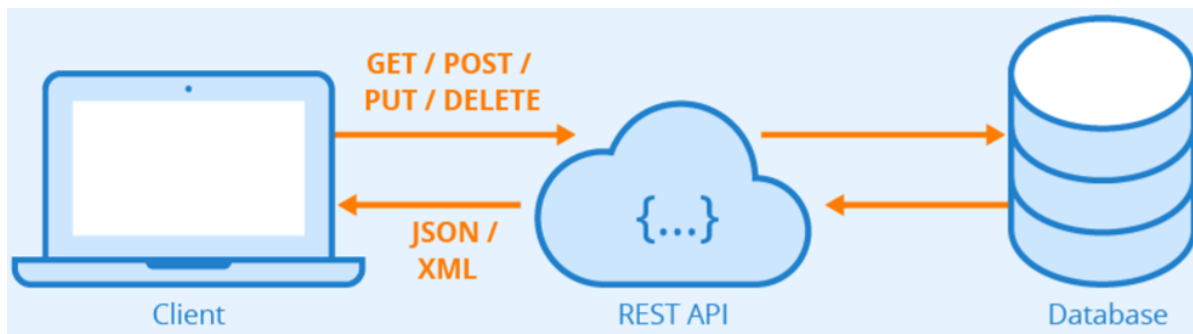
Flix performs a wide variety of actions on the database and requires an extended permission set. Flix checks your permissions on start-up to ensure it has the correct privileges.



API and Websockets

Flix Server provides a [RESTful API](#) over HTTP(S). This API manages the majority of data creation and retrieval on the server. The API not only allows for Flix Client to communicate with the server, but it also allows for a high degree of interoperability between other software, including your own custom scripts.

When a client makes a request to the server Flix Server ensures that the request is valid and then takes the appropriate action. For example, creating a show in Flix Client causes Flix Server to store the show information in the database. Once stored, the server creates a response message to the client to indicate that the operation was successful.



Flix also uses websockets for realtime communication between the client and server. We use websockets mostly for signalling purposes. For example, when the server has completed a long-running operation, it then notifies the client using a websocket message that it is complete, allowing the UI to be updated. This is a very performant way of providing realtime interactivity between the client and server. For Flix to work effectively, it needs to maintain a connection between the client and server. Flix notifies you if the connection drops at any time by displaying a **Reconnecting to Flix** popup.

Security

Flix is designed with security in mind to ensure we can commit to remote working capabilities. All communication between Flix Client and Flix Server is over HTTP(S), which is a widely accepted and well

understood communication protocol allowing us to take full advantage of the features it provides. Using HTTPS (SSL/TLS) encrypts communication packets between the client and server. This ensures that the data cannot be intercepted and read by a third-party in transit. This is especially important for remote working with Flix to prevent data leaks.

HTTPS requires you to create certificates for your servers using well known software such as OpenSSL or a third-party vendor who can generate certificates for you.

Flix also signs all HTTP requests between the client and server for authorization and security purposes. Flix uses private and public keys for each user to sign each request. This ensures that when the request reaches the server, it is guaranteed to be the same request that was sent from the client, eliminating the possibility of man-in-the-middle attacks.

See [Setting Up HTTPS](#) for more information on configuring Flix to run over HTTPS.

Licensing

To make it easy on artists, Flix Client is a non-licensed application, meaning anyone can download and launch it. Flix Server, however, requires a license to run, and handles the authentication of artists logging in using Flix Client.

For Flix deployments with a single server, the easiest licensing method is using a node-locked license, installed on the server/VM used to run Flix Server. For multi-server Flix deployments, we recommend using a server, or floating, license instead. This option shares a single license file between all the servers, rather than having a node-locked license on each Flix Server. Floating licenses can be installed on a dedicated License Server or on one of the servers/VMs running Flix Server in your deployment. Flix uses the same licensing mechanism as every other Foundry product. More detailed information on the Foundry's licensing can be found in our [Licensing Documentation](#).

Flix licenses control how many clients can be logged in to Flix concurrently. If you have a 10-seat Flix license, 10 clients can work in Flix at the same time, but an 11th client is denied access.

See [Licensing Flix Server](#) for more information.

Remote Access

Flix is designed for artists to be able to log in from anywhere, whether they're on premises or working remotely.

To achieve this, the server and port on which Flix Server is running must be reachable by the artist's Flix Client app publicly or over a virtual private network (VPN).

We recommend using a fully qualified domain (FQDN) for each Flix Server you set up for remote artists. The domain name must be resolvable by the Flix Client app for Flix to work.

Minimum Bandwidth Requirements for Flix

The table below shows a series of tests at different bandwidths and the corresponding results. We cannot guarantee that Flix will work if your internet speed is below 4 Mbps. For the optimal experience of Flix, we recommend using an internet speed of or above 16 Mbps.

Action	Filesize (MB)	Bandwidth (megabits/second)	Latency (ms)	Time Taken	Errors/Warnings	
Manual Import	245.7	Control: 60 gigabit/second	0	2:45	None	Maximum Bandwidth
Manual Import	245.7	0.25	0	>30 min	Error: chunk upload took too long, ~17 seconds	
Manual Import	245.7	0.5	0	>30 min	Warning: chunk upload was a little slow: ~9 seconds	
Manual Import	245.7	1	0	>30 min	Warning: chunk upload was a little slow: ~4.5 seconds	
Manual Import	245.7	2	0	25 min	Warning: chunk upload was a little slow: ~2 seconds	
Manual Import	245.7	2.5	0	22 min	None	Minimum Bandwidth
Manual Import	245.7	3	0	17:45	None	
Manual Import	245.7	4	0	13:40	None	
Manual Import	245.7	6	0	10:35	None	

Action	Filesize (MB)	Bandwidth (megabits/second)	Latency (ms)	Time Taken	Errors/Warnings	
Manual Import	245.7	8	0	8:05	None	
Manual Import	245.7	10	0	6:35	None	
Manual Import	245.7	15	0	5:35	None	
Manual Import	245.7	20	0	4:45	None	
Manual Import	245.7	30	0	3:55	None	

Flix Server Technical Overview

This technical overview aims to give Systems Administrators and Tech Teams deploying Flix in a studio environment some context as to what the requirements are, how it works under the hood, and to provide a high level understanding of Flix's overall architecture.

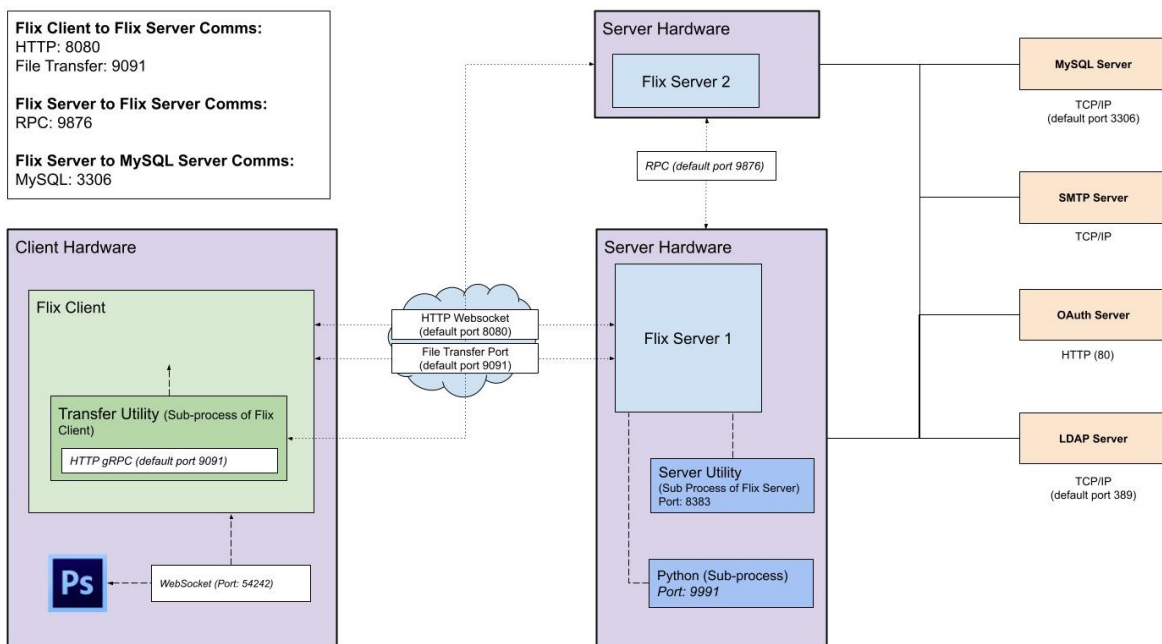
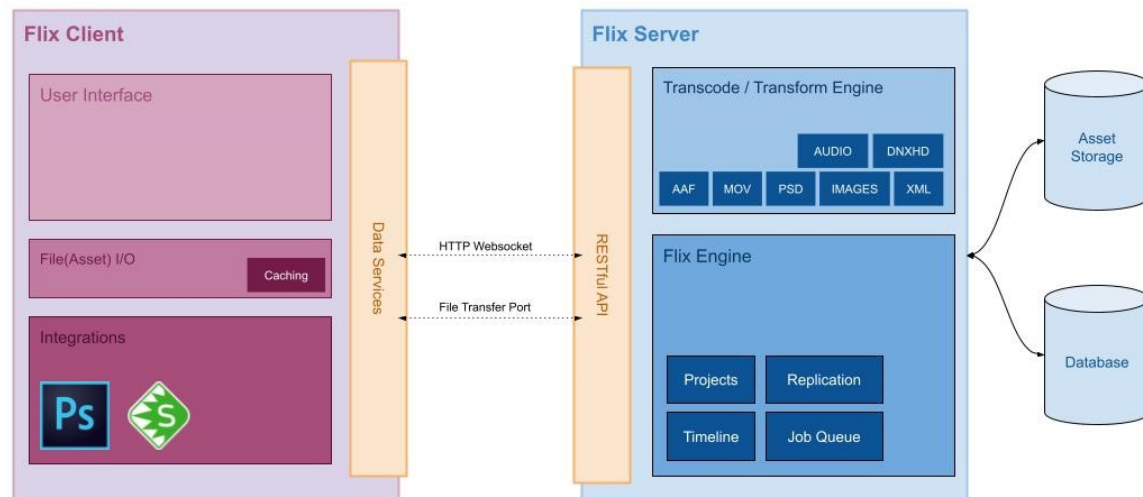
This overview assumes some familiarity with systems administration and client/server architecture.

Architecture

Flix is a client / server architecture application. The client (Flix Client) is the user interface for interacting with Flix, creating boards, viewing sequences, and interaction with third-party tools like Photoshop. The server side (Flix Server) is where the bulk of Flix's processing takes place. Flix Server provides a RESTful HTTP API for Flix Client, or custom scripts, to consume. Communication between Flix Client and Flix Server is done over HTTP(S) and Websockets.

The server utilizes a MySQL database for data persistence of shows, episodes, sequences, panels, dialogue, and so on. Although the database holds image data, such as panel thumbnails or artwork, it doesn't store the actual files or assets. These are stored in a separate asset location. Each Flix Server requires access to the same MySQL server to operate.

Flix Server handles all image transcoding, manipulation, and storage. Asset storage on the server can be local, or on a network share. Flix can be configured to use either option. Network shares must be presented to Flix as a file system directory.



Server Requirements

Flix Server can be installed on a physical server or virtual machine, and it's recommended to have multiple servers set up in your Flix deployment to provide scalability and redundancy. Flix spreads requests across servers equally, to ensure all servers resources are used to their full extent.

A single-server setup can be enough for a small deployment (3-5 users), but a multi-server setup is better suited to a larger user base, especially for workflows which involve greater involvement with Editorial. Servers can be added after the initial setup to scale up when needed.



Note: Flix handles its own load-balancing, so we do not recommend adding another load-balancer to your deployment, such as Varnish.

Hardware Specifications

The recommended hardware specs for Flix Server are highly dependent on usage. The main areas that require considerable resources are image processing and file transfers. Image processing in Flix is mostly handled on the server side, so the server requires resources to do these tasks. These tasks can be CPU intensive, especially the creation of DNxHD for Avid Media Composer. More powerful CPUs process images faster, resulting in faster editorial round-trips and panel creation.



Note: Flix Server does not require a GPU.

Flix Server architecture is built on the basis of concurrency and requires multiple CPU cores to operate effectively. Flix Server requires a CPU with a minimum of 2 cores, but we recommend 4 cores, or more, to allow Flix to multitask more effectively.

The number of tasks Flix Server can perform also depends on the amount of available RAM. More concurrent tasks may require more memory, so having enough RAM available is essential for good performance. A minimum of 4 GB of RAM is required, but we recommend 8 GB or 16 GB for larger installations.

Flix Server can be scaled in two different ways, horizontally and vertically. Horizontal scaling refers to the number of nodes, or servers, in your cluster. Vertical scaling refers to the amount of resources available to each of your nodes, such as CPU cores and RAM. Scaling in each direction provides a distinct advantage to your Flix installation:

Horizontal Scaling

 Increases API throughput and allows more Flix Client connections

- ⊕ Reduces load on each node in the cluster (if you have high resource utilization on your Flix Servers, you may want to scale horizontally)
- ⊕ Adds redundancy and prevents downtime if a server outage occurs
- ⊖ Adds pressure to your database server

Vertical Scaling

- ⊕ Improves rendering speed
- ⊕ Reduces time for panel creation
- ⊕ Improves DNxHD rendering speeds
- ⊕ Each server can handle more tasks concurrently

Storage

Flix Server stores and manages all of the assets imported in Flix as panels, audio files, AAFs, and so on, and therefore requires access to a file share with enough free storage. The storage requirements vary heavily depending on the size of the production, its duration, and the type of usage it sees from Flix Clients.

1 TB storage for a feature production is a good recommendation as a starting point, as long as this can be expanded on as needed.

Flix Server stores the assets on the machine it's running on by default, but we highly recommend pointing Flix Server to an external file share. The external file share can then be accessed by multiple Flix Servers to avoid data duplication and to centralize all of your assets. If you configure Flix Server to use a shared network mount, we recommend you use the 'shared storage' feature, to allow any Flix Server to serve assets. Using this feature decreases unnecessary data duplication in your assets directories.

Flix stores all assets in the **Asset** directory, which is configured on the server. This directory is managed by the Flix Servers, and should not be directly accessed by sysadmins or artists.



Warning: We strongly advise against renaming, deleting, or tampering with files from the **Asset** directory, as it may cause failures in Flix or result in missing media.

Flix supports any file system available on the operating system. Flix expects a mounted file system directory with full read and write permissions to function correctly. SMB or NFS are recommended as they are widely available, though NFS setups perform better in some instances.

Assets are partitioned by **Show** in the **Asset** directory for the purposes of archiving, if required. Once a show becomes old, and no longer in use, its assets can be moved out of the **Asset** directory for backup purposes to free up storage availability. However, any access to assets requested for those shows is no longer available and appears offline within Flix.

Database

Flix uses the relational database MySQL for data persistence. MySQL has many features making data storage efficient and scalable and it is a tried and tested database server, ensuring we have a reliable data storage layer.



Warning: Flix MySQL server now requires the **innodb_autoinc_lock_mode** global variable to be set to 1, consecutive mode. If MySQL is configured to use interleaved mode, the server will not start.

All persistent data in Flix is stored in the database, except for the assets, which are stored on the file system separately. However, metadata for the assets, including location and naming, is stored in the database. This ensures we don't have to access the file system when querying asset data, which can be incredibly fast if the data is in the database's caches. This way, Flix doesn't rely on continuous reads that can put a lot of pressure on file system storage.



Article: If you want to learn more about what info is stored in the database and how this is connected to the assets and Flix, take a look at this [Knowledge Base Article](#).

It is imperative that you back up the Flix databases, along with your **Asset** directories regularly to prevent data loss. Backups of the database can be completed using Flix Server and the **--db-backup** command. Backups can also be completed using MySQL directly for more advanced users.

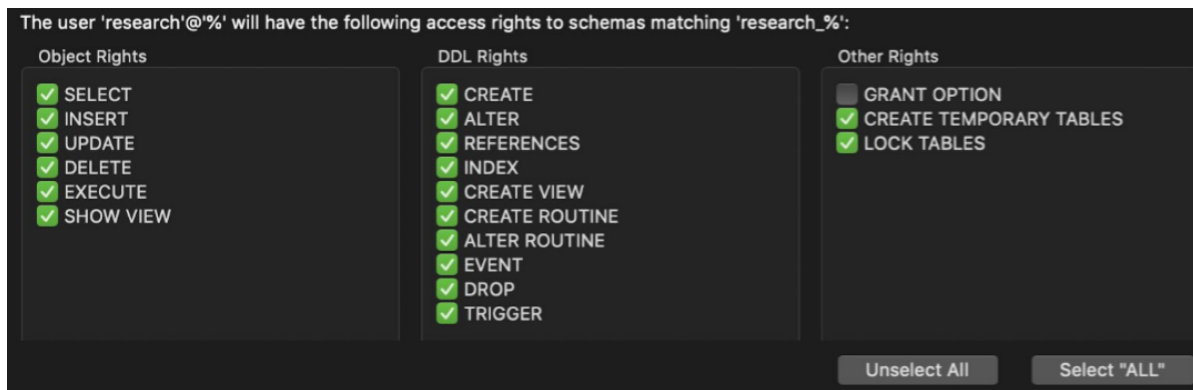


Article: For more information on backing up your Flix database, take a look at this [Knowledge Base Article](#).

Flix installs all the relevant MySQL tables on startup, when required. You are prompted to backup your databases whenever Flix needs to make schema changes. Flix makes full use of database normalization to

ensure data integrity and improved performance. We do not recommend making direct data changes to the database as this could cause unexpected data inconsistencies.

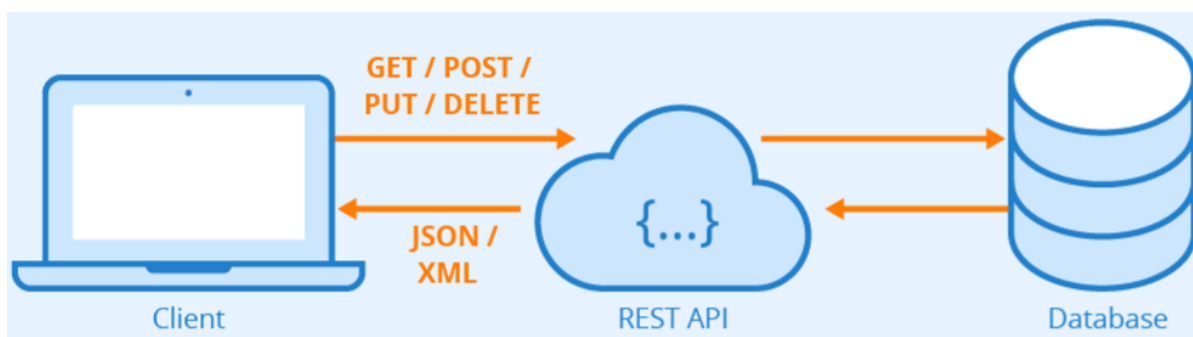
Flix performs a wide variety of actions on the database and requires an extended permission set. Flix checks your permissions on start-up to ensure it has the correct privileges.



API and Websockets

Flix Server provides a [RESTful API](#) over HTTP(S). This API manages the majority of data creation and retrieval on the server. The API not only allows for Flix Client to communicate with the server, but it also allows for a high degree of interoperability between other software, including your own custom scripts.

When a client makes a request to the server Flix Server ensures that the request is valid and then takes the appropriate action. For example, creating a show in Flix Client causes Flix Server to store the show information in the database. Once stored, the server creates a response message to the client to indicate that the operation was successful.



Flix also uses websockets for realtime communication between the client and server. We use websockets mostly for signalling purposes. For example, when the server has completed a long-running operation, it then notifies the client using a websocket message that it is complete, allowing the UI to be updated. This is a very performant way of providing realtime interactivity between the client and server. For Flix to work effectively, it needs to maintain a connection between the client and server. Flix notifies you if the connection drops at any time by displaying a **Reconnecting to Flix** popup.

Security

Flix is designed with security in mind to ensure we can commit to remote working capabilities. All communication between Flix Client and Flix Server is over HTTP(S), which is a widely accepted and well understood communication protocol allowing us to take full advantage of the features it provides. Using HTTPS (SSL/TLS) encrypts communication packets between the client and server. This ensures that the data cannot be intercepted and read by a third-party in transit. This is especially important for remote working with Flix to prevent data leaks.

HTTPS requires you to create certificates for your servers using well known software such as OpenSSL or a third-party vendor who can generate certificates for you.

Flix also signs all HTTP requests between the client and server for authorization and security purposes. Flix uses private and public keys for each user to sign each request. This ensures that when the request reaches the server, it is guaranteed to be the same request that was sent from the client, eliminating the possibility of man-in-the-middle attacks.

See [Setting Up HTTPS](#) for more information on configuring Flix to run over HTTPS.

Licensing

To make it easy on artists, Flix Client is a non-licensed application, meaning anyone can download and launch it. Flix Server, however, requires a license to run, and handles the authentication of artists logging in using Flix Client.

For Flix deployments with a single server, the easiest licensing method is using a node-locked license, installed on the server/VM used to run Flix Server. For multi-server Flix deployments, we recommend using a server, or floating, license instead. This option shares a single license file between all the servers, rather than having a node-locked license on each Flix Server. Floating licenses can be installed on a dedicated License Server or on one of the servers/VMs running Flix Server in your deployment. Flix uses the same licensing mechanism as every other Foundry product. More detailed information on the Foundry's licensing can be found in our [Licensing Documentation](#).

Flix licenses control how many clients can be logged in to Flix concurrently. If you have a 10-seat Flix license, 10 clients can work in Flix at the same time, but an 11th client is denied access.

See [Licensing Flix Server](#) for more information.

Remote Access

Flix is designed for artists to be able to log in from anywhere, whether they're on premises or working remotely.

To achieve this, the server and port on which Flix Server is running must be reachable by the artist's Flix Client app publicly or over a virtual private network (VPN).

We recommend using a fully qualified domain (FQDN) for each Flix Server you set up for remote artists. The domain name must be resolvable by the Flix Client app for Flix to work.

Minimum Bandwidth Requirements for Flix

The table below shows a series of tests at different bandwidths and the corresponding results. We cannot guarantee that Flix will work if your internet speed is below 4 Mbps. For the optimal experience of Flix, we recommend using an internet speed of or above 16 Mbps.

Action	Filesize (MB)	Bandwidth (megabits/second)	Latency (ms)	Time Taken	Errors/Warnings	
Manual Import	245.7	Control: 60 gigabit/second	0	2:45	None	Maximum Bandwidth
Manual Import	245.7	0.25	0	>30 min	Error: chunk upload took too long, ~17 seconds	
Manual Import	245.7	0.5	0	>30 min	Warning: chunk upload was a little slow: ~9 seconds	
Manual Import	245.7	1	0	>30 min	Warning: chunk upload was a little slow: ~4.5 seconds	
Manual Import	245.7	2	0	25 min	Warning: chunk upload was a little slow: ~2 seconds	
Manual Import	245.7	2.5	0	22 min	None	Minimum Bandwidth

Action	Filesize (MB)	Bandwidth (megabits/second)	Latency (ms)	Time Taken	Errors/Warnings	
Manual Import	245.7	3	0	17:45	None	
Manual Import	245.7	4	0	13:40	None	
Manual Import	245.7	6	0	10:35	None	
Manual Import	245.7	8	0	8:05	None	
Manual Import	245.7	10	0	6:35	None	
Manual Import	245.7	15	0	5:35	None	
Manual Import	245.7	20	0	4:45	None	
Manual Import	245.7	30	0	3:55	None	

Flix Server and Client API Reference

Flix has extensive API support for custom tools and pipeline integration.

Flix's open, flexible APIs give you the freedom to integrate Flix with your pipeline and create custom tools.



[Flix Server](#)

A reference guide to Flix Server's API.



[Flix Client](#)

A reference guide to Flix Client's API.

Supported File Formats

The following tables list file formats and codec support in Flix and compatible third-party applications.

Panel Formats

Stills

Format	Extension
JPEG	.jpeg, .jpg
PNG	.png
PSD	.psd
OGG	.ogg
TIFF	.tif, .tiff

Audio Formats

Format	Extension
WAV	.wav
MP3	.mp3

Flix and Storyboard Pro 20 Container Formats

For more information on format support in Storyboard Pro 20, see [About Video Clip Format](#).

OS	Container	Codec
Windows	.mov	H.264 Motion JPEG MPEG-4
macOS	.mov	H.264 Apple ProRes

Flix and Adobe Premiere Pro Container Formats

For more information on format support in Adobe Premiere Pro, see [Supported File Formats](#).

OS	Container	Codec
Windows	.mov	M-JPEG MPEG-1 MPEG-2 MPEG-3 MPEG-4 H.264 DVC Pro 50 Apple ProRes DNxHD
macOS	.mov	M-JPEG MPEG-2 H.264

OS	Container	Codec
		DVC Pro 50
		Apple ProRes
		DNxHD

Flix and Avid Media Composer Container Formats

For more information on format support in Avid Media Composer, see [Avid Supported Video File Formats](#).



Note: From Media Composer version 2020.4 onwards, DNxHD resolution names have been simplified:

OS	Container	Codec	In Flix	In Avid Media Composer
Windows	.mov	M-JPEG	DNxHD 36, 45	DNxHD LB
		MPEG-1	DNxHD 115, 120, and 145	DNxHD SQ
		MPEG-2		
		MPEG-3		
		MPEG-4		
		H.264		
		DVC Pro 50		
		Apple ProRes		
macOS	.mov	DNxHD		
		M-JPEG	DNxHD 36, 45	DNxHD LB
		MPEG-2	DNxHD 115, 120, and 145	DNxHD SQ
		H.264		

OS	Container	Codec	In Flix	In Avid Media Composer
		Apple ProRes		
		DNxHD		

Third-Party Application Support

This page lists the supported applications used with different versions of Flix.



Note: Other versions of third-party software may work, but they haven't been fully tested with the listed version of Flix, so they cannot be officially supported.
If you have any problems with a particular application's version, please contact the Foundry Support Team, refer to [Contacting Customer Support](#).

Flix 8.0 Series

Applications

Application	Supported Versions
Adobe Photoshop	CC 2024, 2025
Adobe Premiere	CC 2024, 2025
Toon Boom Storyboard Pro	24, 25
Avid Media Composer	2023.3, 2024

Flix

Item	OS	Supported Versions
Flix Server	Linux	Rocky 9
Flix Client	Windows OS	Windows 11
Flix Client	macOS	15.x (Sequoia) 16.x (Tahoe)
Database	MySQL	8

Flix 7.1 Series

Applications

Application	Supported Versions
Adobe Photoshop	CC 2023, 2024
Adobe Premiere	CC 2023, 2024
Toon Boom Storyboard Pro	20, 22, 24
Avid Media Composer	2022.12, 2023.3

Flix

Item	OS	Supported Versions
Flix Server	Linux	CentOS 7, Rocky 9
Flix Client	Windows OS	Windows 10 Windows 11
Flix Client	macOS	13.x (Ventura) 14.x (Sonoma) 15.x (Sequoia)
Database	MySQL	8

Flix 7.0 Series

Applications

Application	Supported Versions
Adobe Photoshop	CC 2023, 2024
Adobe Premiere	CC 2023, 2024
Toon Boom Storyboard Pro	20, 22
Avid Media Composer	2022.12, 2023.3

Flix

Item	OS	Supported Versions
Flix Server	Linux	CentOS 7, Rocky 9
Flix Client	Windows OS	Windows 10 Windows 11
Flix Client	macOS	13.x (Ventura) 14.x (Sonoma) 15.x (Sequoia)
Database	MySQL	8, 5.7

Flix 6.6 Series

Applications



Note: Photoshop 2022 support requires some extra set up before you install the plug-in, see [Flix for Story](#) for more information.

Application	Supported Versions
Adobe Photoshop	CC 2021, 2022
Adobe Premiere	CC 2021, 2022
Toon Boom Storyboard Pro	2020 (20.10.2, or later) and 2022
Avid Media Composer	2020, 2021

Flix

Item	OS	Supported Versions
Flix Server	Linux	CentOS 7
Flix Client	Windows OS	Windows 10 Windows 11
Flix Client	macOS	11.x (Big Sur) 12.x (Monterey) 13.x (Ventura)

Flix 6.5 Series

Applications



Note: Photoshop 2022 support requires some extra set up before you install the plug-in, see [Flix for Story](#) for more information.

Application	Supported Versions
Adobe Photoshop	CC 2021, 2022
Adobe Premiere	CC 2021, 2022
Toon Boom Storyboard Pro	Storyboard Pro 2020 (20.10.2, or later)
Avid Media Composer	2020, 2021

Flix

Item	OS	Supported Versions
Flix Server	Linux	CentOS 7
Flix Client	Windows OS	Windows 10 Windows 11
Flix Client	macOS	11.x (Big Sur) 12.x (Monterey)

Flix 6.4 Series



Note: From Flix 6.5.0 onwards, Photoshop CC 2020 and Avid Media Composer 2018 will no longer be supported.



Note: Photoshop 2022 support requires some extra set up before you install the plug-in, see [Flix for Story](#) for more information.

Applications

Application	Supported Versions
Adobe Photoshop	CC 2020, 2021, 2022
Adobe Premiere	CC 2020, 2021
Toon Boom Storyboard Pro	7.0.2 (version 17.10.2, or later), 20 (20.10.2, or later)
Avid Media Composer	2019, 2020

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.15.x (Catalina) 11.x (Big Sur) 12.x (Monterey)
Linux	CentOS 7

Flix 6.3.7

Applications

Application	Supported Versions
Adobe Photoshop	CC 2019, 2020, 2021
Adobe Premiere	CC 2019, 2020
Toon Boom Storyboard Pro	7.0.2 (version 17.10.2, or later), 20 (20.10.2, or later)
Avid Media Composer	2019, 2020

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.14 (Mojave) 10.15.x (Catalina) 11.x (Big Sur)
Linux	CentOS 7

Flix 6.3.6

Applications

Application	Supported Versions
Adobe Photoshop	CC 2019, 2020
Adobe Premiere	CC 2019, 2020
Toon Boom Storyboard Pro	7.0.2 (version 17.10.2, or later)
Avid Media Composer	2019, 2020

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.14 (Mojave) 10.15.x (Catalina)
Linux	CentOS 6 CentOS 7

Flix 6.3.5

Applications

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6
Avid Media Composer	8.10, 2018

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.13 (High Sierra) 10.14 (Mojave)
Linux	CentOS 6 CentOS 7

Flix 6.3.4

Applications

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6
Avid Media Composer	8.10, 2018

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.13 (High Sierra) 10.14 (Mojave)
Linux	CentOS 6 CentOS 7

Flix 6.3.3

Applications

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6
Avid Media Composer	8.9, 8.10, 2018

Flix

Operating System	Supported Versions
Windows OS	10
macOS	10.12 (Sierra)

Operating System	Supported Versions
Linux	10.13 (High Sierra)
	10.14 (Mojave)
	CentOS 6
	CentOS 7

Flix 6.3.2

Applications

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6
Avid Media Composer	8.9, 8.10, 2018

Flix

Operating System	Supported Versions
Windows	10
macOS	10.12 (Sierra)
	10.13 (High Sierra)

Operating System	Supported Versions
	10.14 (Mojave)
Linux	CentOS 6
	CentOS 7

Third-Party Software Notices

This page lists third-party packages and versions used in Flix, along with their respective licenses.

Flix Server

Name	Version	License
cloud.google.com/go	v0.110.0	Apache-2.0
cloud.google.com/go/accessapproval	v1.6.0	Apache-2.0
cloud.google.com/go/accesscontextmanager	v1.7.0	Apache-2.0
cloud.google.com/go/aiplatform	v1.37.0	Apache-2.0
cloud.google.com/go/analytics	v0.19.0	Apache-2.0
cloud.google.com/go/apigateway	v1.5.0	Apache-2.0
cloud.google.com/go/apigeeconnect	v1.5.0	Apache-2.0
cloud.google.com/go/apigeeregistry	v0.6.0	Apache-2.0
cloud.google.com/go/apikeys	v0.6.0	Apache-2.0
cloud.google.com/go/appengine	v1.7.1	Apache-2.0
cloud.google.com/go/area120	v0.7.1	Apache-2.0
cloud.google.com/go/artifactregistry	v1.13.0	Apache-2.0
cloud.google.com/go/asset	v1.13.0	Apache-2.0

Name	Version	License
cloud.google.com/go/assuredworkloads	v1.10.0	Apache-2.0
cloud.google.com/go/automl	v1.12.0	Apache-2.0
cloud.google.com/go/baremetalsolution	v0.5.0	Apache-2.0
cloud.google.com/go/batch	v0.7.0	Apache-2.0
cloud.google.com/go/beyondcorp	v0.5.0	Apache-2.0
cloud.google.com/go/bigquery	v1.50.0	Apache-2.0
cloud.google.com/go/billing	v1.13.0	Apache-2.0
cloud.google.com/go/binaryauthorization	v1.5.0	Apache-2.0
cloud.google.com/go/certificatemanager	v1.6.0	Apache-2.0
cloud.google.com/go/channel	v1.12.0	Apache-2.0
cloud.google.com/go/cloudbuild	v1.9.0	Apache-2.0
cloud.google.com/go/clouddms	v1.5.0	Apache-2.0
cloud.google.com/go/cloudtasks	v1.10.0	Apache-2.0
cloud.google.com/go/compute	v1.19.3	Apache-2.0
cloud.google.com/go/compute/metadata	v0.2.3	Apache-2.0
cloud.google.com/go/contactcenterinsights	v1.6.0	Apache-2.0
cloud.google.com/go/container	v1.15.0	Apache-2.0
cloud.google.com/go/containeranalysis	v0.9.0	Apache-2.0
cloud.google.com/go/datacatalog	v1.13.0	Apache-2.0
cloud.google.com/go/dataflow	v0.8.0	Apache-2.0
cloud.google.com/go/dataform	v0.7.0	Apache-2.0
cloud.google.com/go/datafusion	v1.6.0	Apache-2.0
cloud.google.com/go/datalabeling	v0.7.0	Apache-2.0

Name	Version	License
cloud.google.com/go/dataplex	v1.6.0	Apache-2.0
cloud.google.com/go/dataproc	v1.12.0	Apache-2.0
cloud.google.com/go/dataqna	v0.7.0	Apache-2.0
cloud.google.com/go/datastore	v1.11.0	Apache-2.0
cloud.google.com/go/datastream	v1.7.0	Apache-2.0
cloud.google.com/go/deploy	v1.8.0	Apache-2.0
cloud.google.com/go/dialogflow	v1.32.0	Apache-2.0
cloud.google.com/go/dlp	v1.9.0	Apache-2.0
cloud.google.com/go/documentai	v1.18.0	Apache-2.0
cloud.google.com/go/domains	v0.8.0	Apache-2.0
cloud.google.com/go/edgecontainer	v1.0.0	Apache-2.0
cloud.google.com/go/errorreporting	v0.3.0	Apache-2.0
cloud.google.com/go/essentialcontacts	v1.5.0	Apache-2.0
cloud.google.com/go/eventarc	v1.11.0	Apache-2.0
cloud.google.com/go/filestore	v1.6.0	Apache-2.0
cloud.google.com/go/firestore	v1.9.0	Apache-2.0
cloud.google.com/go/functions	v1.13.0	Apache-2.0
cloud.google.com/go/gaming	v1.9.0	Apache-2.0
cloud.google.com/go/gkebackup	v0.4.0	Apache-2.0
cloud.google.com/go/gkeconnect	v0.7.0	Apache-2.0
cloud.google.com/go/gkehub	v0.12.0	Apache-2.0
cloud.google.com/go/gkemulticloud	v0.5.0	Apache-2.0
cloud.google.com/go/gsuiteaddons	v1.5.0	Apache-2.0

Name	Version	License
cloud.google.com/go/iam	v0.13.0	Apache-2.0
cloud.google.com/go/iap	v1.7.1	Apache-2.0
cloud.google.com/go/ids	v1.3.0	Apache-2.0
cloud.google.com/go/iot	v1.6.0	Apache-2.0
cloud.google.com/go/kms	v1.10.1	Apache-2.0
cloud.google.com/go/language	v1.9.0	Apache-2.0
cloud.google.com/go/lifesciences	v0.8.0	Apache-2.0
cloud.google.com/go/logging	v1.7.0	Apache-2.0
cloud.google.com/go/longrunning	v0.4.1	Apache-2.0
cloud.google.com/go/managedidentities	v1.5.0	Apache-2.0
cloud.google.com/go/maps	v0.7.0	Apache-2.0
cloud.google.com/go/mediatranslation	v0.7.0	Apache-2.0
cloud.google.com/go/memcache	v1.9.0	Apache-2.0
cloud.google.com/go/metastore	v1.10.0	Apache-2.0
cloud.google.com/go/monitoring	v1.13.0	Apache-2.0
cloud.google.com/go/networkconnectivity	v1.11.0	Apache-2.0
cloud.google.com/go/networkmanagement	v1.6.0	Apache-2.0
cloud.google.com/go/networksecurity	v0.8.0	Apache-2.0
cloud.google.com/go/notebooks	v1.8.0	Apache-2.0
cloud.google.com/go/optimization	v1.3.1	Apache-2.0
cloud.google.com/go/orchestration	v1.6.0	Apache-2.0
cloud.google.com/go/orgpolicy	v1.10.0	Apache-2.0
cloud.google.com/go/osconfig	v1.11.0	Apache-2.0

Name	Version	License
cloud.google.com/go/oslogin	v1.9.0	Apache-2.0
cloud.google.com/go/phishingprotection	v0.7.0	Apache-2.0
cloud.google.com/go/policytroubleshooter	v1.6.0	Apache-2.0
cloud.google.com/go/privatecatalog	v0.8.0	Apache-2.0
cloud.google.com/go/pubsub	v1.30.0	Apache-2.0
cloud.google.com/go/pubsublite	v1.7.0	Apache-2.0
cloud.google.com/go/recaptchaenterprise/v2	v2.7.0	Apache-2.0
cloud.google.com/go/recommendationengine	v0.7.0	Apache-2.0
cloud.google.com/go/recommender	v1.9.0	Apache-2.0
cloud.google.com/go/redis	v1.11.0	Apache-2.0
cloud.google.com/go/resourcemanager	v1.7.0	Apache-2.0
cloud.google.com/go/resourcesettings	v1.5.0	Apache-2.0
cloud.google.com/go/retail	v1.12.0	Apache-2.0
cloud.google.com/go/run	v0.9.0	Apache-2.0
cloud.google.com/go/scheduler	v1.9.0	Apache-2.0
cloud.google.com/go/secretmanager	v1.10.0	Apache-2.0
cloud.google.com/go/security	v1.13.0	Apache-2.0
cloud.google.com/go/securitycenter	v1.19.0	Apache-2.0
cloud.google.com/go/servicecontrol	v1.11.1	Apache-2.0
cloud.google.com/go/servicedirectory	v1.9.0	Apache-2.0
cloud.google.com/go/servicemanagement	v1.8.0	Apache-2.0
cloud.google.com/go/serviceusage	v1.6.0	Apache-2.0
cloud.google.com/go/shell	v1.6.0	Apache-2.0

Name	Version	License
cloud.google.com/go/spanner	v1.45.0	Apache-2.0
cloud.google.com/go/speech	v1.15.0	Apache-2.0
cloud.google.com/go/storagetransfer	v1.8.0	Apache-2.0
cloud.google.com/go/talent	v1.5.0	Apache-2.0
cloud.google.com/go/texttospeech	v1.6.0	Apache-2.0
cloud.google.com/go/tpu	v1.5.0	Apache-2.0
cloud.google.com/go/trace	v1.9.0	Apache-2.0
cloud.google.com/go/translate	v1.7.0	Apache-2.0
cloud.google.com/go/video	v1.15.0	Apache-2.0
cloud.google.com/go/videointelligence	v1.10.0	Apache-2.0
cloud.google.com/go/vision/v2	v2.7.0	Apache-2.0
cloud.google.com/go/vmmigration	v1.6.0	Apache-2.0
cloud.google.com/go/vmwareengine	v0.3.0	Apache-2.0
cloud.google.com/go/vpcaccess	v1.6.0	Apache-2.0
cloud.google.com/go/webrisk	v1.8.0	Apache-2.0
cloud.google.com/go/websecurityscanner	v1.5.0	Apache-2.0
cloud.google.com/go/workflows	v1.10.0	Apache-2.0
fyne.io/fyne/v2	v2.2.1	BSD-3- Clause
fyne.io/systray	v1.10.0	Apache-2.0
gioui.org	v0.0.0-20230506155350- febadd314531	MIT
gioui.org/cpu	v0.0.0-20220412190645- f1e9e8c3b1f7	MIT

Name	Version	License
gioui.org/shader	v1.0.6	MIT
gioui.org/x	v0.0.0-20230426160849-752f112c7a59	MIT
git.sr.ht/~sbinet/gg	v0.4.1	MIT
github.com/BurntSushi/toml	v1.2.1	MIT
github.com/ByteArena/poly2tri-go	v0.0.0-20170716161910-d102ad91854f	BSD-3-Clause
github.com/CloudyKit/fastprinter	v0.0.0-20200109182630-33d98a066a53	MIT
github.com/CloudyKit/jet/v6	v6.2.0	Apache-2.0
github.com/DATA-DOG/go-sqlmock	v1.5.0	BSD-3-Clause
github.com/DATA-DOG/godog	v0.7.13	BSD-3-Clause
github.com/Joker/jade	v1.1.3	BSD-3-Clause
github.com/Masterminds/squirrel	v1.5.4	MIT
github.com/Shopify/goreferrer	v0.0.0-20220729165902-8cddb4f5de06	MIT
github.com/TheFoundryVisionmongers/canvas	v0.0.0-20221117083838-abfb2b0570fd	MIT
github.com/VividCortex/mysqlerr	v1.0.0	MIT
github.com/adrg/strutil	v0.3.0	MIT
github.com/adrg/sysfont	v0.1.2	MIT
github.com/adrg/xdg	v0.4.0	MIT
github.com/ajstarks/svgo	v0.0.0-20211024235047-	CC-BY-4.0

Name	Version	License
	1546f124cd8b	
github.com/andybalholm/brotli	v1.0.5	MIT
github.com/andybalholm/stroke	v0.0.0-20221221101821-bd29b49d73f0	MIT
github.com/antihax/optional	v1.0.0	MIT
github.com/arvidfm/jsonschema	v0.6.0-1	MIT
github.com/aymerick/douceur	v0.2.0	MIT
github.com/beevik/etree	v1.1.0	BSD-2-Clause
github.com/benoitkugler/pstokenizer	v1.0.0	MIT
github.com/benoitkugler/textlayout	v0.3.0	MIT
github.com/benoitkugler/textlayout-testdata	v0.1.1	MIT
github.com/blang/semver	v3.5.1	MIT
github.com/census-instrumentation/opencensus-proto	v0.4.1	Apache-2.0
github.com/cespare/xxhash/v2	v2.2.0	MIT
github.com/cheekybits/is	v0.0.0-20150225183255-68e9c0620927	MIT
github.com/client9/misspell	v0.3.4	MIT
github.com/cncf/udpa/go	v0.0.0-20220112060539-c52dc94e7fbe	Apache-2.0
github.com/cncf/xds/go	v0.0.0-20230310173818-32f1caf87195	Apache-2.0
github.com/codegangsta/inject	v0.0.0-20150114235600-33e0aa1cb7c0	MIT
github.com/coreos/go-oidc/v3	v3.6.0	Apache-2.0

Name	Version	License
github.com/corona10/goimagehash	v1.1.0	BSD-2-Clause
github.com/creack/pty	v1.1.9	MIT
github.com/crewjam/httperr	v0.2.0	BSD-2-Clause
github.com/crewjam/saml	v0.4.13	BSD-2-Clause
github.com/davecgh/go-spew	v1.1.1	ISC
github.com/dchest/uniuri	v1.2.0	CC0-1.0
github.com/disintegration/imaging	v1.6.2	MIT
github.com/djherbis/atime	v1.1.0	MIT
github.com/dsnet/compress	v0.0.1	BSD-3-Clause
github.com/dsnet/golib	v0.0.0-20171103203638-1ea166775780	BSD-3-Clause
github.com/dustin/go-humanize	v1.0.1	MIT
github.com/eknkc/amber	v0.0.0-20171010120322-cdade1c07385	MIT
github.com/envoyproxy/go-control-plane	v0.11.0	Apache-2.0
github.com/envoyproxy/protoc-gen-validate	v0.10.0	Apache-2.0
github.com/fatih/structs	v1.1.0	MIT
github.com/felixge/httpsnoop	v1.0.3	MIT
github.com/flopp/go-findfont	v0.1.0	MIT
github.com/flosch/pongo2/v4	v4.0.2	MIT
github.com/fogleman/gg	v1.3.0	MIT

Name	Version	License
github.com/fredbi/uri	v0.0.0-20181227131451-3dcfdacbaaf3	MIT
github.com/fsnotify/fsnotify	v1.6.0	BSD-3-Clause
github.com/fyne-io/gl-js	v0.0.0-20220516203408-b35fbccb7063	BSD-3-Clause
github.com/fyne-io/glfw-js	v0.0.0-20220120001248-ee7290d23504	MIT
github.com/fyne-io/image	v0.0.0-20220602074514-4956b0afb3d2	BSD-3-Clause
github.com/getsentry/sentry-go	v0.21.0	MIT
github.com/ghodss/yaml	v1.0.0	MIT
github.com/gin-contrib/sse	v0.1.0	MIT
github.com/gin-gonic/gin	v1.8.1	MIT
github.com/gioui/uax	v0.2.1-0.20220325163150-e3d987515a12	BSD-3-Clause
github.com/go-errors/errors	v1.4.2	MIT
github.com/go-fonts/dejavu	v0.1.0	BSD-3-Clause
github.com/go-fonts/latin-modern	v0.3.1	BSD-3-Clause
github.com/go-fonts/liberation	v0.3.1	BSD-3-Clause
github.com/go-fonts/stix	v0.1.0	BSD-3-Clause
github.com/go-gl/gl	v0.0.0-20211210172815-726fda9656d6	MIT
github.com/go-gl/glfw/v3.3/glfw	v0.0.0-20220622232848-	BSD-3-

Name	Version	License
	a6c407ee30a0	Clause
github.com/go-jose/go-jose/v3	v3.0.0	Apache-2.0
github.com/go-latex/latex	v0.0.0-20230307184459-12ec69307ad9	BSD-3-Clause
github.com/go-martini/martini	v0.0.0-20170121215854-22fa46961aab	MIT
github.com/go-pdf/fpdf	v0.8.0	MIT
github.com/go-playground/locales	v0.14.0	MIT
github.com/go-playground/universal-translator	v0.18.0	MIT
github.com/go-playground/validator/v10	v10.11.1	MIT
github.com/go-sql-driver/mysql	v1.7.1	MPL-2.0
github.com/go-text/typesetting	v0.0.0-20230502123426-87572f5551cf	BSD-3-Clause
github.com/goccy/go-json	v0.9.11	MIT
github.com/godbus/dbus/v5	v5.1.0	BSD-2-Clause
github.com/gojuno/minimock/v3	v3.0.10	MIT
github.com/goki/freetype	v0.0.0-20220119013949-7a161fd3728c	FTL
github.com/golang-jwt/jwt/v4	v4.4.3	MIT
github.com/golang/freetype	v0.0.0-20170609003504-e2365dfdc4a0	FTL
github.com/golang/glog	v1.1.0	Apache-2.0
github.com/golang/groupcache	v0.0.0-20210331224755-41bb18bfe9da	Apache-2.0

Name	Version	License
github.com/golang/mock	v1.1.1	Apache-2.0
github.com/golang/protobuf	v1.5.3	BSD-3-Clause
github.com/golang/snappy	v0.0.4	BSD-3-Clause
github.com/google/go-cmp	v0.5.9	BSD-3-Clause
github.com/google/gofuzz	v1.2.0	Apache-2.0
github.com/google/s2a-go	v0.1.4	Apache-2.0
github.com/google/uuid	v1.3.0	BSD-3-Clause
github.com/googleapis/enterprise-certificate-proxy	v0.2.3	Apache-2.0
github.com/googleapis/gax-go/v2	v2.8.0	BSD-3-Clause
github.com/gopherjs/gopherjs	v1.17.2	BSD-2-Clause
github.com/gorilla/css	v1.0.0	BSD-3-Clause
github.com/gorilla/handlers	v1.5.1	BSD-2-Clause
github.com/gorilla/mux	v1.8.0	BSD-3-Clause
github.com/gorilla/websocket	v1.5.0	BSD-2-Clause
github.com/grpc-ecosystem/grpc-gateway	v1.16.0	BSD-3-Clause

Name	Version	License
github.com/hpcloud/tail	v1.0.0	MIT
github.com/iancoleman/orderedmap	v0.0.0-20190318233801-ac98e3ecb4b0	MIT
github.com/inconshreveable/mousetrap	v1.0.0	Apache-2.0
github.com/iris-contrib/schema	v0.0.6	BSD-3-Clause
github.com/jmoiron/sqlx	v1.3.5	MIT
github.com/jonboulle/clockwork	v0.2.2	Apache-2.0
github.com/josharian/intern	v1.0.0	MIT
github.com/json-iterator/go	v1.1.12	MIT
github.com/jsummers/gobmp	v0.0.0-20151104160322-e2ba15ffa76e	MIT
github.com/kataras/blocks	v0.0.7	MIT
github.com/kataras/golog	v0.1.8	BSD-3-Clause
github.com/kataras/iris/v12	v12.2.0	BSD-3-Clause
github.com/kataras/pio	v0.0.11	BSD-3-Clause
github.com/kataras/sitemap	v0.0.6	MIT
github.com/kataras/tunnel	v0.0.4	MIT
github.com/klauspost/compress	v1.16.0	Apache-2.0
github.com/klauspost/cpuid	v1.2.0	MIT
github.com/kr/pretty	v0.3.1	MIT
github.com/kr/pty	v1.1.1	MIT

Name	Version	License
github.com/kr/text	v0.2.0	MIT
github.com/labstack/echo/v4	v4.10.0	MIT
github.com/labstack/gommon	v0.4.0	MIT
github.com/lann/builder	v0.0.0-20180802200727-47ae307949d0	MIT
github.com/lann/ps	v0.0.0-20150810152359-62de8c46ede0	MIT
github.com/leodido/go-urn	v1.2.1	MIT
github.com/mailgun/raymond/v2	v2.0.48	MIT
github.com/mailru/easyjson	v0.7.7	MIT
github.com/matryer/try	v0.0.0-20161228173917-9ac251b645a2	MIT
github.com/mattermost/xml-roundtrip-validator	v0.1.0	Apache-2.0
github.com/mattn/go-colorable	v0.1.13	MIT
github.com/mattn/go-isatty	v0.0.17	MIT
github.com/microcosm-cc/bluemonday	v1.0.23	BSD-3-Clause
github.com/mitchellh/go-homedir	v1.1.0	MIT
github.com/modern-go/concurrent	v0.0.0-20180306012644-bacd9c7ef1dd	Apache-2.0
github.com/modern-go/reflect2	v1.0.2	Apache-2.0
github.com/natefinch/lumberjack	v2.0.0	MIT
github.com/neelance/astrewrite	v0.0.0-20160511093645-99348263ae86	BSD-2-Clause
github.com/neelance/sourcemap	v0.0.0-20200213170602-2833bce08e4c	BSD-2-Clause

Name	Version	License
github.com/nfnt/resize	v0.0.0-20180221191011-83c6a9932646	ISC
github.com/oliveagle/jsonpath	v0.0.0-20180606110733-2e52cf6e6852	MIT
github.com/onsi/ginkgo	v1.8.0	MIT
github.com/onsi/gomega	v1.5.0	MIT
github.com/oov/psd	v0.0.0-20220121172623-5db5eafcecb	MIT
github.com/paulmach/orb	v0.7.1	MIT
github.com/paulmach/osm	v0.5.0	MIT
github.com/pelletier/go-toml/v2	v2.0.5	MIT
github.com/pingcap/errors	v0.11.4	BSD-2-Clause
github.com/pkg/browser	v0.0.0-20210911075715-681adbf594b8	BSD-2-Clause
github.com/pkg/diff	v0.0.0-20210226163009-20ebb0f2a09e	BSD-3-Clause
github.com/pkg/errors	v0.9.1	BSD-2-Clause
github.com/pmezard/go-difflib	v1.0.0	BSD-3-Clause
github.com/prometheus/client_model	v0.0.0-20190812154241-14fe0d1b01d4	Apache-2.0
github.com/rogppe/fastuuid	v1.2.0	BSD-3-Clause
github.com/rogppe/go-internal	v1.9.0	BSD-3-Clause

Name	Version	License
github.com/russellhaering/goxmldsig	v1.2.0	Apache-2.0
github.com/russross/blackfriday/v2	v2.1.0	BSD-2-Clause
github.com/schollz/closestmatch	v2.1.0	MIT
github.com/shurcool/go	v0.0.0-20200502201357-93f07166e636	MIT
github.com/shurcool/httpfs	v0.0.0-20190707220628-8d4bc4ba7749	MIT
github.com/shurcool/vfsgen	v0.0.0-20200824052919-0d455de96546	MIT
github.com/sirupsen/logrus	v1.9.2	MIT
github.com/spf13/cobra	v1.2.1	Apache-2.0
github.com/spf13/pflag	v1.0.5	BSD-3-Clause
github.com/srwiley/oksvg	v0.0.0-20220128195007-1f435e4c2b44	BSD-3-Clause
github.com/srwiley/rasterx	v0.0.0-20220615024203-67b7089efd25	BSD-3-Clause
github.com/stretchr/objx	v0.5.0	MIT
github.com/stretchr/testify	v1.8.2	MIT
github.com/tdewolff/argp	v0.0.0-20221007181215-ebef9ed8a560	MIT
github.com/tdewolff/minify	v2.3.6	MIT
github.com/tdewolff/minify/v2	v2.12.5	MIT
github.com/tdewolff/parse	v2.3.4	MIT
github.com/tdewolff/parse/v2	v2.6.6	MIT

Name	Version	License
github.com/tdewolff/test	v1.0.7	MIT
github.com/tevino/abool	v1.2.0	MIT
github.com/ugorji/go/codec	v1.2.7	MIT
github.com/ulikunitz/xz	v0.5.6	BSD-3- Clause
github.com/urfave/negroni	v1.0.0	MIT
github.com/valyala/bytebufferpool	v1.0.0	MIT
github.com/valyala/fasthttp	v1.40.0	MIT
github.com/valyala/fasttemplate	v1.2.2	MIT
github.com/vmihailenco/msgpack/v5	v5.3.5	BSD-2- Clause
github.com/vmihailenco/tagparser/v2	v2.0.0	BSD-2- Clause
github.com/wcharczuk/go-chart/v2	v2.1.0	MIT
github.com/yosssi/ace	v0.0.5	MIT
github.com/yuin/goldmark	v1.4.13	MIT
github.com/zenazn/goji	v1.0.1	MIT
go.opencensus.io	v0.24.0	Apache-2.0
go.opentelemetry.io/proto/otlp	v0.7.0	Apache-2.0
golang.org/x/crypto	v0.9.0	BSD-3- Clause
golang.org/x/exp	v0.0.0-20230519143937- 03e91628a987	BSD-3- Clause
golang.org/x/exp/shiny	v0.0.0-20230425010034- 47ecfdc1ba53	BSD-3- Clause

Name	Version	License
golang.org/x/image	v0.7.0	BSD-3-Clause
golang.org/x/lint	v0.0.0-20190313153728-d0100b6bd8b3	BSD-3-Clause
golang.org/x/mobile	v0.0.0-20220518205345-8578da9835fd	BSD-3-Clause
golang.org/x/mod	v0.8.0	BSD-3-Clause
golang.org/x/net	v0.10.0	BSD-3-Clause
golang.org/x/oauth2	v0.8.0	BSD-3-Clause
golang.org/x/sync	v0.2.0	BSD-3-Clause
golang.org/x/sys	v0.8.0	BSD-3-Clause
golang.org/x/term	v0.8.0	BSD-3-Clause
golang.org/x/text	v0.9.0	BSD-3-Clause
golang.org/x/time	v0.3.0	BSD-3-Clause
golang.org/x/tools	v0.6.0	BSD-3-Clause
golang.org/x/xerrors	v0.0.0-20200804184101-5ec99f83aff1	BSD-3-Clause
gonum.org/v1/gonum	v0.13.0	BSD-3-Clause

Name	Version	License
gonum.org/v1/plot	v0.13.0	BSD-3-Clause
google.golang.org/api	v0.123.0	BSD-3-Clause
google.golang.org/appengine	v1.6.7	Apache-2.0
google.golang.org/genproto	v0.0.0-20230410155749-daa745c078e1	Apache-2.0
google.golang.org/genproto/googleapis/rpc	v0.0.0-20230530153820-e85fd2cbaebc	Apache-2.0
google.golang.org/grpc	v1.55.0	Apache-2.0
google.golang.org/protobuf	v1.30.0	BSD-3-Clause
gopkg.in/alexcesaro/quotedprintable.v3	v3.0.0-20150716171945-2caba252f4dc	MIT
gopkg.in/asn1-ber.v1	v1.0.0-20181015200546-f715ec2f112d	MIT
gopkg.in/check.v1	v1.0.0-20201130134442-10cb98267c6c	BSD-2-Clause
gopkg.in/errgo.v2	v2.1.0	BSD-3-Clause
gopkg.in/fsnotify.v1	v1.4.7	BSD-3-Clause
gopkg.in/gomail.v2	v2.0.0-20160411212932-81ebce5c23df	MIT
gopkg.in/ini.v1	v1.67.0	Apache-2.0
gopkg.in/ldap.v2	v2.5.1	MIT
gopkg.in/natefinch/lumberjack.v2	v2.2.1	MIT

Name	Version	License
gopkg.in/tomb.v1	v1.0.0-20141024135613-dd632973f1e7	BSD-3-Clause
gopkg.in/yaml.v2	v2.4.0	Apache-2.0
gopkg.in/yaml.v3	v3.0.1	MIT
gotest.tools	v2.2.0	Apache-2.0
honnef.co/go/js/dom	v0.0.0-20210725211120-f030747120f2	MIT
honnef.co/go/tools	v0.0.0-20190523083050-ea95bdfd59fc	MIT
rsc.io/pdf	v0.1.1	BSD-3-Clause

Transfer Utility

Name	Version	License
cloud.google.com/go	v0.26.0	Apache-2.0
cloud.google.com/go/compute	v1.18.0	Apache-2.0
cloud.google.com/go/compute/metadata	v0.2.3	Apache-2.0
github.com/BurntSushi/toml	v1.2.1	MIT
github.com/ByteArena/poly2tri-go	v0.0.0-20170716161910-d102ad91854f	BSD-3-Clause
github.com/CloudyKit/fastprinter	v0.0.0-20200109182630-33d98a066a53	MIT
github.com/CloudyKit/jet/v6	v6.2.0	Apache-2.0
github.com/DATA-DOG/go-sqlmock	v1.3.3	BSD-3-

Name	Version	License
		Clause
github.com/DATA-DOG/godog	v0.7.13	BSD-3- Clause
github.com/Joker/jade	v1.1.3	BSD-3- Clause
github.com/Shopify/goreferrer	v0.0.0-20220729165902-8cddb4f5de06	MIT
github.com/adrg/strutil	v0.3.0	MIT
github.com/adrg/sysfont	v0.1.2	MIT
github.com/adrg/xdg	v0.4.0	MIT
github.com/alecthomas/template	v0.0.0-20160405071501-a0175ee3bccc	BSD-3- Clause
github.com/alecthomas/units	v0.0.0-20151022065526-2efee857e7cf	MIT
github.com/andybalholm/brotli	v1.0.5	MIT
github.com/aymerick/douceur	v0.2.0	MIT
github.com/benoitkugler/textlayout	v0.1.3	MIT
github.com/beorn7/perks	v1.0.0	MIT
github.com/census-instrumentation/opencensus-proto	v0.4.1	Apache-2.0
github.com/cespare/xxhash/v2	v2.2.0	MIT
github.com/client9/misspell	v0.3.4	MIT
github.com/cncf/udpa/go	v0.0.0-20220112060539-c52dc94e7fbe	Apache-2.0
github.com/cncf/xds/go	v0.0.0-20230310173818-32f1caf87195	Apache-2.0

Name	Version	License
github.com/codegangsta/inject	v0.0.0-20150114235600-33e0aa1cb7c0	MIT
github.com/corona10/goimagehash	v1.1.0	BSD-2-Clause
github.com/creack/pty	v1.1.9	MIT
github.com/davecgh/go-spew	v1.1.1	ISC
github.com/disintegration/imaging	v1.6.2	MIT
github.com/dsnet/compress	v0.0.1	BSD-3-Clause
github.com/eknkc/amber	v0.0.0-20171010120322-cdade1c07385	MIT
github.com/envoyproxy/go-control-plane	v0.11.0	Apache-2.0
github.com/envoyproxy/protoc-gen-validate	v0.10.0	Apache-2.0
github.com/fatih/structs	v1.1.0	MIT
github.com/flosch/pongo2/v4	v4.0.2	MIT
github.com/fsnotify/fsnotify	v1.4.9	BSD-3-Clause
github.com/getsentry/sentry-go	v0.21.0	MIT
github.com/gin-contrib/sse	v0.1.0	MIT
github.com/gin-gonic/gin	v1.8.1	MIT
github.com/go-errors/errors	v1.4.2	MIT
github.com/go-kit/kit	v0.8.0	MIT
github.com/go-latex/latex	v0.0.0-20210823091927-c0d11ff05a81	BSD-3-Clause
github.com/go-logfmt/logfmt	v0.3.0	MIT

Name	Version	License
github.com/go-martini/martini	v0.0.0-20170121215854-22fa46961aab	MIT
github.com/go-playground/locales	v0.14.0	MIT
github.com/go-playground/universal-translator	v0.18.0	MIT
github.com/go-playground/validator/v10	v10.11.1	MIT
github.com/go-sql-driver/mysql	v1.6.0	MPL-2.0
github.com/go-stack/stack	v1.8.0	MIT
github.com/goccy/go-json	v0.9.11	MIT
github.com/gogo/protobuf	v1.1.1	BSD-3-Clause
github.com/gojuno/minimock/v3	v3.0.10	MIT
github.com/golang/freetype	v0.0.0-20170609003504-e2365dfdc4a0	FTL
github.com/golang/glog	v1.1.0	Apache-2.0
github.com/golang/mock	v1.1.1	Apache-2.0
github.com/golang/protobuf	v1.5.3	BSD-3-Clause
github.com/golang/snappy	v0.0.4	BSD-3-Clause
github.com/google/go-cmp	v0.5.9	BSD-3-Clause
github.com/google/gofuzz	v1.2.0	Apache-2.0
github.com/google/uuid	v1.3.0	BSD-3-Clause
github.com/gopherjs/gopherjs	v1.17.2	BSD-2-Clause

Name	Version	License
github.com/gorilla/css	v1.0.0	BSD-3-Clause
github.com/gorilla/websocket	v1.4.0	BSD-2-Clause
github.com/hexdigest/gowrap	v1.1.8	MIT
github.com/hpcloud/tail	v1.0.0	MIT
github.com/inconshreveable/mousetrap	v1.0.0	Apache-2.0
github.com/iris-contrib/schema	v0.0.6	BSD-3-Clause
github.com/jmoiron/sqlx	v1.3.5	MIT
github.com/josharian/intern	v1.0.0	MIT
github.com/json-iterator/go	v1.1.12	MIT
github.com/julienschmidt/httprouter	v1.2.0	BSD-3-Clause
github.com/kataras/blocks	v0.0.7	MIT
github.com/kataras/golog	v0.1.8	BSD-3-Clause
github.com/kataras/iris/v12	v12.2.0	BSD-3-Clause
github.com/kataras/pio	v0.0.11	BSD-3-Clause
github.com/kataras/sitemap	v0.0.6	MIT
github.com/kataras/tunnel	v0.0.4	MIT
github.com/klauspost/compress	v1.16.0	Apache-2.0
github.com/konsorten/go-windows-terminal-sequences	v1.0.1	MIT

Name	Version	License
github.com/kr/logfmt	v0.0.0-20140226030751-b84e30acd515	MIT
github.com/kr/pretty	v0.2.1	MIT
github.com/kr/pty	v1.1.1	MIT
github.com/kr/text	v0.2.0	MIT
github.com/labstack/echo/v4	v4.10.0	MIT
github.com/labstack/gommon	v0.4.0	MIT
github.com/leodido/go-urn	v1.2.1	MIT
github.com/mailgun/raymond/v2	v2.0.48	MIT
github.com/mailru/easyjson	v0.7.7	MIT
github.com/matttn/go-colorable	v0.1.13	MIT
github.com/matttn/go-isatty	v0.0.17	MIT
github.com/matttpproud/golang_protobuf_extensions	v1.0.1	Apache-2.0
github.com/microcosm-cc/bluemonday	v1.0.23	BSD-3-Clause
github.com/mitchellh/go-homedir	v1.1.0	MIT
github.com/modern-go/concurrent	v0.0.0-20180306012644-bacd9c7ef1dd	Apache-2.0
github.com/modern-go/reflect2	v1.0.2	Apache-2.0
github.com/mwitkow/go-conntrack	v0.0.0-20161129095857-cc309e4a2223	Apache-2.0
github.com/natefinch/lumberjack	v2.0.0	MIT
github.com/neelance/astrewrite	v0.0.0-20160511093645-99348263ae86	BSD-2-Clause

Name	Version	License
github.com/neelance/sourcemap	v0.0.0-20200213170602-2833bce08e4c	BSD-2-Clause
github.com/nfnt/resize	v0.0.0-20180221191011-83c6a9932646	ISC
github.com/oliveagle/jsonpath	v0.0.0-20180606110733-2e52cf6e6852	MIT
github.com/onsi/ginkgo	v1.8.0	MIT
github.com/onsi/gomega	v1.5.0	MIT
github.com/oov/psd	v0.0.0-20220121172623-5db5eafcecb	MIT
github.com/opentracing/opentracing-go	v1.0.2	MIT
github.com/pelletier/go-toml/v2	v2.0.5	MIT
github.com/pingcap/errors	v0.11.4	BSD-2-Clause
github.com/pkg/errors	v0.9.1	BSD-2-Clause
github.com/pmezard/go-difflib	v1.0.0	BSD-3-Clause
github.com/prometheus/client_golang	v1.0.0	Apache-2.0
github.com/prometheus/client_model	v0.0.0-20190812154241-14fe0d1b01d4	Apache-2.0
github.com/prometheus/common	v0.4.1	Apache-2.0
github.com/prometheus/procfs	v0.0.2	Apache-2.0
github.com/russross/blackfriday/v2	v2.1.0	BSD-2-Clause
github.com/schollz/closestmatch	v2.1.0	MIT

Name	Version	License
github.com/shurcool/go	v0.0.0-20200502201357-93f07166e636	MIT
github.com/shurcool/httpfs	v0.0.0-20190707220628-8d4bc4ba7749	MIT
github.com/shurcool/vfsgen	v0.0.0-20200824052919-0d455de96546	MIT
github.com/sirupsen/logrus	v1.9.2	MIT
github.com/spf13/cobra	v1.2.1	Apache-2.0
github.com/spf13/pflag	v1.0.5	BSD-3-Clause
github.com/stretchr/objx	v0.1.1	MIT
github.com/stretchr/testify	v1.8.2	MIT
github.com/tdewolff/canvas	v0.0.0-20220627195642-6566432f4b20	MIT
github.com/tdewolff/minify/v2	v2.12.4	MIT
github.com/tdewolff/parse/v2	v2.6.4	MIT
github.com/twitchtv/twirp	v5.8.0	Apache-2.0
github.com/ugorji/go/codec	v1.2.7	MIT
github.com/urfave/negroni	v1.0.0	MIT
github.com/valyala/bytebufferpool	v1.0.0	MIT
github.com/valyala/fasthttp	v1.40.0	MIT
github.com/valyala/fasttemplate	v1.2.2	MIT
github.com/vmihailenco/msgpack/v5	v5.3.5	BSD-2-Clause
github.com/vmihailenco/tagparser/v2	v2.0.0	BSD-2-Clause

Name	Version	License
github.com/wcharczuk/go-chart/v2	v2.1.0	MIT
github.com/yosssi/ace	v0.0.5	MIT
github.com/yuin/goldmark	v1.3.5	MIT
golang.org/x/crypto	v0.7.0	BSD-3- Clause
golang.org/x/exp	v0.0.0-20220613132600-b0d781184e0d	BSD-3- Clause
golang.org/x/image	v0.0.0-20220617043117-41969df76e82	BSD-3- Clause
golang.org/x/lint	v0.0.0-20190313153728-d0100b6bd8b3	BSD-3- Clause
golang.org/x/mod	v0.8.0	BSD-3- Clause
golang.org/x/net	v0.10.0	BSD-3- Clause
golang.org/x/oauth2	v0.6.0	BSD-3- Clause
golang.org/x/sync	v0.0.0-20210220032951-036812b2e83c	BSD-3- Clause
golang.org/x/sys	v0.8.0	BSD-3- Clause
golang.org/x/term	v0.8.0	BSD-3- Clause
golang.org/x/text	v0.9.0	BSD-3- Clause
golang.org/x/time	v0.3.0	BSD-3- Clause
golang.org/x/tools	v0.6.0	BSD-3-

Name	Version	License
		Clause
golang.org/x/xerrors	v0.0.0-20200804184101-5ec99f83aff1	BSD-3-Clause
gonum.org/v1/plot	v0.11.0	BSD-3-Clause
google.golang.org/appengine	v1.6.7	Apache-2.0
google.golang.org/genproto	v0.0.0-20230525234025-438c736192d0	Apache-2.0
google.golang.org/genproto/googleapis/rpc	v0.0.0-20230530153820-e85fd2cbaebc	Apache-2.0
google.golang.org/grpc	v1.55.0	Apache-2.0
google.golang.org/protobuf	v1.30.0	BSD-3-Clause
gopkg.in/alecthomas/kingpin.v2	v2.2.6	MIT
gopkg.in/check.v1	v1.0.0-20201130134442-10cb98267c6c	BSD-2-Clause
gopkg.in/fsnotify.v1	v1.4.7	BSD-3-Clause
gopkg.in/ini.v1	v1.67.0	Apache-2.0
gopkg.in/natefinch/lumberjack.v2	v2.0.0	MIT
gopkg.in/tomb.v1	v1.0.0-20141024135613-dd632973f1e7	BSD-3-Clause
gopkg.in/yaml.v2	v2.4.0	Apache-2.0
gopkg.in/yaml.v3	v3.0.1	MIT
honnef.co/go/tools	v0.0.0-20190523083050-ea95bdfd59fc	MIT

Python

Name	Version	License
Cython	0.29.30	Apache Software License
grpcio	1.46.3	Apache Software License
grpcio-tools	1.46.3	Apache Software License
protobuf	3.20.1	BSD-3-Clause
pyaaf2	1.4.0	MIT License
six	1.16.0	MIT License
typing_extensions	4.2.0	Python Software Foundation License

Flix Client

Package	Version	License
7zip-bin	5.1.1	MIT
@angular-eslint/bundled-angular-compiler	15.2.1	MIT
@angular-eslint/eslint-plugin-template	15.2.1	MIT
@angular-eslint/eslint-plugin	15.2.1	MIT
@angular-eslint/utils	15.2.1	MIT
@babel/code-frame	7.22.5	MIT
@babel/generator	7.22.5	MIT
@babel/helper-environment-visitor	7.22.5	MIT
@babel/helper-function-name	7.22.5	MIT
@babel/helper-hoist-variables	7.22.5	MIT

Package	Version	License
@babel/helper-split-export-declaration	7.22.5	MIT
@babel/helper-string-parser	7.22.5	MIT
@babel/helper-validator-identifier	7.22.5	MIT
@babel/highlight	7.22.5	MIT
@babel/parser	7.22.5	MIT
@babel/runtime	7.22.5	MIT
@babel/template	7.22.5	MIT
@babel/traverse	7.22.5	MIT
@babel/types	7.22.5	MIT
@cspotcode/source-map-support	0.8.1	MIT
@develar/schema-utils	2.6.5	MIT
@electron/asar	3.2.4	MIT
@electron/get	2.0.2	MIT
@electron/notarize	1.2.3	MIT
@electron/osx-sign	1.0.4	BSD-2-Clause
@electron/rebuild	3.2.13	MIT
@electron/universal	1.3.4	MIT
@eslint-community/eslint-utils	4.4.0	MIT
@eslint-community/regexpp	4.5.1	MIT
@eslint/eslintrc	2.0.3	MIT
@eslint/js	8.40.0	MIT
@gar/promisify	1.1.3	MIT
@humanwhocodes/config-array	0.11.10	Apache-2.0

Package	Version	License
@humanwhocodes/module-importer	1.0.1	Apache-2.0
@humanwhocodes/object-schema	1.2.1	BSD-3-Clause
@hutson/parse-repository-url	3.0.2	Apache-2.0
@isaacs/cliui	8.0.2	ISC
@jridgewell/gen-mapping	0.3.3	MIT
@jridgewell/resolve-uri	3.1.0	MIT
@jridgewell/set-array	1.1.2	MIT
@jridgewell/sourcemap-codec	1.4.14	MIT
@jridgewell/sourcemap-codec	1.4.15	MIT
@jridgewell/trace-mapping	0.3.18	MIT
@jridgewell/trace-mapping	0.3.9	MIT
@malept/cross-spawn-promise	1.1.1	Apache-2.0
@malept/cross-spawn-promise	2.0.0	Apache-2.0
@malept/flatpak-bundler	0.4.0	MIT
@nodelib/fs.scandir	2.1.5	MIT
@nodelib/fs.stat	2.0.5	MIT
@nodelib/fs.walk	1.2.8	MIT
@npmcli/fs	2.1.2	ISC
@npmcli/move-file	2.0.1	MIT
@pkgjs/parseargs	0.11.0	MIT
@playwright/test	1.33.0	Apache-2.0
@sindresorhus/is	4.6.0	MIT
@szmarczak/http-timer	4.0.6	MIT

Package	Version	License
@tootallnate/once	2.0.0	MIT
@tsconfig/node10	1.0.9	MIT
@tsconfig/node12	1.0.11	MIT
@tsconfig/node14	1.0.3	MIT
@tsconfig/node16	1.0.4	MIT
@types/cacheable-request	6.0.3	MIT
@types/http-cache-semantics	4.0.1	MIT
@types/jasmine	4.3.1	MIT
@types/json-schema	7.0.12	MIT
@types/keyv	3.1.4	MIT
@types/minimist	1.2.2	MIT
@types/node	16.18.25	MIT
@types/node	18.16.18	MIT
@types/normalize-package-data	2.4.1	MIT
@types/responselike	1.0.0	MIT
@types/semver	7.5.0	MIT
@types/yargs-parser	21.0.0	MIT
@types/yargs	17.0.24	MIT
@types/yauzl	2.10.0	MIT
@typescript-eslint/eslint-plugin	5.59.5	MIT
@typescript-eslint/experimental-utils	5.59.9	MIT
@typescript-eslint/parser	5.59.5	BSD-2-Clause
@typescript-eslint/scope-manager	5.48.2	MIT

Package	Version	License
@typescript-eslint/scope-manager	5.59.5	MIT
@typescript-eslint/scope-manager	5.59.9	MIT
@typescript-eslint/type-utils	5.48.2	MIT
@typescript-eslint/type-utils	5.59.5	MIT
@typescript-eslint/types	5.48.2	MIT
@typescript-eslint/types	5.59.5	MIT
@typescript-eslint/types	5.59.9	MIT
@typescript-eslint/typescript-estree	5.48.2	BSD-2-Clause
@typescript-eslint/typescript-estree	5.59.5	BSD-2-Clause
@typescript-eslint/typescript-estree	5.59.9	BSD-2-Clause
@typescript-eslint/utils	5.48.2	MIT
@typescript-eslint/utils	5.59.5	MIT
@typescript-eslint/utils	5.59.9	MIT
@typescript-eslint/visitor-keys	5.48.2	MIT
@typescript-eslint/visitor-keys	5.59.5	MIT
@typescript-eslint/visitor-keys	5.59.9	MIT
Flix	7.0.0	UNLICENSED
JSONStream	1.3.5	(MIT OR Apache-2.0)
abbrev	1.1.1	ISC
acorn-jsx	5.3.2	MIT
acorn-walk	8.2.0	MIT
acorn	8.8.2	MIT
add-stream	1.0.0	MIT

Package	Version	License
agent-base	6.0.2	MIT
agentkeepalive	4.3.0	MIT
aggregate-error	3.1.0	MIT
aggregate-error	4.0.1	MIT
ajv-keywords	3.5.2	MIT
ajv	6.12.6	MIT
ansi-regex	5.0.1	MIT
ansi-regex	6.0.1	MIT
ansi-styles	3.2.1	MIT
ansi-styles	4.3.0	MIT
ansi-styles	6.2.1	MIT
anymatch	3.1.3	ISC
app-builder-bin	4.0.0	MIT
app-builder-lib	24.3.0	MIT
aproba	2.0.0	ISC
are-we-there-yet	3.0.1	ISC
arg	4.1.3	MIT
argparse	2.0.1	Python-2.0
aria-query	5.1.3	Apache-2.0
array-buffer-byte-length	1.0.0	MIT
array-ify	1.0.0	MIT
array-union	2.1.0	MIT
arrify	1.0.1	MIT

Package	Version	License
async-exit-hook	2.0.1	MIT
async	3.2.4	MIT
asynckit	0.4.0	MIT
at-least-node	1.0.0	ISC
available-typed-arrays	1.0.5	MIT
axobject-query	3.1.1	Apache-2.0
balanced-match	1.0.2	MIT
base64-js	1.5.1	MIT
bent	7.3.12	Apache-2.0
binary-extensions	2.2.0	MIT
bl	4.1.0	MIT
bluebird-lst	1.0.9	MIT
bluebird	3.7.2	MIT
boolean	3.2.0	MIT
brace-expansion	1.1.11	MIT
brace-expansion	2.0.1	MIT
braces	3.0.2	MIT
buffer-crc32	0.2.13	MIT
buffer-equal	1.0.1	MIT
buffer-from	1.1.2	MIT
buffer	5.7.1	MIT
builder-util-runtime	9.2.1	MIT
builder-util	24.3.0	MIT

Package	Version	License
bytesish	0.4.4	(Apache-2.0 AND MIT)
cacache	16.1.3	ISC
cacheable-lookup	5.0.4	MIT
cacheable-request	7.0.4	MIT
call-bind	1.0.2	MIT
callsites	3.1.0	MIT
camelcase-keys	6.2.2	MIT
camelcase-keys	7.0.2	MIT
camelcase	5.3.1	MIT
camelcase	6.3.0	MIT
caseless	0.12.0	Apache-2.0
chalk	2.4.2	MIT
chalk	4.1.2	MIT
chokidar	3.5.3	MIT
chownr	2.0.0	ISC
chromium-pickle-js	0.2.0	MIT
ci-info	3.8.0	MIT
clean-stack	2.2.0	MIT
clean-stack	4.2.0	MIT
cli-cursor	3.1.0	MIT
cli-spinners	2.9.0	MIT
cliui	7.0.4	ISC
cliui	8.0.1	ISC

Package	Version	License
clone-response	1.0.3	MIT
clone	1.0.4	MIT
color-convert	1.9.3	MIT
color-convert	2.0.1	MIT
color-name	1.1.3	MIT
color-name	1.1.4	MIT
color-support	1.1.3	ISC
combined-stream	1.0.8	MIT
commander	5.1.0	MIT
common-tags	1.8.2	MIT
compare-func	2.0.0	MIT
compare-version	0.1.2	MIT
concat-map	0.0.1	MIT
concurrently	8.0.1	MIT
config-file-ts	0.2.4	MIT
console-control-strings	1.1.0	ISC
conventional-changelog-angular	5.0.13	ISC
conventional-changelog-atom	2.0.8	ISC
conventional-changelog-cli	2.2.2	MIT
conventional-changelog-codemirror	2.0.8	ISC
conventional-changelog-conventionalcommits	4.6.3	ISC
conventional-changelog-core	4.2.4	MIT

Package	Version	License
conventional-changelog-ember	2.0.9	ISC
conventional-changelog-eslint	3.0.9	ISC
conventional-changelog-express	2.0.6	ISC
conventional-changelog-jquery	3.0.11	ISC
conventional-changelog-jshint	2.0.9	ISC
conventional-changelog-preset-loader	2.3.4	MIT
conventional-changelog-writer	5.0.1	MIT
conventional-changelog	3.1.25	MIT
conventional-commits-filter	2.0.7	MIT
conventional-commits-parser	3.2.4	MIT
core-util-is	1.0.2	MIT
create-require	1.1.1	MIT
cross-spawn	7.0.3	MIT
dargs	7.0.0	MIT
date-fns	2.30.0	MIT
dateformat	3.0.3	MIT
debug	3.2.7	MIT
debug	4.3.4	MIT
decamelize-keys	1.1.1	MIT
decamelize	1.2.0	MIT
decamelize	5.0.1	MIT
decompress-response	6.0.0	MIT
deep-equal	2.2.1	MIT

Package	Version	License
deep-is	0.1.4	MIT
defaults	1.0.4	MIT
defer-to-connect	2.0.1	MIT
define-properties	1.2.0	MIT
del-cli	5.0.0	MIT
del	7.0.0	MIT
delayed-stream	1.0.0	MIT
delegates	1.0.0	MIT
depd	2.0.0	MIT
detect-libc	2.0.1	Apache-2.0
detect-node	2.1.0	MIT
diff	4.0.2	BSD-3-Clause
dir-compare	3.3.0	MIT
dir-glob	3.0.1	MIT
dmg-builder	24.3.0	MIT
doctrine	3.0.0	Apache-2.0
dot-prop	5.3.0	MIT
dotenv-expand	5.1.0	BSD-2-Clause
dotenv	9.0.2	BSD-2-Clause
eastasianwidth	0.2.0	MIT
ejs	3.1.9	Apache-2.0
electron-builder	24.3.0	MIT
electron-notarize	1.2.2	MIT

Package	Version	License
electron-publish	24.3.0	MIT
electron	24.3.0	MIT
emoji-regex	8.0.0	MIT
emoji-regex	9.2.2	MIT
encoding	0.1.13	MIT
end-of-stream	1.4.4	MIT
env-paths	2.2.1	MIT
err-code	2.0.3	MIT
error-ex	1.3.2	MIT
es-get-iterator	1.1.3	MIT
es6-error	4.1.1	MIT
escalade	3.1.1	MIT
escape-string-regexp	1.0.5	MIT
escape-string-regexp	4.0.0	MIT
escape-string-regexp	5.0.0	MIT
eslint-config-prettier	8.8.0	MIT
eslint-etc	5.2.1	MIT
eslint-plugin-rxjs-angular	2.0.0	MIT
eslint-plugin-rxjs	5.0.2	MIT
eslint-scope	5.1.1	BSD-2-Clause
eslint-scope	7.2.0	BSD-2-Clause
eslint-utils	3.0.0	MIT
eslint-visitor-keys	2.1.0	Apache-2.0

Package	Version	License
eslint-visitor-keys	3.4.1	Apache-2.0
eslint	8.40.0	MIT
espre	9.5.2	BSD-2-Clause
esquery	1.5.0	BSD-3-Clause
esrecurse	4.3.0	BSD-2-Clause
estraverse	4.3.0	BSD-2-Clause
estraverse	5.3.0	BSD-2-Clause
esutils	2.0.3	BSD-2-Clause
extract-zip	2.0.1	BSD-2-Clause
fast-deep-equal	3.1.3	MIT
fast-glob	3.2.12	MIT
fast-json-stable-stringify	2.1.0	MIT
fast-levenshtein	2.0.6	MIT
fastq	1.15.0	ISC
fd-slicer	1.1.0	MIT
file-entry-cache	6.0.1	MIT
filelist	1.0.4	Apache-2.0
fill-range	7.0.1	MIT
find-up	2.1.0	MIT
find-up	4.1.0	MIT
find-up	5.0.0	MIT
flat-cache	3.0.4	MIT
flatted	3.2.7	ISC

Package	Version	License
for-each	0.3.3	MIT
foreground-child	3.1.1	ISC
form-data	4.0.0	MIT
fs-extra	10.1.0	MIT
fs-extra	8.1.0	MIT
fs-extra	9.1.0	MIT
fs-minipass	2.1.0	ISC
fs.realpath	1.0.0	ISC
function-bind	1.1.1	MIT
functions-have-names	1.2.3	MIT
gauge	4.0.4	ISC
get-caller-file	2.0.5	ISC
get-intrinsic	1.2.1	MIT
get-pkg-repo	4.2.1	MIT
get-stream	5.2.0	MIT
git-raw-commits	2.0.11	MIT
git-remote-origin-url	2.0.0	MIT
git-semver-tags	4.1.1	MIT
gitconfiglocal	1.0.0	BSD*
glob-parent	5.1.2	ISC
glob-parent	6.0.2	ISC
glob	10.2.7	ISC
glob	7.2.3	ISC

Package	Version	License
glob	8.1.0	ISC
global-agent	3.0.0	BSD-3-Clause
globals	11.12.0	MIT
globals	13.20.0	MIT
globalthis	1.0.3	MIT
globby	11.1.0	MIT
globby	13.2.0	MIT
gopd	1.0.1	MIT
got	11.8.6	MIT
graceful-fs	4.2.11	ISC
grapheme-splitter	1.0.4	MIT
handlebars	4.7.7	MIT
hard-rejection	2.1.0	MIT
has-bigints	1.0.2	MIT
has-flag	3.0.0	MIT
has-flag	4.0.0	MIT
has-property-descriptors	1.0.0	MIT
has-proto	1.0.1	MIT
has-symbols	1.0.3	MIT
has-tostringtag	1.0.0	MIT
has-unicode	2.0.1	ISC
has	1.0.3	MIT
hosted-git-info	2.8.9	ISC

Package	Version	License
hosted-git-info	4.1.0	ISC
http-cache-semantics	4.1.1	BSD-2-Clause
http-proxy-agent	5.0.0	MIT
http2-wrapper	1.0.3	MIT
https-proxy-agent	5.0.1	MIT
humanize-ms	1.2.1	MIT
iconv-lite	0.6.3	MIT
ieee754	1.2.1	BSD-3-Clause
ignore-by-default	1.0.1	ISC
ignore	5.2.4	MIT
import-fresh	3.3.0	MIT
imurmurhash	0.1.4	MIT
indent-string	4.0.0	MIT
indent-string	5.0.0	MIT
infer-owner	1.0.4	ISC
inflight	1.0.6	ISC
inherits	2.0.4	ISC
ini	1.3.8	ISC
internal-slot	1.0.5	MIT
ip	2.0.0	MIT
is-arguments	1.1.1	MIT
is-array-buffer	3.0.2	MIT
is-arrayish	0.2.1	MIT

Package	Version	License
is-bigint	1.0.4	MIT
is-binary-path	2.1.0	MIT
is-boolean-object	1.1.2	MIT
is-callable	1.2.7	MIT
is-ci	3.0.1	MIT
is-core-module	2.12.1	MIT
is-date-object	1.0.5	MIT
is-extglob	2.1.1	MIT
is-fullwidth-code-point	3.0.0	MIT
is-glob	4.0.3	MIT
is-interactive	1.0.0	MIT
is-lambda	1.0.1	MIT
is-map	2.0.2	MIT
is-number-object	1.0.7	MIT
is-number	7.0.0	MIT
is-obj	2.0.0	MIT
is-path-cwd	3.0.0	MIT
is-path-inside	3.0.3	MIT
is-path-inside	4.0.0	MIT
is-plain-obj	1.1.0	MIT
is-regex	1.1.4	MIT
is-set	2.0.2	MIT
is-shared-array-buffer	1.0.2	MIT

Package	Version	License
is-stream	2.0.1	MIT
is-string	1.0.7	MIT
is-symbol	1.0.4	MIT
is-text-path	1.0.1	MIT
is-typed-array	1.1.10	MIT
is-unicode-supported	0.1.0	MIT
is-weakmap	2.0.1	MIT
is-weakset	2.0.2	MIT
isarray	1.0.0	MIT
isarray	2.0.5	MIT
isbinaryfile	4.0.10	MIT
isbinaryfile	5.0.0	MIT
isexe	2.0.0	ISC
jackspeak	2.2.1	BlueOak-1.0.0
jake	10.8.7	Apache-2.0
jasmine-core	5.0.1	MIT
jasmine	5.0.2	MIT
js-sdsl	4.4.1	MIT
js-tokens	4.0.0	MIT
js-yaml	4.1.0	MIT
jsesc	2.5.2	MIT
json-buffer	3.0.1	MIT
json-parse-better-errors	1.0.2	MIT

Package	Version	License
json-parse-even-better-errors	2.3.1	MIT
json-schema-traverse	0.4.1	MIT
json-stable-stringify-without-jsonify	1.0.1	MIT
json-stringify-safe	5.0.1	ISC
json5	2.2.3	MIT
jsonfile	4.0.0	MIT
jsonfile	6.1.0	MIT
jsonparse	1.3.1	MIT
keyv	4.5.2	MIT
kind-of	6.0.3	MIT
kleur	3.0.3	MIT
lazy-val	1.0.5	MIT
levn	0.4.1	MIT
lines-and-columns	1.2.4	MIT
load-json-file	4.0.0	MIT
locate-path	2.0.0	MIT
locate-path	5.0.0	MIT
locate-path	6.0.0	MIT
lodash.ismatch	4.4.0	MIT
lodash.merge	4.6.2	MIT
lodash	4.17.21	MIT
log-symbols	4.1.0	MIT
lowercase-keys	2.0.0	MIT

Package	Version	License
lru-cache	6.0.0	ISC
lru-cache	7.18.3	ISC
lru-cache	9.1.2	ISC
make-error	1.3.6	ISC
make-fetch-happen	10.2.1	ISC
map-obj	1.0.1	MIT
map-obj	4.3.0	MIT
matcher	3.0.0	MIT
meow	10.1.5	MIT
meow	8.1.2	MIT
merge2	1.4.1	MIT
micromatch	4.0.5	MIT
mime-db	1.52.0	MIT
mime-types	2.1.35	MIT
mime	2.6.0	MIT
mimic-fn	2.1.0	MIT
mimic-response	1.0.1	MIT
mimic-response	3.1.0	MIT
min-indent	1.0.1	MIT
minimatch	3.1.2	ISC
minimatch	5.1.6	ISC
minimatch	9.0.1	ISC
minimist-options	4.1.0	MIT

Package	Version	License
minimist	1.2.8	MIT
minipass-collect	1.0.2	ISC
minipass-fetch	2.1.2	MIT
minipass-flush	1.0.5	ISC
minipass-pipeline	1.2.4	ISC
minipass-sized	1.0.3	ISC
minipass	3.3.6	ISC
minipass	5.0.0	ISC
minipass	6.0.2	ISC
minizlib	2.1.2	MIT
mkdirp	1.0.4	MIT
modify-values	1.0.1	MIT
ms	2.1.2	MIT
natural-compare-lite	1.4.0	MIT
natural-compare	1.4.0	MIT
negotiator	0.6.3	MIT
neo-async	2.6.2	MIT
node-abi	3.45.0	MIT
node-api-version	0.1.4	MIT
node-gyp	9.3.1	MIT
nodemon	2.0.22	MIT
nopt	1.0.10	MIT
nopt	6.0.0	ISC

Package	Version	License
normalize-package-data	2.5.0	BSD-2-Clause
normalize-package-data	3.0.3	BSD-2-Clause
normalize-path	3.0.0	MIT
normalize-url	6.1.0	MIT
npmlog	6.0.2	ISC
object-inspect	1.12.3	MIT
object-is	1.1.5	MIT
object-keys	1.1.1	MIT
object.assign	4.1.4	MIT
once	1.4.0	ISC
onetime	5.1.2	MIT
optionator	0.9.1	MIT
ora	5.4.1	MIT
p-cancelable	2.1.1	MIT
p-limit	1.3.0	MIT
p-limit	2.3.0	MIT
p-limit	3.1.0	MIT
p-locate	2.0.0	MIT
p-locate	4.1.0	MIT
p-locate	5.0.0	MIT
p-map	4.0.0	MIT
p-map	5.5.0	MIT
p-try	1.0.0	MIT

Package	Version	License
p-try	2.2.0	MIT
parent-module	1.0.1	MIT
parse-json	4.0.0	MIT
parse-json	5.2.0	MIT
path-exists	3.0.0	MIT
path-exists	4.0.0	MIT
path-is-absolute	1.0.1	MIT
path-key	3.1.1	MIT
path-parse	1.0.7	MIT
path-scurry	1.9.2	BlueOak-1.0.0
path-type	3.0.0	MIT
path-type	4.0.0	MIT
pend	1.2.0	MIT
picomatch	2.3.1	MIT
pify	2.3.0	MIT
pify	3.0.0	MIT
playwright-core	1.33.0	Apache-2.0
playwright	1.33.0	Apache-2.0
plist	3.0.6	MIT
prelude-ls	1.2.1	MIT
prettier	3.0.3	MIT
process-nextick-args	2.0.1	MIT
progress	2.0.3	MIT

Package	Version	License
promise-inflight	1.0.1	ISC
promise-retry	2.0.1	MIT
prompts	2.4.2	MIT
pstree.remy	1.1.8	MIT
pump	3.0.0	MIT
punycode	2.3.0	MIT
q	1.5.1	MIT
queue-microtask	1.2.3	MIT
quick-lru	4.0.1	MIT
quick-lru	5.1.1	MIT
read-config-file	6.3.2	MIT
read-pkg-up	3.0.0	MIT
read-pkg-up	7.0.1	MIT
read-pkg-up	8.0.0	MIT
read-pkg	3.0.0	MIT
read-pkg	5.2.0	MIT
read-pkg	6.0.0	MIT
readable-stream	2.3.8	MIT
readable-stream	3.6.2	MIT
readdirp	3.6.0	MIT
redent	3.0.0	MIT
redent	4.0.0	MIT
regenerator-runtime	0.13.11	MIT

Package	Version	License
regexp.prototype.flags	1.5.0	MIT
require-directory	2.1.1	MIT
requireindex	1.2.0	MIT
resolve-alpn	1.2.1	MIT
resolve-from	4.0.0	MIT
resolve	1.22.1	MIT
responselike	2.0.1	MIT
restore-cursor	3.1.0	MIT
retry	0.12.0	MIT
reusify	1.0.4	MIT
rimraf	3.0.2	ISC
roarr	2.15.4	BSD-3-Clause
run-parallel	1.2.0	MIT
rxjs-report-usage	1.0.6	MIT
rxjs	7.8.1	Apache-2.0
safe-buffer	5.1.2	MIT
safe-buffer	5.2.1	MIT
safer-buffer	2.1.2	MIT
sanitize-filename	1.6.3	WTFPL OR ISC
sax	1.2.4	ISC
semver-compare	1.0.0	MIT
semver	5.7.1	ISC
semver	6.3.0	ISC

Package	Version	License
semver	7.0.0	ISC
semver	7.3.8	ISC
serialize-error	7.0.1	MIT
set-blocking	2.0.0	ISC
shebang-command	2.0.0	MIT
shebang-regex	3.0.0	MIT
shell-quote	1.8.1	MIT
side-channel	1.0.4	MIT
signal-exit	3.0.7	ISC
signal-exit	4.0.2	ISC
simple-update-notifier	1.1.0	MIT
sisteransi	1.0.5	MIT
slash	3.0.0	MIT
slash	4.0.0	MIT
smart-buffer	4.2.0	MIT
socks-proxy-agent	7.0.0	MIT
socks	2.7.1	MIT
source-map-support	0.5.21	MIT
source-map	0.6.1	BSD-3-Clause
spawn-command	0.0.2-1	MIT
spdx-correct	3.2.0	Apache-2.0
spdx-exceptions	2.3.0	CC-BY-3.0
spdx-expression-parse	3.0.1	MIT

Package	Version	License
spdx-license-ids	3.0.13	CC0-1.0
split2	3.2.2	ISC
split	1.0.1	MIT
sprintf-js	1.1.2	BSD-3-Clause
ssri	9.0.1	ISC
stat-mode	1.0.0	MIT
stop-iteration-iterator	1.0.0	MIT
string-width	4.2.3	MIT
string-width	5.1.2	MIT
string_decoder	1.1.1	MIT
string_decoder	1.3.0	MIT
strip-ansi	6.0.1	MIT
strip-ansi	7.1.0	MIT
strip-bom	3.0.0	MIT
strip-indent	3.0.0	MIT
strip-indent	4.0.0	MIT
strip-json-comments	3.1.1	MIT
sumchecker	3.0.1	Apache-2.0
supports-color	5.5.0	MIT
supports-color	7.2.0	MIT
supports-color	8.1.1	MIT
supports-preserve-symlinks-flag	1.0.0	MIT
tar	6.1.15	ISC

Package	Version	License
temp-dir	2.0.0	MIT
temp-file	3.4.0	MIT
tempfile	3.0.0	MIT
text-extensions	1.9.0	MIT
text-table	0.2.0	MIT
through2	2.0.5	MIT
through2	4.0.2	MIT
through	2.3.8	MIT
tmp-promise	3.0.3	MIT
tmp	0.2.1	MIT
to-fast-properties	2.0.0	MIT
to-regex-range	5.0.1	MIT
touch	3.1.0	ISC
tree-kill	1.2.2	MIT
trim-newlines	3.0.1	MIT
trim-newlines	4.1.1	MIT
truncate-utf8-bytes	1.0.2	WTFPL
ts-node	10.9.1	MIT
tslib	1.14.1	0BSD
tslib	2.6.0	0BSD
tsutils-etc	1.4.2	MIT
tsutils	3.21.0	MIT
type-check	0.4.0	MIT

Package	Version	License
type-fest	0.13.1	(MIT OR CC0-1.0)
type-fest	0.18.1	(MIT OR CC0-1.0)
type-fest	0.20.2	(MIT OR CC0-1.0)
type-fest	0.6.0	(MIT OR CC0-1.0)
type-fest	0.8.1	(MIT OR CC0-1.0)
type-fest	1.4.0	(MIT OR CC0-1.0)
typescript	4.9.5	Apache-2.0
uglify-js	3.17.4	BSD-2-Clause
undefsafe	2.0.5	MIT
unique-filename	2.0.1	ISC
unique-slug	3.0.0	ISC
universalify	0.1.2	MIT
universalify	2.0.0	MIT
uri-js	4.4.1	BSD-2-Clause
utf8-byte-length	1.0.4	WTFPL
util-deprecate	1.0.2	MIT
uuid	3.4.0	MIT
v8-compile-cache-lib	3.0.1	MIT
validate-npm-package-license	3.0.4	Apache-2.0
wcwidth	1.0.1	MIT
which-boxed-primitive	1.0.2	MIT
which-collection	1.0.1	MIT
which-typed-array	1.1.9	MIT

Package	Version	License
which	2.0.2	ISC
wide-align	1.1.5	ISC
word-wrap	1.2.3	MIT
wordwrap	1.0.0	MIT
wrap-ansi	7.0.0	MIT
wrap-ansi	8.1.0	MIT
wrappy	1.0.2	ISC
xmlbuilder	15.1.1	MIT
xtend	4.0.2	MIT
y18n	5.0.8	ISC
yallist	4.0.0	ISC
yargs-parser	20.2.9	ISC
yargs-parser	21.1.1	ISC
yargs	16.2.0	MIT
yargs	17.6.2	MIT
yargs	17.7.2	MIT
yauszl	2.10.0	MIT
yn	3.1.1	MIT
yocto-queue	0.1.0	MIT
@ampproject/remapping	2.2.1	Apache-2.0
@angular-devkit/core	16.0.1	MIT
@angular-devkit/schematics-cli	16.0.1	MIT
@angular-devkit/schematics	16.0.1	MIT

Package	Version	License
@babel/compat-data	7.22.5	MIT
@babel/core	7.22.5	MIT
@babel/helper-compilation-targets	7.22.5	MIT
@babel/helper-module-imports	7.22.5	MIT
@babel/helper-module-transforms	7.22.5	MIT
@babel/helper-plugin-utils	7.22.5	MIT
@babel/helper-simple-access	7.22.5	MIT
@babel/helper-validator-option	7.22.5	MIT
@babel/helpers	7.22.5	MIT
@babel/plugin-syntax-async-generators	7.8.4	MIT
@babel/plugin-syntax-bigint	7.8.3	MIT
@babel/plugin-syntax-class-properties	7.12.13	MIT
@babel/plugin-syntax-import-meta	7.10.4	MIT
@babel/plugin-syntax-json-strings	7.8.3	MIT
@babel/plugin-syntax-jsx	7.22.5	MIT
@babel/plugin-syntax-logical-assignment-operators	7.10.4	MIT
@babel/plugin-syntax-nullish-coalescing-operator	7.8.3	MIT
@babel/plugin-syntax-numeric-separator	7.10.4	MIT
@babel/plugin-syntax-object-rest-spread	7.8.3	MIT
@babel/plugin-syntax-optional-catch-binding	7.8.3	MIT
@babel/plugin-syntax-optional-chaining	7.8.3	MIT

Package	Version	License
@babel/plugin-syntax-top-level-await	7.14.5	MIT
@babel/plugin-syntax-typescript	7.22.5	MIT
@bcoe/v8-coverage	0.2.3	MIT
@colors/colors	1.5.0	MIT
@fastify/ajv-compiler	3.5.0	MIT
@fastify/cors	8.3.0	MIT
@fastify/deepmerge	1.3.0	MIT
@fastify/error	3.2.1	MIT
@fastify/fast-json-stringify-compiler	4.3.0	MIT
@fastify/formbody	7.4.0	MIT
@fastify/middie	8.3.0	MIT
@grpc/grpc-js	1.8.14	Apache-2.0
@grpc/proto-loader	0.7.7	Apache-2.0
@istanbuljs/load-nyc-config	1.1.0	ISC
@istanbuljs/schema	0.1.3	MIT
@jest/console	29.5.0	MIT
@jest/core	29.5.0	MIT
@jest/environment	29.5.0	MIT
@jest/expect-utils	29.5.0	MIT
@jest/expect	29.5.0	MIT
@jest/fake-timers	29.5.0	MIT
@jest/globals	29.5.0	MIT
@jest/reporters	29.5.0	MIT

Package	Version	License
@jest/schemas	29.4.3	MIT
@jest/source-map	29.4.3	MIT
@jest/test-result	29.5.0	MIT
@jest/test-sequencer	29.5.0	MIT
@jest/transform	29.5.0	MIT
@jest/types	29.5.0	MIT
@jridgewell/resolve-uri	3.1.1	MIT
@jridgewell/source-map	0.3.3	MIT
@lukeed/csprng	1.1.0	MIT
@nestjs/cli	9.5.0	MIT
@nestjs/common	9.4.3	MIT
@nestjs/core	9.4.3	MIT
@nestjs/platform-express	9.4.3	MIT
@nestjs/platform-fastify	9.4.3	MIT
@nestjs/platform-socket.io	9.4.3	MIT
@nestjs/schematics	9.2.0	MIT
@nestjs/testing	9.4.3	MIT
@nestjs/websockets	9.4.3	MIT
@nuxtjs/opencollective	0.3.2	MIT
@protobufjs/aspromise	1.1.2	BSD-3-Clause
@protobufjs/base64	1.1.2	BSD-3-Clause
@protobufjs/codegen	2.0.4	BSD-3-Clause
@protobufjs/eventemitter	1.1.0	BSD-3-Clause

Package	Version	License
@protobufjs/fetch	1.1.0	BSD-3-Clause
@protobufjs/float	1.0.2	BSD-3-Clause
@protobufjs/inquire	1.1.0	BSD-3-Clause
@protobufjs/path	1.1.2	BSD-3-Clause
@protobufjs/pool	1.1.0	BSD-3-Clause
@protobufjs/utf8	1.1.0	BSD-3-Clause
@sinclair/typebox	0.25.24	MIT
@sinonjs/commons	3.0.0	BSD-3-Clause
@sinonjs/fake-timers	10.1.0	BSD-3-Clause
@socket.io/component-emitter	3.1.0	MIT
@types/babel__core	7.20.1	MIT
@types/babel__generator	7.6.4	MIT
@types/babel__template	7.4.1	MIT
@types/babel__traverse	7.20.1	MIT
@types/cookie	0.4.1	MIT
@types/cookiejar	2.1.2	MIT
@types/cors	2.8.13	MIT
@types/crypto-js	4.1.1	MIT
@types/eslint-scope	3.7.4	MIT
@types/eslint	8.40.2	MIT
@types/estree	1.0.1	MIT
@types/google-protobuf	3.15.6	MIT
@types/graceful-fs	4.1.6	MIT

Package	Version	License
@types/istanbul-lib-coverage	2.0.4	MIT
@types/istanbul-lib-report	3.0.0	MIT
@types/istanbul-reports	3.0.1	MIT
@types/jest	29.5.1	MIT
@types/long	4.0.2	MIT
@types/parse-json	4.0.0	MIT
@types/prettier	2.7.3	MIT
@types/stack-utils	2.0.1	MIT
@types/superagent	4.1.18	MIT
@types/supertest	2.0.12	MIT
@types/validator	13.7.17	MIT
@typescript-eslint/experimental-utils	5.59.11	MIT
@typescript-eslint/scope-manager	5.59.11	MIT
@typescript-eslint/types	5.59.11	MIT
@typescript-eslint/typescript-estree	5.59.11	BSD-2-Clause
@typescript-eslint/utils	5.59.11	MIT
@typescript-eslint/visitor-keys	5.59.11	MIT
@webassemblyjs/ast	1.11.6	MIT
@webassemblyjs/floating-point-hex-parser	1.11.6	MIT
@webassemblyjs/helper-api-error	1.11.6	MIT
@webassemblyjs/helper-buffer	1.11.6	MIT
@webassemblyjs/helper-numbers	1.11.6	MIT
@webassemblyjs/helper-wasm-bytecode	1.11.6	MIT

Package	Version	License
@webassemblyjs/helper-wasm-section	1.11.6	MIT
@webassemblyjs/ieee754	1.11.6	MIT
@webassemblyjs/leb128	1.11.6	Apache-2.0
@webassemblyjs/utf8	1.11.6	MIT
@webassemblyjs/wasm-edit	1.11.6	MIT
@webassemblyjs/wasm-gen	1.11.6	MIT
@webassemblyjs/wasm-opt	1.11.6	MIT
@webassemblyjs/wasm-parser	1.11.6	MIT
@webassemblyjs/wast-printer	1.11.6	MIT
@xtuc/ieee754	1.2.0	BSD-3-Clause
@xtuc/long	4.2.2	Apache-2.0
abort-controller	3.0.0	MIT
abstract-logging	2.0.1	MIT
accepts	1.3.8	MIT
acorn-import-assertions	1.9.0	MIT
ajv-formats	2.1.1	MIT
ajv	8.12.0	MIT
ansi-colors	4.1.3	MIT
ansi-escapes	4.3.2	MIT
ansi-styles	5.2.0	MIT
append-field	1.0.0	MIT
archy	1.0.0	MIT
argparse	1.0.10	MIT

Package	Version	License
array-flatten	1.1.1	MIT
asap	2.0.6	MIT
async	2.6.4	MIT
atomic-sleep	1.0.0	MIT
avvio	8.2.1	MIT
axios	1.4.0	MIT
babel-jest	29.5.0	MIT
babel-plugin-istanbul	6.1.1	BSD-3-Clause
babel-plugin-jest-hoist	29.5.0	MIT
babel-preset-current-node-syntax	1.0.1	MIT
babel-preset-jest	29.5.0	MIT
base64id	2.0.0	MIT
big-integer	1.6.51	Unlicense
binary	0.3.0	MIT
bluebird	3.4.7	MIT
body-parser	1.20.1	MIT
body-parser	1.20.2	MIT
browserslist	4.21.9	MIT
bs-logger	0.2.6	MIT
bser	2.1.1	Apache-2.0
buffer-indexof-polyfill	1.0.2	MIT
buffer	6.0.3	MIT
buffers	0.1.1	Custom:

Package	Version	License
		http://github.com/substack/node-bufferlist
builder-util-runtime	9.1.1	MIT
busboy	1.6.0	MIT
bytes	3.1.2	MIT
caniuse-lite	1.0.30001504	CC-BY-4.0
chainsaw	0.1.0	MIT*
char-regex	1.0.2	MIT
chardet	0.7.0	MIT
chrome-trace-event	1.0.3	MIT
cjs-module-lexer	1.2.3	MIT
class-transformer	0.5.1	MIT
class-validator	0.14.0	MIT
cli-table3	0.6.3	MIT
cli-width	3.0.0	ISC
co	4.6.0	MIT
collect-v8-coverage	1.0.1	MIT
commander	2.20.3	MIT
commander	4.1.1	MIT
component-emitter	1.3.0	MIT
concat-stream	1.6.2	MIT
consola	2.15.3	MIT
content-disposition	0.5.4	MIT

Package	Version	License
content-type	1.0.5	MIT
convert-source-map	1.9.0	MIT
convert-source-map	2.0.0	MIT
cookie-signature	1.0.6	MIT
cookie	0.4.2	MIT
cookie	0.5.0	MIT
cookiejar	2.1.4	MIT
core-util-is	1.0.3	MIT
cors	2.8.5	MIT
cosmiconfig	7.1.0	MIT
cross-env	7.0.3	MIT
crypto-js	4.1.1	MIT
debug	2.6.9	MIT
dedent	0.7.0	MIT
deepmerge	4.3.1	MIT
destroy	1.2.0	MIT
detect-newline	3.1.0	MIT
dezalgo	1.0.4	ISC
diff-sequences	29.4.3	MIT
duplexer2	0.1.4	BSD-3-Clause
ee-first	1.1.1	MIT
electron-json-storage	4.6.0	MIT
electron-log	4.4.8	MIT

Package	Version	License
electron-reload	1.5.0	MIT
electron-to-chromium	1.4.433	ISC
electron-updater	5.3.0	MIT
emittery	0.13.1	MIT
encodeurl	1.0.2	MIT
engine.io-parser	5.0.7	MIT
engine.io	6.4.2	MIT
enhanced-resolve	5.15.0	MIT
es-module-lexer	1.3.0	MIT
escape-html	1.0.3	MIT
escape-string-regexp	2.0.0	MIT
esprima	4.0.1	BSD-2-Clause
etag	1.8.1	MIT
event-target-shim	5.0.1	MIT
events	3.3.0	MIT
execa	4.1.0	MIT
execa	5.1.1	MIT
exit	0.1.2	MIT
expect	29.5.0	MIT
express	4.18.2	MIT
external-editor	3.1.0	MIT
fast-content-type-parse	1.0.0	MIT
fast-decode-uri-component	1.0.1	MIT

Package	Version	License
fast-json-stringify	5.7.0	MIT
fast-querystring	1.1.2	MIT
fast-redact	3.2.0	MIT
fast-safe-stringify	2.1.1	MIT
fast-uri	2.2.0	MIT
fastify-plugin	4.5.0	MIT
fastify	4.18.0	MIT
fb-watchman	2.0.2	Apache-2.0
figures	3.2.0	MIT
finalhandler	1.2.0	MIT
find-my-way	7.6.2	MIT
follow-redirects	1.15.2	MIT
fork-ts-checker-webpack-plugin	8.0.0	MIT
formidable	2.1.2	MIT
forwarded	0.2.0	MIT
fresh	0.5.2	MIT
fs-monkey	1.0.4	Unlicense
fstream	1.0.12	ISC
gensync	1.0.0-beta.2	MIT
get-package-type	0.1.0	MIT
get-stream	6.0.1	MIT
glob-to-regexp	0.4.1	BSD-2-Clause
glob	9.3.5	ISC

Package	Version	License
globby	13.1.4	MIT
google-protobuf	3.21.2	(BSD-3-Clause AND Apache-2.0)
hexoid	1.0.0	MIT
html-escaper	2.0.2	MIT
http-errors	2.0.0	MIT
human-signals	1.1.1	Apache-2.0
human-signals	2.1.0	Apache-2.0
iconv-lite	0.4.24	MIT
import-local	3.1.0	MIT
inquirer	8.2.4	MIT
inquirer	8.2.5	MIT
interpret	1.4.0	MIT
ipaddr.js	1.9.1	MIT
is-generator-fn	2.1.0	MIT
istanbul-lib-coverage	3.2.0	BSD-3-Clause
istanbul-lib-instrument	5.2.1	BSD-3-Clause
istanbul-lib-report	3.0.0	BSD-3-Clause
istanbul-lib-source-maps	4.0.1	BSD-3-Clause
istanbul-reports	3.1.5	BSD-3-Clause
iterare	1.2.1	ISC
jest-changed-files	29.5.0	MIT
jest-circus	29.5.0	MIT
jest-cli	29.5.0	MIT

Package	Version	License
jest-config	29.5.0	MIT
jest-diff	29.5.0	MIT
jest-docblock	29.4.3	MIT
jest-each	29.5.0	MIT
jest-environment-node	29.5.0	MIT
jest-get-type	29.4.3	MIT
jest-haste-map	29.5.0	MIT
jest-leak-detector	29.5.0	MIT
jest-matcher-utils	29.5.0	MIT
jest-message-util	29.5.0	MIT
jest-mock	29.5.0	MIT
jest-pnp-resolver	1.2.3	MIT
jest-regex-util	29.4.3	MIT
jest-resolve-dependencies	29.5.0	MIT
jest-resolve	29.5.0	MIT
jest-runner	29.5.0	MIT
jest-runtime	29.5.0	MIT
jest-snapshot	29.5.0	MIT
jest-util	29.5.0	MIT
jest-validate	29.5.0	MIT
jest-watcher	29.5.0	MIT
jest-worker	27.5.1	MIT
jest-worker	29.5.0	MIT

Package	Version	License
jest	29.5.0	MIT
js-yaml	3.14.1	MIT
json-schema-traverse	1.0.0	MIT
jsonc-parser	3.2.0	MIT
leven	3.1.0	MIT
libphonenumber-js	1.10.36	MIT
light-my-request	5.9.1	BSD-3-Clause
listenercount	1.0.1	ISC
loader-runner	4.3.0	MIT
lockfile	1.0.4	ISC
lodash.camelcase	4.3.0	MIT
lodash.escaperegexp	4.1.2	MIT
lodash.isequal	4.5.0	MIT
lodash.memoize	4.1.2	MIT
long	4.0.0	Apache-2.0
long	5.2.3	Apache-2.0
lru-cache	5.1.1	ISC
macos-release	2.5.1	MIT
magic-string	0.30.0	MIT
make-dir	3.1.0	MIT
makeerror	1.0.12	BSD-3-Clause
media-typer	0.3.0	MIT
memfs	3.5.3	Unlicense

Package	Version	License
merge-descriptors	1.0.1	MIT
merge-stream	2.0.0	MIT
methods	1.1.2	MIT
mime	1.6.0	MIT
minimatch	8.0.4	ISC
minipass	4.2.8	ISC
mkdirp	0.5.6	MIT
mnemonist	0.39.5	MIT
moment	2.29.4	MIT
ms	2.0.0	MIT
ms	2.1.3	MIT
multer	1.4.4-lts.1	MIT
mute-stream	0.0.8	ISC
node-abort-controller	3.1.1	MIT
node-emoji	1.11.0	MIT
node-fetch	2.6.11	MIT
node-int64	0.4.0	MIT
node-releases	2.0.12	MIT
npm-run-path	4.0.1	MIT
object-assign	4.1.1	MIT
object-hash	3.0.0	MIT
obliterator	2.0.4	MIT
on-exit-leak-free	2.1.0	MIT

Package	Version	License
on-finished	2.4.1	MIT
os-name	4.0.1	MIT
os-tmpdir	1.0.2	MIT
parseurl	1.3.3	MIT
path-to-regexp	0.1.7	MIT
path-to-regexp	3.2.0	MIT
path-to-regexp	6.2.1	MIT
picocolors	1.0.0	ISC
pino-abstract-transport	1.0.0	MIT
pino-std-serializers	6.2.1	MIT
pino	8.14.1	MIT
pirates	4.0.5	MIT
pkg-dir	4.2.0	MIT
pluralize	8.0.0	MIT
prettier	2.8.8	MIT
pretty-format	29.5.0	MIT
process-warning	2.2.0	MIT
process	0.11.10	MIT
protobufjs	7.2.3	BSD-3-Clause
proxy-addr	2.0.7	MIT
proxy-from-env	1.1.0	MIT
pure-rand	6.0.2	MIT
qs	6.11.0	BSD-3-Clause

Package	Version	License
quick-format-unescaped	4.0.4	MIT
randombytes	2.1.0	MIT
range-parser	1.2.1	MIT
raw-body	2.5.1	MIT
raw-body	2.5.2	MIT
react-is	18.2.0	MIT
readable-stream	4.4.0	MIT
real-require	0.2.0	MIT
rechoir	0.6.2	MIT
reflect-metadata	0.1.13	Apache-2.0
require-from-string	2.0.2	MIT
resolve-cwd	3.0.0	MIT
resolve-from	5.0.0	MIT
resolve.exports	2.0.2	MIT
resolve	1.22.2	MIT
ret	0.2.2	MIT
rfdc	1.3.0	MIT
rimraf	2.7.1	ISC
rimraf	4.4.1	ISC
run-async	2.4.1	MIT
safe-regex2	2.0.0	MIT
safe-stable-stringify	2.4.3	MIT
schema-utils	3.3.0	MIT

Package	Version	License
secure-json-parse	2.7.0	BSD-3-Clause
semver	7.5.1	ISC
send	0.18.0	MIT
serialize-javascript	6.0.1	BSD-3-Clause
serve-static	1.15.0	MIT
set-cookie-parser	2.6.0	MIT
setimmediate	1.0.5	MIT
setprototypeof	1.2.0	ISC
shelljs	0.8.5	BSD-3-Clause
socket.io-adapter	2.5.2	MIT
socket.io-parser	4.2.4	MIT
socket.io	4.6.2	MIT
sonic-boom	3.3.0	MIT
source-map-support	0.5.13	MIT
source-map	0.7.4	BSD-3-Clause
split2	4.2.0	ISC
sprintf-js	1.0.3	BSD-3-Clause
stack-utils	2.0.6	MIT
statuses	2.0.1	MIT
streamsearch	1.1.0	MIT
string-length	4.0.2	MIT
strip-bom	4.0.0	MIT
strip-final-newline	2.0.0	MIT

Package	Version	License
superagent	8.0.9	MIT
supertest	6.3.3	MIT
symbol-observable	4.0.0	MIT
tapable	2.2.1	MIT
tar	6.2.0	ISC
terser-webpack-plugin	5.3.9	MIT
terser	5.18.1	BSD-2-Clause
test-exclude	6.0.0	ISC
thread-stream	2.3.0	MIT
tiny-lru	11.0.1	BSD-3-Clause
tmp	0.0.33	MIT
tmpl	1.0.5	BSD-3-Clause
toidentifier	1.0.1	MIT
tr46	0.0.3	MIT
traverse	0.3.9	MIT*
ts-jest	29.1.0	MIT
ts-loader	9.4.3	MIT
tsconfig-paths-webpack-plugin	4.0.1	MIT
tsconfig-paths	4.2.0	MIT
tslib	2.5.3	0BSD
type-detect	4.0.8	MIT
type-fest	0.21.3	(MIT OR CC0-1.0)
type-is	1.6.18	MIT

Package	Version	License
typed-emitter	2.1.0	MIT
typedarray	0.0.6	MIT
uid	2.0.2	MIT
unpipe	1.0.0	MIT
unzipper	0.10.14	MIT
update-browserslist-db	1.0.11	MIT
utils-merge	1.0.1	MIT
uuid	9.0.0	MIT
v8-to-istanbul	9.1.0	ISC
validator	13.9.0	MIT
vary	1.1.2	MIT
walker	1.0.8	Apache-2.0
watchpack	2.4.0	MIT
webidl-conversions	3.0.1	BSD-2-Clause
webpack-node-externals	3.0.0	MIT
webpack-sources	3.2.3	MIT
webpack	5.82.1	MIT
webpack	5.87.0	MIT
whatwg-url	5.0.0	MIT
windows-release	4.0.0	MIT
write-file-atomic	2.4.3	ISC
write-file-atomic	4.0.2	ISC
ws	8.11.0	MIT

Package	Version	License
ws	8.13.0	MIT
yallist	3.1.1	ISC
yaml	1.10.2	ISC

Photoshop

Package	Version	License
@aashutoshrathi/word-wrap	1.2.6	MIT
@adobe/css-tools	4.2.0	MIT
@ampproject/remapping	2.2.1	Apache-2.0
@babel/code-frame	7.22.5	MIT
@babel/compat-data	7.22.5	MIT
@babel/core	7.22.1	MIT
@babel/generator	7.22.5	MIT
@babel/helper-annotate-as-pure	7.22.5	MIT
@babel/helper-builder-binary-assignment-operator-visitor	7.22.5	MIT
@babel/helper-builder-react-jsx	7.22.5	MIT
@babel/helper-compilation-targets	7.22.5	MIT
@babel/helper-create-class-features-plugin	7.22.5	MIT
@babel/helper-create-regexp-features-plugin	7.22.5	MIT
@babel/helper-define-polyfill-provider	0.4.0	MIT
@babel/helper-environment-visitor	7.22.5	MIT
@babel/helper-function-name	7.22.5	MIT

Package	Version	License
@babel/helper-hoist-variables	7.22.5	MIT
@babel/helper-member-expression-to-functions	7.22.5	MIT
@babel/helper-module-imports	7.22.5	MIT
@babel/helper-module-transforms	7.22.5	MIT
@babel/helper-optimise-call-expression	7.22.5	MIT
@babel/helper-plugin-utils	7.22.5	MIT
@babel/helper-remap-async-to-generator	7.22.5	MIT
@babel/helper-replace-supers	7.22.5	MIT
@babel/helper-simple-access	7.22.5	MIT
@babel/helper-skip-transparent-expression-wrappers	7.22.5	MIT
@babel/helper-split-export-declaration	7.22.5	MIT
@babel/helper-string-parser	7.22.5	MIT
@babel/helper-validator-identifier	7.22.5	MIT
@babel/helper-validator-option	7.22.5	MIT
@babel/helper-wrap-function	7.22.5	MIT
@babel/helpers	7.22.5	MIT
@babel/highlight	7.22.5	MIT
@babel/parser	7.22.5	MIT
@babel/plugin-bugfix-safari-id-destructuring-collision-in-function-expression	7.22.5	MIT
@babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining	7.22.5	MIT
@babel/plugin-proposal-class-properties	7.18.6	MIT

Package	Version	License
@babel/plugin-proposal-private-property-in-object	7.21.11	MIT
@babel/plugin-proposal-unicode-property-regex	7.18.6	MIT
@babel/plugin-syntax-async-generators	7.8.4	MIT
@babel/plugin-syntax-bigint	7.8.3	MIT
@babel/plugin-syntax-class-properties	7.12.13	MIT
@babel/plugin-syntax-class-static-block	7.14.5	MIT
@babel/plugin-syntax-dynamic-import	7.8.3	MIT
@babel/plugin-syntax-export-namespace-from	7.8.3	MIT
@babel/plugin-syntax-import-assertions	7.22.5	MIT
@babel/plugin-syntax-import-attributes	7.22.5	MIT
@babel/plugin-syntax-import-meta	7.10.4	MIT
@babel/plugin-syntax-json-strings	7.8.3	MIT
@babel/plugin-syntax-jsx	7.22.5	MIT
@babel/plugin-syntax-logical-assignment-operators	7.10.4	MIT
@babel/plugin-syntax-nullish-coalescing-operator	7.8.3	MIT
@babel/plugin-syntax-numeric-separator	7.10.4	MIT
@babel/plugin-syntax-object-rest-spread	7.8.3	MIT
@babel/plugin-syntax-optional-catch-binding	7.8.3	MIT
@babel/plugin-syntax-optional-chaining	7.8.3	MIT
@babel/plugin-syntax-private-property-in-object	7.14.5	MIT
@babel/plugin-syntax-top-level-await	7.14.5	MIT
@babel/plugin-syntax-typescript	7.22.5	MIT
@babel/plugin-syntax-unicode-sets-regex	7.18.6	MIT

Package	Version	License
@babel/plugin-transform-arrow-functions	7.22.5	MIT
@babel/plugin-transform-async-generator-functions	7.22.5	MIT
@babel/plugin-transform-async-to-generator	7.22.5	MIT
@babel/plugin-transform-block-scoped-functions	7.22.5	MIT
@babel/plugin-transform-block-scoping	7.22.5	MIT
@babel/plugin-transform-class-properties	7.22.5	MIT
@babel/plugin-transform-class-static-block	7.22.5	MIT
@babel/plugin-transform-classes	7.22.5	MIT
@babel/plugin-transform-computed-properties	7.22.5	MIT
@babel/plugin-transform-destructuring	7.22.5	MIT
@babel/plugin-transform-dotall-regex	7.22.5	MIT
@babel/plugin-transform-duplicate-keys	7.22.5	MIT
@babel/plugin-transform-dynamic-import	7.22.5	MIT
@babel/plugin-transform-exponentiation-operator	7.22.5	MIT
@babel/plugin-transform-export-namespace-from	7.22.5	MIT
@babel/plugin-transform-for-of	7.22.5	MIT
@babel/plugin-transform-function-name	7.22.5	MIT
@babel/plugin-transform-json-strings	7.22.5	MIT
@babel/plugin-transform-literals	7.22.5	MIT
@babel/plugin-transform-logical-assignment-operators	7.22.5	MIT
@babel/plugin-transform-member-expression-literals	7.22.5	MIT

Package	Version	License
@babel/plugin-transform-modules-amd	7.22.5	MIT
@babel/plugin-transform-modules-commonjs	7.22.5	MIT
@babel/plugin-transform-modules-systemjs	7.22.5	MIT
@babel/plugin-transform-modules-umd	7.22.5	MIT
@babel/plugin-transform-named-capturing-groups-regex	7.22.5	MIT
@babel/plugin-transform-new-target	7.22.5	MIT
@babel/plugin-transform-nullish-coalescing-operator	7.22.5	MIT
@babel/plugin-transform-numeric-separator	7.22.5	MIT
@babel/plugin-transform-object-rest-spread	7.22.5	MIT
@babel/plugin-transform-object-super	7.22.5	MIT
@babel/plugin-transform-optional-catch-binding	7.22.5	MIT
@babel/plugin-transform-optional-chaining	7.22.5	MIT
@babel/plugin-transform-parameters	7.22.5	MIT
@babel/plugin-transform-private-methods	7.22.5	MIT
@babel/plugin-transform-private-property-in-object	7.22.5	MIT
@babel/plugin-transform-property-literals	7.22.5	MIT
@babel/plugin-transform-react-constant-elements	7.22.3	MIT
@babel/plugin-transform-react-display-name	7.22.5	MIT
@babel/plugin-transform-react-inline-elements	7.21.0	MIT
@babel/plugin-transform-react-jsx-development	7.22.5	MIT
@babel/plugin-transform-react-jsx	7.22.5	MIT

Package	Version	License
@babel/plugin-transform-react-pure-annotations	7.22.5	MIT
@babel/plugin-transform-regenerator	7.22.5	MIT
@babel/plugin-transform-reserved-words	7.22.5	MIT
@babel/plugin-transform-runtime	7.22.4	MIT
@babel/plugin-transform-shorthand-properties	7.22.5	MIT
@babel/plugin-transform-spread	7.22.5	MIT
@babel/plugin-transform-sticky-regex	7.22.5	MIT
@babel/plugin-transform-template-literals	7.22.5	MIT
@babel/plugin-transform-typeof-symbol	7.22.5	MIT
@babel/plugin-transform-typescript	7.22.5	MIT
@babel/plugin-transform-unicode-escapes	7.22.5	MIT
@babel/plugin-transform-unicode-property-regex	7.22.5	MIT
@babel/plugin-transform-unicode-regex	7.22.5	MIT
@babel/plugin-transform-unicode-sets-regex	7.22.5	MIT
@babel/polyfill	7.12.1	MIT
@babel/preset-env	7.22.4	MIT
@babel/preset-modules	0.1.5	MIT
@babel/preset-react	7.22.3	MIT
@babel/preset-typescript	7.21.5	MIT
@babel/regjsgen	0.8.0	MIT
@babel/runtime	7.22.3	MIT
@babel/template	7.22.5	MIT
@babel/traverse	7.22.5	MIT

Package	Version	License
@babel/types	7.22.5	MIT
@bcoe/v8-coverage	0.2.3	MIT
@discoveryjs/json-ext	0.5.7	MIT
@eslint-community/eslint-utils	4.4.0	MIT
@eslint-community/regexpp	4.5.1	MIT
@eslint/eslintrc	2.1.0	MIT
@eslint/js	8.41.0	MIT
@humanwhocodes/config-array	0.11.10	Apache-2.0
@humanwhocodes/module-importer	1.0.1	Apache-2.0
@humanwhocodes/object-schema	1.2.1	BSD-3-Clause
@istanbuljs/load-nyc-config	1.1.0	ISC
@istanbuljs/schema	0.1.3	MIT
@jest/console	29.5.0	MIT
@jest/core	29.5.0	MIT
@jest/environment	29.5.0	MIT
@jest/expect-utils	29.5.0	MIT
@jest/expect	29.5.0	MIT
@jest/fake-timers	29.5.0	MIT
@jest/globals	29.5.0	MIT
@jest/reporters	29.5.0	MIT
@jest/schemas	29.4.3	MIT
@jest/source-map	29.4.3	MIT
@jest/test-result	29.5.0	MIT

Package	Version	License
@jest/test-sequencer	29.5.0	MIT
@jest/transform	29.5.0	MIT
@jest/types	29.5.0	MIT
@jridgewell/gen-mapping	0.3.3	MIT
@jridgewell/resolve-uri	3.1.0	MIT
@jridgewell/set-array	1.1.2	MIT
@jridgewell/source-map	0.3.4	MIT
@jridgewell/sourcemap-codec	1.4.14	MIT
@jridgewell/sourcemap-codec	1.4.15	MIT
@jridgewell/trace-mapping	0.3.18	MIT
@nodelib/fs.scandir	2.1.5	MIT
@nodelib/fs.stat	2.0.5	MIT
@nodelib/fs.walk	1.2.8	MIT
@oozcitak/dom	1.15.10	MIT
@oozcitak/infra	1.0.8	MIT
@oozcitak/url	1.0.4	MIT
@oozcitak/util	8.3.8	MIT
@sinclair/typebox	0.25.24	MIT
@sinonjs/commons	3.0.0	BSD-3-Clause
@sinonjs/fake-timers	10.3.0	BSD-3-Clause
@socket.io/component-emitter	3.1.0	MIT
@testing-library/dom	8.20.1	MIT
@testing-library/jest-dom	5.16.5	MIT

Package	Version	License
@testing-library/react-hooks	8.0.1	MIT
@testing-library/react	12.1.5	MIT
@tootallnate/once	2.0.0	MIT
@types/aria-query	5.0.1	MIT
@types/babel__core	7.20.1	MIT
@types/babel__generator	7.6.4	MIT
@types/babel__template	7.4.1	MIT
@types/babel__traverse	7.20.1	MIT
@types/eslint-scope	3.7.4	MIT
@types/eslint	8.40.2	MIT
@types/estree	1.0.1	MIT
@types/glob	7.2.0	MIT
@types/graceful-fs	4.1.6	MIT
@types/istanbul-lib-coverage	2.0.4	MIT
@types/istanbul-lib-report	3.0.0	MIT
@types/istanbul-reports	3.0.1	MIT
@types/jest	29.5.2	MIT
@types/jsdom	20.0.1	MIT
@types/json-schema	7.0.12	MIT
@types/json5	0.0.29	MIT
@types/minimatch	5.1.2	MIT
@types/node	20.3.3	MIT
@types/prettier	2.7.3	MIT

Package	Version	License
@types/prop-types	15.7.5	MIT
@types/react-dom	17.0.20	MIT
@types/react	17.0.62	MIT
@types/scheduler	0.16.3	MIT
@types/semver	7.5.0	MIT
@types/stack-utils	2.0.1	MIT
@types/testing-library__jest-dom	5.14.6	MIT
@types/tough-cookie	4.0.2	MIT
@types/yargs-parser	21.0.0	MIT
@types/yargs	17.0.24	MIT
@typescript-eslint/eslint-plugin	5.59.9	MIT
@typescript-eslint/parser	5.59.9	BSD-2-Clause
@typescript-eslint/scope-manager	5.59.9	MIT
@typescript-eslint/type-utils	5.59.9	MIT
@typescript-eslint/types	5.59.9	MIT
@typescript-eslint/typescript-estree	5.59.9	BSD-2-Clause
@typescript-eslint/utils	5.59.9	MIT
@typescript-eslint/visitor-keys	5.59.9	MIT
@webassemblyjs/ast	1.11.6	MIT
@webassemblyjs/floating-point-hex-parser	1.11.6	MIT
@webassemblyjs/helper-api-error	1.11.6	MIT
@webassemblyjs/helper-buffer	1.11.6	MIT
@webassemblyjs/helper-numbers	1.11.6	MIT

Package	Version	License
@webassemblyjs/helper-wasm-bytecode	1.11.6	MIT
@webassemblyjs/helper-wasm-section	1.11.6	MIT
@webassemblyjs/ieee754	1.11.6	MIT
@webassemblyjs/leb128	1.11.6	Apache-2.0
@webassemblyjs/utf8	1.11.6	MIT
@webassemblyjs/wasm-edit	1.11.6	MIT
@webassemblyjs/wasm-gen	1.11.6	MIT
@webassemblyjs/wasm-opt	1.11.6	MIT
@webassemblyjs/wasm-parser	1.11.6	MIT
@webassemblyjs/wast-printer	1.11.6	MIT
@webpack-cli/configtest	2.1.1	MIT
@webpack-cli/info	2.0.2	MIT
@webpack-cli/serve	2.0.5	MIT
@xtuc/ieee754	1.2.0	BSD-3-Clause
@xtuc/long	4.2.2	Apache-2.0
abab	2.0.6	BSD-3-Clause
acorn-globals	7.0.1	MIT
acorn-import-assertions	1.9.0	MIT
acorn-jsx	5.3.2	MIT
acorn-walk	8.2.0	MIT
acorn	8.9.0	MIT
agent-base	6.0.2	MIT
ajv-formats	2.1.1	MIT

Package	Version	License
ajv-keywords	3.5.2	MIT
ajv-keywords	5.1.0	MIT
ajv	6.12.6	MIT
ajv	8.12.0	MIT
ansi-escapes	4.3.2	MIT
ansi-regex	5.0.1	MIT
ansi-styles	3.2.1	MIT
ansi-styles	4.3.0	MIT
ansi-styles	5.2.0	MIT
anymatch	3.1.3	ISC
archiver-utils	2.1.0	MIT
archiver	5.3.1	MIT
argparse	1.0.10	MIT
argparse	2.0.1	Python-2.0
aria-query	5.1.3	Apache-2.0
aria-query	5.3.0	Apache-2.0
array-buffer-byte-length	1.0.0	MIT
array-includes	3.1.6	MIT
array-union	1.0.2	MIT
array-union	2.1.0	MIT
array-uniq	1.0.3	MIT
array.prototype.flat	1.3.1	MIT
array.prototype.flatmap	1.3.1	MIT

Package	Version	License
array.prototype.tosorted	1.1.1	MIT
ast-types-flow	0.0.7	ISC
async	3.2.4	MIT
asynckit	0.4.0	MIT
available-typed-arrays	1.0.5	MIT
axe-core	4.7.2	MPL-2.0
axobject-query	3.2.1	Apache-2.0
babel-jest	29.5.0	MIT
babel-loader	9.1.2	MIT
babel-plugin-istanbul	6.1.1	BSD-3-Clause
babel-plugin-jest-hoist	29.5.0	MIT
babel-plugin-polyfill-corejs2	0.4.3	MIT
babel-plugin-polyfill-corejs3	0.8.1	MIT
babel-plugin-polyfill-regenerator	0.5.0	MIT
babel-plugin-transform-react-remove-prop-types	0.4.24	MIT
babel-preset-current-node-syntax	1.0.1	MIT
babel-preset-jest	29.5.0	MIT
balanced-match	1.0.2	MIT
base64-js	1.5.1	MIT
bl	4.1.0	MIT
brace-expansion	1.1.11	MIT
brace-expansion	2.0.1	MIT
braces	3.0.2	MIT

Package	Version	License
browserslist	4.21.9	MIT
bs-logger	0.2.6	MIT
bser	2.1.1	Apache-2.0
buffer-crc32	0.2.13	MIT
buffer-from	1.1.2	MIT
buffer	5.7.1	MIT
call-bind	1.0.2	MIT
callsites	3.1.0	MIT
camelcase	5.3.1	MIT
camelcase	6.3.0	MIT
caniuse-lite	1.0.30001511	CC-BY-4.0
chalk	2.4.2	MIT
chalk	3.0.0	MIT
chalk	4.1.2	MIT
char-regex	1.0.2	MIT
chownr	2.0.0	ISC
chrome-trace-event	1.0.3	MIT
ci-info	3.8.0	MIT
cjs-module-lexer	1.2.3	MIT
clean-webpack-plugin	4.0.0	MIT
cliui	8.0.1	ISC
clone-deep	4.0.1	MIT
co	4.6.0	MIT

Package	Version	License
collect-v8-coverage	1.0.1	MIT
color-convert	1.9.3	MIT
color-convert	2.0.1	MIT
color-name	1.1.3	MIT
color-name	1.1.4	MIT
colorette	2.0.20	MIT
combined-stream	1.0.8	MIT
commander	10.0.1	MIT
commander	2.20.3	MIT
comondir	1.0.1	MIT
compress-commons	4.1.1	MIT
concat-map	0.0.1	MIT
convert-source-map	1.9.0	MIT
convert-source-map	2.0.0	MIT
copy-webpack-plugin	11.0.0	MIT
core-js-compat	3.31.0	MIT
core-js	2.6.12	MIT
core-util-is	1.0.3	MIT
crc-32	1.2.2	Apache-2.0
crc32-stream	4.0.2	MIT
cross-spawn	7.0.3	MIT
css-loader	6.8.1	MIT
css.escape	1.5.1	MIT

Package	Version	License
cssesc	3.0.0	MIT
cssom	0.3.8	MIT
cssom	0.5.0	MIT
cssstyle	2.3.0	MIT
csstype	3.1.2	MIT
damerau-levenshtein	1.0.8	BSD-2-Clause
data-urls	3.0.2	MIT
debug	3.2.7	MIT
debug	4.3.4	MIT
decimal.js	10.4.3	MIT
dedent	0.7.0	MIT
deep-equal	2.2.1	MIT
deep-is	0.1.4	MIT
deepmerge	4.3.1	MIT
define-properties	1.2.0	MIT
del	4.1.1	MIT
delayed-stream	1.0.0	MIT
dequal	2.0.3	MIT
detect-newline	3.1.0	MIT
diff-sequences	29.4.3	MIT
dir-glob	3.0.1	MIT
doctrine	2.1.0	Apache-2.0
doctrine	3.0.0	Apache-2.0

Package	Version	License
dom-accessibility-api	0.5.16	MIT
domexception	4.0.0	MIT
electron-to-chromium	1.4.447	ISC
emittery	0.13.1	MIT
emoji-regex	8.0.0	MIT
emoji-regex	9.2.2	MIT
end-of-stream	1.4.4	MIT
engine.io-client	6.5.1	MIT
engine.io-parser	5.1.0	MIT
enhanced-resolve	5.15.0	MIT
entities	4.5.0	BSD-2-Clause
envinfo	7.10.0	MIT
error-ex	1.3.2	MIT
es-abstract	1.21.2	MIT
es-get-iterator	1.1.3	MIT
es-module-lexer	1.3.0	MIT
es-set-tostringtag	2.0.1	MIT
es-shim-unscopables	1.0.0	MIT
es-to-primitive	1.2.1	MIT
escalade	3.1.1	MIT
escape-string-regexp	1.0.5	MIT
escape-string-regexp	2.0.0	MIT
escape-string-regexp	4.0.0	MIT

Package	Version	License
escodegen	2.1.0	BSD-2-Clause
eslint-config-prettier	8.8.0	MIT
eslint-import-resolver-node	0.3.7	MIT
eslint-module-utils	2.8.0	MIT
eslint-plugin-import	2.27.5	MIT
eslint-plugin-jsx-a11y	6.7.1	MIT
eslint-plugin-react-hooks	4.6.0	MIT
eslint-plugin-react	7.32.2	MIT
eslint-scope	5.1.1	BSD-2-Clause
eslint-scope	7.2.0	BSD-2-Clause
eslint-visitor-keys	3.4.1	Apache-2.0
eslint	8.41.0	MIT
espre	9.6.0	BSD-2-Clause
esprima	4.0.1	BSD-2-Clause
esquery	1.5.0	BSD-3-Clause
esrecurse	4.3.0	BSD-2-Clause
estaverse	4.3.0	BSD-2-Clause
estaverse	5.3.0	BSD-2-Clause
esutils	2.0.3	BSD-2-Clause
events	3.3.0	MIT
execa	5.1.1	MIT
exit	0.1.2	MIT
expect	29.5.0	MIT

Package	Version	License
fast-deep-equal	3.1.3	MIT
fast-glob	3.3.0	MIT
fast-json-stable-stringify	2.1.0	MIT
fast-levenshtein	2.0.6	MIT
fastest-levenshtein	1.0.16	MIT
fastq	1.15.0	ISC
fb-watchman	2.0.2	Apache-2.0
file-entry-cache	6.0.1	MIT
fill-range	7.0.1	MIT
find-cache-dir	3.3.2	MIT
find-up	4.1.0	MIT
find-up	5.0.0	MIT
flat-cache	3.0.4	MIT
flatted	3.2.7	ISC
fliX-ps-uxp-plugin	1.0.0	Custom: https://github.com/adobe-uxp/devtools-cli
for-each	0.3.3	MIT
form-data	4.0.0	MIT
fs-constants	1.0.0	MIT
fs-minipass	2.1.0	ISC
fs.realpath	1.0.0	ISC
function-bind	1.1.1	MIT

Package	Version	License
function.prototype.name	1.1.5	MIT
functions-have-names	1.2.3	MIT
gensync	1.0.0-beta.2	MIT
get-caller-file	2.0.5	ISC
get-intrinsic	1.2.1	MIT
get-package-type	0.1.0	MIT
get-stream	6.0.1	MIT
get-symbol-description	1.0.0	MIT
glob-parent	5.1.2	ISC
glob-parent	6.0.2	ISC
glob-to-regexp	0.4.1	BSD-2-Clause
glob	7.2.3	ISC
globals	11.12.0	MIT
globals	13.20.0	MIT
globalthis	1.0.3	MIT
globby	11.1.0	MIT
globby	13.2.1	MIT
globby	6.1.0	MIT
gopd	1.0.1	MIT
graceful-fs	4.2.11	ISC
grapheme-splitter	1.0.4	MIT
graphemer	1.4.0	MIT
harmony-reflect	1.6.2	(Apache-2.0 OR MPL-1.1)

Package	Version	License
has-bigints	1.0.2	MIT
has-flag	3.0.0	MIT
has-flag	4.0.0	MIT
has-property-descriptors	1.0.0	MIT
has-proto	1.0.1	MIT
has-symbols	1.0.3	MIT
has-tostringtag	1.0.0	MIT
has	1.0.3	MIT
html-encoding-sniffer	3.0.0	MIT
html-escaper	2.0.2	MIT
http-proxy-agent	5.0.0	MIT
https-proxy-agent	5.0.1	MIT
human-signals	2.1.0	Apache-2.0
iconv-lite	0.6.3	MIT
icss-utils	5.1.0	ISC
identity-obj-proxy	3.0.0	MIT
ieee754	1.2.1	BSD-3-Clause
ignore	5.2.4	MIT
import-fresh	3.3.0	MIT
import-local	3.1.0	MIT
imurmurhash	0.1.4	MIT
indent-string	4.0.0	MIT
inflight	1.0.6	ISC

Package	Version	License
inherits	2.0.4	ISC
internal-slot	1.0.5	MIT
interpret	3.1.1	MIT
is-arguments	1.1.1	MIT
is-array-buffer	3.0.2	MIT
is-arrayish	0.2.1	MIT
is-bigint	1.0.4	MIT
is-boolean-object	1.1.2	MIT
is-callable	1.2.7	MIT
is-core-module	2.12.1	MIT
is-date-object	1.0.5	MIT
is-extglob	2.1.1	MIT
is-fullwidth-code-point	3.0.0	MIT
is-generator-fn	2.1.0	MIT
is-glob	4.0.3	MIT
is-map	2.0.2	MIT
is-negative-zero	2.0.2	MIT
is-number-object	1.0.7	MIT
is-number	7.0.0	MIT
is-path-cwd	2.2.0	MIT
is-path-in-cwd	2.1.0	MIT
is-path-inside	2.1.0	MIT
is-path-inside	3.0.3	MIT

Package	Version	License
is-plain-object	2.0.4	MIT
is-potential-custom-element-name	1.0.1	MIT
is-regex	1.1.4	MIT
is-set	2.0.2	MIT
is-shared-array-buffer	1.0.2	MIT
is-stream	2.0.1	MIT
is-string	1.0.7	MIT
is-symbol	1.0.4	MIT
is-typed-array	1.1.10	MIT
is-weakmap	2.0.1	MIT
is-weakref	1.0.2	MIT
is-weakset	2.0.2	MIT
isarray	1.0.0	MIT
isarray	2.0.5	MIT
isexe	2.0.0	ISC
isobject	3.0.1	MIT
istanbul-lib-coverage	3.2.0	BSD-3-Clause
istanbul-lib-instrument	5.2.1	BSD-3-Clause
istanbul-lib-report	3.0.0	BSD-3-Clause
istanbul-lib-source-maps	4.0.1	BSD-3-Clause
istanbul-reports	3.1.5	BSD-3-Clause
jest-changed-files	29.5.0	MIT
jest-circus	29.5.0	MIT

Package	Version	License
jest-cli	29.5.0	MIT
jest-config	29.5.0	MIT
jest-diff	29.5.0	MIT
jest-docblock	29.4.3	MIT
jest-each	29.5.0	MIT
jest-environment-jsdom	29.5.0	MIT
jest-environment-node	29.5.0	MIT
jest-get-type	29.4.3	MIT
jest-haste-map	29.5.0	MIT
jest-leak-detector	29.5.0	MIT
jest-matcher-utils	29.5.0	MIT
jest-message-util	29.5.0	MIT
jest-mock	29.5.0	MIT
jest-pnp-resolver	1.2.3	MIT
jest-regex-util	29.4.3	MIT
jest-resolve-dependencies	29.5.0	MIT
jest-resolve	29.5.0	MIT
jest-runner	29.5.0	MIT
jest-runtime	29.5.0	MIT
jest-snapshot	29.5.0	MIT
jest-util	29.5.0	MIT
jest-validate	29.5.0	MIT
jest-watcher	29.5.0	MIT

Package	Version	License
jest-worker	27.5.1	MIT
jest-worker	29.5.0	MIT
jest	29.5.0	MIT
js-tokens	4.0.0	MIT
js-yaml	3.14.1	MIT
js-yaml	4.1.0	MIT
jsdom	20.0.3	MIT
jsesc	0.5.0	MIT
jsesc	2.5.2	MIT
json-parse-even-better-errors	2.3.1	MIT
json-schema-traverse	0.4.1	MIT
json-schema-traverse	1.0.0	MIT
json-stable-stringify-without-jsonify	1.0.1	MIT
json5	1.0.2	MIT
json5	2.2.3	MIT
jsx-ast-utils	3.3.4	MIT
kind-of	6.0.3	MIT
kleur	3.0.3	MIT
language-subtag-registry	0.3.22	CC0-1.0
language-tags	1.0.5	MIT
lazystream	1.0.1	MIT
leven	3.1.0	MIT
levn	0.4.1	MIT

Package	Version	License
lines-and-columns	1.2.4	MIT
loader-runner	4.3.0	MIT
locate-path	5.0.0	MIT
locate-path	6.0.0	MIT
lodash.debounce	4.0.8	MIT
lodash.defaults	4.2.0	MIT
lodash.difference	4.5.0	MIT
lodash.flatten	4.4.0	MIT
lodash.isplainobject	4.0.6	MIT
lodash.memoize	4.1.2	MIT
lodash.merge	4.6.2	MIT
lodash.union	4.6.0	MIT
lodash	4.17.21	MIT
loose-envify	1.4.0	MIT
lru-cache	5.1.1	ISC
lru-cache	6.0.0	ISC
lz-string	1.5.0	MIT
make-dir	3.1.0	MIT
make-error	1.3.6	ISC
makeerror	1.0.12	BSD-3-Clause
merge-stream	2.0.0	MIT
merge2	1.4.1	MIT
micromatch	4.0.5	MIT

Package	Version	License
mime-db	1.52.0	MIT
mime-types	2.1.35	MIT
mimic-fn	2.1.0	MIT
min-indent	1.0.1	MIT
minimatch	3.1.2	ISC
minimatch	5.1.6	ISC
minimist	1.2.8	MIT
minipass	3.3.6	ISC
minipass	5.0.0	ISC
minizlib	2.1.2	MIT
mkdirp	1.0.4	MIT
ms	2.1.2	MIT
nanoid	3.3.6	MIT
natural-compare-lite	1.4.0	MIT
natural-compare	1.4.0	MIT
neo-async	2.6.2	MIT
node-int64	0.4.0	MIT
node-releases	2.0.12	MIT
normalize-path	3.0.0	MIT
npm-run-path	4.0.1	MIT
nwsapi	2.2.5	MIT
object-assign	4.1.1	MIT
object-inspect	1.12.3	MIT

Package	Version	License
object-is	1.1.5	MIT
object-keys	1.1.1	MIT
object.assign	4.1.4	MIT
object.entries	1.1.6	MIT
object.fromentries	2.0.6	MIT
object.hasown	1.1.2	MIT
object.values	1.1.6	MIT
once	1.4.0	ISC
onetime	5.1.2	MIT
optionator	0.9.3	MIT
p-limit	2.3.0	MIT
p-limit	3.1.0	MIT
p-locate	4.1.0	MIT
p-locate	5.0.0	MIT
p-map	2.1.0	MIT
p-try	2.2.0	MIT
parent-module	1.0.1	MIT
parse-json	5.2.0	MIT
parse5	7.1.2	MIT
path-exists	4.0.0	MIT
path-is-absolute	1.0.1	MIT
path-is-inside	1.0.2	(WTFPL OR MIT)
path-key	3.1.1	MIT

Package	Version	License
path-parse	1.0.7	MIT
path-type	4.0.0	MIT
picocolors	1.0.0	ISC
picomatch	2.3.1	MIT
pify	2.3.0	MIT
pify	4.0.1	MIT
pinkie-promise	2.0.1	MIT
pinkie	2.0.4	MIT
pirates	4.0.6	MIT
pkg-dir	4.2.0	MIT
postcss-modules-extract-imports	3.0.0	ISC
postcss-modules-local-by-default	4.0.3	MIT
postcss-modules-scope	3.0.0	ISC
postcss-modules-values	4.0.0	ISC
postcss-selector-parser	6.0.13	MIT
postcss-value-parser	4.2.0	MIT
postcss	8.4.24	MIT
prelude-ls	1.2.1	MIT
prettier	2.8.8	MIT
pretty-format	27.5.1	MIT
pretty-format	29.5.0	MIT
process-nextick-args	2.0.1	MIT
prompts	2.4.2	MIT

Package	Version	License
prop-types	15.8.1	MIT
psl	1.9.0	MIT
punycode	2.3.0	MIT
pure-rand	6.0.2	MIT
querystringify	2.2.0	MIT
queue-microtask	1.2.3	MIT
randombytes	2.1.0	MIT
react-dom	17.0.2	MIT
react-error-boundary	3.1.4	MIT
react-is	16.13.1	MIT
react-is	17.0.2	MIT
react-is	18.2.0	MIT
react	17.0.2	MIT
readable-stream	2.3.8	MIT
readable-stream	3.6.2	MIT
readdir-glob	1.1.3	Apache-2.0
rechoir	0.8.0	MIT
redent	3.0.0	MIT
regenerate-unicode-properties	10.1.0	MIT
regenerate	1.4.2	MIT
regenerator-runtime	0.13.11	MIT
regenerator-transform	0.15.1	MIT
regexp.prototype.flags	1.5.0	MIT

Package	Version	License
regexpu-core	5.3.2	MIT
regjsparser	0.9.1	BSD-2-Clause
require-directory	2.1.1	MIT
require-from-string	2.0.2	MIT
requires-port	1.0.0	MIT
resolve-cwd	3.0.0	MIT
resolve-from	4.0.0	MIT
resolve-from	5.0.0	MIT
resolve.exports	2.0.2	MIT
resolve	1.22.2	MIT
resolve	2.0.0-next.4	MIT
reusify	1.0.4	MIT
rimraf	2.7.1	ISC
rimraf	3.0.2	ISC
run-parallel	1.2.0	MIT
rxjs	7.8.1	Apache-2.0
safe-buffer	5.1.2	MIT
safe-buffer	5.2.1	MIT
safe-regex-test	1.0.0	MIT
safer-buffer	2.1.2	MIT
saxes	6.0.0	ISC
scheduler	0.20.2	MIT
schema-utils	3.3.0	MIT

Package	Version	License
schema-utils	4.2.0	MIT
semver	6.3.0	ISC
semver	7.5.3	ISC
serialize-javascript	6.0.1	BSD-3-Clause
shallow-clone	3.0.1	MIT
shebang-command	2.0.0	MIT
shebang-regex	3.0.0	MIT
side-channel	1.0.4	MIT
signal-exit	3.0.7	ISC
sisteransi	1.0.5	MIT
slash	3.0.0	MIT
slash	4.0.0	MIT
socket.io-client	4.7.1	MIT
socket.io-parser	4.2.4	MIT
source-map-js	1.0.2	BSD-3-Clause
source-map-loader	4.0.1	MIT
source-map-support	0.5.13	MIT
source-map-support	0.5.21	MIT
source-map	0.6.1	BSD-3-Clause
sprintf-js	1.0.3	BSD-3-Clause
stack-utils	2.0.6	MIT
stop-iteration-iterator	1.0.0	MIT
string-length	4.0.2	MIT

Package	Version	License
string-width	4.2.3	MIT
string.prototype.matchall	4.0.8	MIT
string.prototype.trim	1.2.7	MIT
string.prototype.trimend	1.0.6	MIT
string.prototype.trimstart	1.0.6	MIT
string_decoder	1.1.1	MIT
string_decoder	1.3.0	MIT
strip-ansi	6.0.1	MIT
strip-bom	3.0.0	MIT
strip-bom	4.0.0	MIT
strip-final-newline	2.0.0	MIT
strip-indent	3.0.0	MIT
strip-json-comments	3.1.1	MIT
style-loader	3.3.3	MIT
supports-color	5.5.0	MIT
supports-color	7.2.0	MIT
supports-color	8.1.1	MIT
supports-preserve-symlinks-flag	1.0.0	MIT
symbol-tree	3.2.4	MIT
tapable	2.2.1	MIT
tar-stream	2.2.0	MIT
tar	6.2.0	ISC
terser-webpack-plugin	5.3.9	MIT

Package	Version	License
terser	5.18.2	BSD-2-Clause
test-exclude	6.0.0	ISC
text-table	0.2.0	MIT
tmpl	1.0.5	BSD-3-Clause
to-fast-properties	2.0.0	MIT
to-regex-range	5.0.1	MIT
tough-cookie	4.1.3	BSD-3-Clause
tr46	3.0.0	MIT
ts-jest	29.1.0	MIT
tsconfig-paths	3.14.2	MIT
tslib	1.14.1	0BSD
tslib	2.6.0	0BSD
tsutils	3.21.0	MIT
type-check	0.4.0	MIT
type-detect	4.0.8	MIT
type-fest	0.20.2	(MIT OR CC0-1.0)
type-fest	0.21.3	(MIT OR CC0-1.0)
typed-array-length	1.0.4	MIT
typescript	5.0.4	Apache-2.0
unbox-primitive	1.0.2	MIT
unicode-canonical-property-names-ecmascript	2.0.0	MIT
unicode-match-property-ecmascript	2.0.0	MIT
unicode-match-property-value-ecmascript	2.1.0	MIT

Package	Version	License
unicode-property-aliases-ecmascript	2.1.0	MIT
universalify	0.2.0	MIT
update-browserslist-db	1.0.11	MIT
uri-js	4.4.1	BSD-2-Clause
url-parse	1.5.10	MIT
util-deprecate	1.0.2	MIT
uxp-types	0.1.3	MIT
v8-to-istanbul	9.1.0	ISC
w3c-xmlserializer	4.0.0	MIT
walker	1.0.8	Apache-2.0
watchpack	2.4.0	MIT
webidl-conversions	7.0.0	BSD-2-Clause
webpack-cli	5.1.1	MIT
webpack-merge	5.9.0	MIT
webpack-sources	3.2.3	MIT
webpack	5.84.1	MIT
whatwg-encoding	2.0.0	MIT
whatwg-mimetype	3.0.0	MIT
whatwg-url	11.0.0	MIT
which-boxed-primitive	1.0.2	MIT
which-collection	1.0.1	MIT
which-typed-array	1.1.9	MIT
which	2.0.2	ISC

Package	Version	License
wildcard	2.0.1	MIT
wrap-ansi	7.0.0	MIT
wrappy	1.0.2	ISC
write-file-atomic	4.0.2	ISC
ws	8.11.0	MIT
ws	8.13.0	MIT
xml-name-validator	4.0.0	Apache-2.0
xmlbuilder2	3.1.1	MIT
xmlchars	2.2.0	MIT
xmlhttprequest-ssl	2.0.0	MIT
y18n	5.0.8	ISC
yallist	3.1.1	ISC
yallist	4.0.0	ISC
yargs-parser	21.1.1	ISC
yargs	17.7.2	MIT
yocto-queue	0.1.0	MIT
zip-a-folder	2.0.0	MIT
zip-stream	4.1.0	MIT

Maya

Name	Version	License
PySide2-stubs	5.15.2.1.2	GNU Lesser General Public License v2 (LGPLv2)
PyYAML	6.0.1	MIT License

Name	Version	License
aiohttp	3.9.1	Apache Software License
aiosignal	1.3.1	Apache Software License
anyio	3.7.1	MIT License
appdirs	1.4.4	MIT License
async-timeout	4.0.3	Apache Software License
asyncclick	8.1.3.4	BSD License
attrs	23.2.0	MIT License
bidict	0.22.1	Mozilla Public License 2.0 (MPL 2.0)
certifi	2023.11.17	Mozilla Public License 2.0 (MPL 2.0)
cffib	1.16.0	MIT License
click	8.1.7	BSD License
cryptography	41.0.7	Apache Software License; BSD License
exceptiongroup	1.2.0	MIT License
flix-sdk	1.1.0	Apache Software License
frozenlist	1.4.1	Apache Software License
grpcio	1.60.0	Apache Software License
h11	0.14.0	MIT License
httpcore	0.17.3	BSD License
httpx	0.24.1	BSD License
idna	3.6	BSD License
lxml	4.9.4	BSD License
lxml-stubs	0.4.0	Apache Software License
maya-stubs	0.4.0	MIT License

Name	Version	License
multidict	6.0.4	Apache Software License
mypy	1.8.0	MIT License
mypy-extensions	1.0.0	MIT License
nodeenv	1.8.0	BSD License
packaging	23.2	Apache Software License; BSD License
protobuf	4.25.1	3-Clause BSD License
pycparser	2.21	BSD License
pyright	1.1.344	MIT
python-dateutil	2.8.2	Apache Software License; BSD License
python-engineio	4.8.1	MIT License
python-socketio	5.10.0	MIT License
semver	3.0.2	BSD License
simple-websocket	1.0.0	MIT License
six	1.16.0	MIT License
sniffio	1.3.0	Apache Software License; MIT License
tomli	2.0.1	MIT License
types-PyYAML	6.0.12.12	Apache Software License
typing_extensions	4.9.0	Python Software Foundation License
wsproto	1.2.0	MIT License
yaml	1.9.4	Apache Software License

Third-Party Licenses

Click [Flix Third-Party Licenses](#) to download third-party licenses.

End User License Agreement (EULA)

PLEASE READ THIS EULA CAREFULLY BEFORE ORDERING OR DOWNLOADING OR USING ANY SOFTWARE PRODUCTS OF FOUNDRY. YOUR ATTENTION IS PARTICULARLY DRAWN TO: (A) CLAUSE 8 IN WHICH SUBSCRIPTION CUSTOMERS AGREE TO THE AUTO-RENEWAL OF THEIR LICENSE ON AN ANNUAL BASIS; (B) CLAUSES 15 AND 16 WHERE WE LIMIT OUR LIABILITY TO USERS OF OUR SOFTWARE PRODUCTS; (C) CLAUSE 19.2 REGARDING THE DATA WE MAY COLLECT AND HOW WE MAY USE IT; AND (D) CLAUSE 19.3 WHERE YOU AUTHORISE FOUNDRY TO USE THE SOFTWARE TO ACCESS AND COLLECT CERTAIN INFORMATION FROM YOUR COMPUTER NETWORKS AND TO TRANSMIT THIS INFORMATION TO FOUNDRY.

IMPORTANT NOTICE TO ALL USERS: BY DOWNLOADING AND/OR USING THIS SOFTWARE YOU ACKNOWLEDGE THAT YOU HAVE READ THIS EULA, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA DO NOT DOWNLOAD, INSTALL, COPY OR USE THE SOFTWARE.

IMPORTANT NOTICE TO CONSUMERS WHO PURCHASE SOFTWARE PRODUCTS DIRECT FROM FOUNDRY: YOU HAVE THE RIGHT TO CANCEL YOUR CONTRACT AND OBTAIN A FULL REFUND IN ACCORDANCE WITH CLAUSE 9. HOWEVER YOU WILL LOSE THIS RIGHT ONCE YOU INSTALL THE SOFTWARE OR LOGIN TO AN INDIVIDUAL LOGIN OR TEAM LOGIN LICENSE. THIS DOES NOT AFFECT YOUR CONSUMER RIGHTS IN RELATION TO DEFECTIVE PRODUCTS OR SERVICES.

This END USER LICENSE AGREEMENT ("**EULA**") is, in cases where you purchase our product(s) direct from Foundry, incorporated into the agreement between The Foundry Visionmongers Ltd a company registered in England and Wales with company number 4642027 and whose registered office is at Squire Patton Boggs Secretarial Services Limited, Rutland House, 148 Edmund Street, Birmingham, United Kingdom, B3 2JR and whose address for correspondence is 5 Golden Square, London W1F 9HT, ("**Foundry**"), and you, as either an individual or a single company or other legal entity ("**Licensee**") on the terms of which you will purchase the products and services of Foundry (the "**Agreement**"). In cases where you purchase our product(s) from one of our resellers, the use of the term "Agreement" in this EULA refers to the arrangements between Foundry and Licensee on which Licensee is permitted to use Foundry's product(s), including this EULA.

Foundry reserves the right to refuse to grant a License (as defined in clause 1.1) to any Licensee who has failed to pay any sum due either to Foundry or to a reseller of Foundry, in connection with the Agreement, in connection with any other software license, in connection with any other agreed terms between Foundry and the Licensee, to use any Software product(s) of Foundry and/or in connection with any Maintenance and Support Agreement as defined in clause 11.1.

1. GRANT OF LICENSE

1.1 Subject to terms and the scope of the applicable licence model as set out in clause 2, the limitations of clause 3 and all the other terms of the Agreement, Foundry grants to Licensee a limited, non-transferable (subject to clause 2.1(b) below) and non-exclusive license to download, install and use a machine readable, object code version (subject to clauses 3 and 4 below) of the software program(s) purchased by Licensee (the “**Software**”) together with, and in accordance with, any accompanying user guide and other documentation (the “**Documentation**”), solely for Licensee’s, where the Licensee is a business, own internal purposes or, where the Licensee is a consumer, domestic and private purposes (the “**License**”); provided, however, that Licensee’s right to download, install and use the Software and the Documentation is limited to those rights expressly set out in this EULA.

1.2 Some types of license models set out in clause 2.1 limit the installation and use of the Software to the country in which Licensee is based at the date of purchase (the “**Home Country**”), unless otherwise agreed by Foundry in writing. For the avoidance of doubt, ‘use’ in a Home Country requires the server, network and user to be based in the Home Country. Notwithstanding such limits, Licensee may still use the Software outside the Home Country if traveling or working outside the Home Country on a temporary basis provided that: a) such use does not exceed 70 days in aggregate in any rolling twelve month period or, in the case of any license which lasts for less than twelve months, does not exceed the number of days representing 20% of the term of the license, and b) our Software may not be accessed or used in countries which are subject to import/export controls or sanctions, as further detailed in Clause 22.

1.3 Only to the extent that is proportionate to, and reasonably necessary to support, Licensee’s licensed use of the Software in accordance with the Agreement, Licensee may (provided valid license keys or license entitlements have been obtained) install the Software on more than one computer, provided always that Licensee’s concurrent use of different installations of the Software does not exceed the number of valid Licenses that Licensee has paid for or licensed (as applicable).

2. LICENSE MODELS

2.1 For each Software product that you purchase from Foundry, the product will be licensed (and not sold) to you on either a rental or subscription basis in accordance with one or more of the license models set out in this clause 2.1 and clause 2.2, as specified in Foundry’s invoice or order confirmation (as applicable), and subject to the other terms and conditions of this EULA. Please note that some licensing models set out below do not apply to certain Software products of Foundry. Whichever licensing model applies, Licensee shall not at any one time use more copies of the Software than the total number of valid licenses purchased by Licensee.

(a) “**Offline Node Locked License**”

If Licensee purchases an Offline Node Locked License, Licensee will install and use only a single copy of the Software on only one computer at a time in the Home Country.

(b) **"Modo Individual License"**

If Licensee has a Modo Individual License then: (a) Licensee warrants and represents that Licensee is a natural person and that only Licensee will use the Software; (b) Licensee may transfer or assign ("**transfer**") the Modo Individual License to another natural person ("**Assignee**") subject to Licensee: (i) notifying Foundry of such transfer and obtaining Foundry's express written consent, (ii) paying an administrative fee with respect to such transfer as may be required by Foundry, and (iii) after transferring a single copy of the Software to the Assignee, deleting any copies of the Software that Licensee may have in Licensee's possession, custody or power; (c) Licensee shall not share its login details for the Software with any third party (d) Licensee shall be entitled to use the Software on different computers which may be located anywhere and use is not limited to the Home Country; (e) use of the Software shall be limited to no more than one concurrent use at all times.

(c) **"Offline Floating License"**

If Licensee purchases an Offline Floating License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Offline Floating Licenses purchased by Licensee; and (b) use of the Software shall be limited to any site in the Home Country.

(d) **"Individual Login License"**

If Licensee purchases an Individual Login License, Licensee warrants and represents that Licensee is a natural person and that only Licensee shall use the Software. Licensee will be issued with log in details and may use the Software on any number of computers (but not simultaneously). Use of the Software shall be limited to any site in the Home Country.

(e) **"Team Login License"**

If Licensee purchases a Team Login License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent users shall never exceed the total number of valid Team Login Licenses purchased by Licensee; (b) use of the Software shall be limited to any site in the Home Country; and (c) use of the Software, and changes to the user authorizations within Licensee's team organization(s), shall be in accordance with any Team Login License guidance and instructions, as published on Foundry's website and which may be amended from time to time.

(f) **"Nuke Indie License"**

If Licensee purchases a License for Nuke Indie, then: (a) Licensee warrants and represents that Licensee (i) is a natural person and that only Licensee will use the Software; (ii) is working independently and shall not use the Nuke Indie License in a pipeline with other Nuke commercial or Nuke Indie licenses, whether those licenses are held by the Licensee, other individuals or other businesses or organisations; and (iii) earns less

than \$100,000 USD (or local equivalent) a year; and (iv) that Licensee satisfies all criteria set out in Foundry's Nuke Indie Eligibility Requirements as published on its website and which may be amended from time to time (the "**Nuke Indie Eligibility Requirements**"); (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee shall not purchase or use more than one Nuke Indie License; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; and (e) Licensee shall use the Software in accordance with terms of the Nuke Indie Eligibility Requirements, including abiding by any functional restrictions; and (f) the provisions of clause 8 shall apply.

(g) "**Modo Subscription License**"

If Licensee has a Modo Subscription License, then: (a) Licensee warrants and represents that Licensee is a natural person and that only Licensee will use the Software; (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee may use the Software on different computers which may be located anywhere and use is not restricted to the Home Country; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; (e) Licensee shall not purchase or use more than one Modo Subscription License; and (f) the provisions of clause 8 shall apply.

(h) "**Mari Individual Subscription License**"

If Licensee purchases a Mari Individual Subscription License then: (a) Licensee warrants and represents that Licensee is a natural person and that only Licensee will use the Software; (b) Licensee shall not share its login details for the Software with any third party; (c) Licensee may use the Software on different computers which may be located anywhere and use is not restricted to the Home Country; (d) Licensee may use the License on different computers, subject to (i) a maximum of two computer authorisations at any one time and (ii) no more than one concurrent user at any one time; (e) Licensee shall not purchase or use more than one Mari Individual Subscription License; and (f) the provisions of clause 8 shall apply.

(i) "**Rental License**"

If Licensee has purchased the Software on a rental basis, the License shall be limited to the term of the rental as agreed in writing with Foundry after which it shall automatically expire.

(j) "**Educational License**"

If Licensee has purchased the Software on the discounted terms of Foundry's educational program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Educational License that: (i) it is an educational institution that will use the Software only for the purpose of training and instruction, and for no other purpose, (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Educational License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Educational License is a Floating License, Licensee shall use the Software on only one computer at a time.

(k) "Graduate License"

If Licensee has purchased the Software on the discounted terms of Foundry's Graduate Licence, Licensee warrants and represents to Foundry as a condition of the Graduate Licence that the Licensee has graduated from a university or tertiary education institute no more than six (6) months prior to the grant of the Graduate Licence. The Licensee may use the Software in a personal capacity only (namely, as a sole trader or freelancer). For the avoidance of doubt, Licensee may not use the Graduate License in connection with any work or employment, whether paid or unpaid, or other such collaborations with studios or similar organisations. Foundry reserves the right to require evidence of graduation by the Licensee. Use of the Software shall be limited to any site in the Home Country.

(l) "Student License"

If Licensee has purchased the Software on the discounted terms of Foundry's student program and eligibility criteria as published on its website and amended from time to time, Licensee warrants and represents to Foundry as a condition of the Student License that: (i) the Licensee is a part-time or full-time student at the time of purchase and will not use the Software for any commercial, professional or for-profit purposes; (ii) Licensee will at all times comply with any eligibility criteria and guidelines in respect of the Student License, as may be communicated to a Licensee and/or published on Foundry's website and amended from time to time, and (iii) use of the Software shall be limited to any site in the Home Country. Unless the Student License is a Floating License, Licensee shall use the Software on only one computer at a time.

(m) "Non-Commercial License"

If the License is a Non-Commercial License, Licensee warrants and represents that Licensee is a natural person, that they will only access and/or use one copy of a Non-Commercial License for personal, recreational and non-commercial purposes and that only Licensee will use the Software. Under a Non-Commercial License, Licensee will not use the Software: (a) in conjunction with any other copies or versions of the Software, under any type of License model; (b) for any commercial, professional, for-profit and/or on-sale purpose or otherwise to provide any commercial service(s) to a third party (whether or not for financial or other reward and including for education, instruction of or demonstration to any third party for commercial purposes); (c) in the course of any employment or business undertaking of Licensee; (d) on any commercial premises during business hours (except where use of the Software is solely for a personal, recreational, educational or other non-commercial purpose); and/or (e) to create any commercial tools or plug ins.

(n) "Mari indie"

Variants of Mari with limited functionality as described in the Documentation are available to purchase. If Licensee has purchased such a variant, Licensee warrants and represents to Foundry as a condition of the Agreement that: (i) Licensee is a natural person; or (ii) Licensee is an entity in the direct ownership of a single natural person; (iii) Licensee will only access and/or use one copy of the Software; and (iv) only Licensee will use the Software.

(o) "Trial License"

Licensee may register for a "Trial License" of the Software (not available for all products or in all regions or markets). A Trial License is for product evaluation and learning purposes only and may only be used by a natural person. A Trial License lasts a limited specified period on the expiry of which the Software will automatically cease to function. Foundry may terminate any Trial License for convenience immediately on notice to Licensee. Licensee will use the Software for product evaluation and learning purposes only.

(p) "Subscription License"

If Licensee has purchased the Software on a subscription basis then: (a) the License shall be limited to the Subscription Period (as defined in clause 8.1) after which it shall automatically expire and; (b) the provisions of clause 8, including but not limited to auto-renewal, shall apply.

(q) "Nuke Stage License"

If Licensee purchases a Nuke Stage License, then: (a) Licensee may use the Software on any number of computers, provided that the number of concurrent Nuke Stage Output Device(s) shall never exceed the total number of valid Nuke Stage Floating Licenses purchased by Licensee; and (b) use of the Software shall be limited to any site in the Home Country. "Nuke Stage Output Device(s)" shall mean one region designated for rendered output, which may be mapped to one endpoint such as display windows, SDI outputs, files, or other render targets.

2.2 If Licensee has purchased a License that permits "non-interactive" use of the Software ("**Headless Rendering**"), Licensee is authorized to use a non-interactive version of the Software for rendering purposes only (i.e. without a user, in a non-interactive capacity) and shall not use such Software on workstations or otherwise in a user-interactive capacity. Headless Rendering is not available on all products. In all cases, Headless Rendering licenses may be used on any number of computers, provided that the number of concurrent computers shall never exceed the total number of valid Headless Rendering licenses purchased by Licensee.

3. RESTRICTIONS ON USE

Please note that in order to guard against unlicensed use of the Software, a license key is required to access and enable the Software. Licensee is authorised to use the Software in machine readable, object code form only (subject to clause 4), and Licensee shall not: (a) assign, sublicense, sell, distribute, transfer, pledge, lease, rent, lend, share or export the Software, the Documentation or Licensee's rights under this EULA; (b) alter or circumvent the license keys or other copy protection mechanisms in the Software or reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software in each case except as and to the extent that applicable law requires such reverse engineering, decompilation, disassembly or discovery to be permitted and it is not lawful to contract out of such requirement; (c) implement or use any method or mechanism designed to enable product functionality not available in the Software but available in (i) other Foundry products; or (ii) other Foundry releases of the

same product; (d) (subject to clause 4) modify, adapt, translate or create derivative works based on the Software or Documentation; (e) use, or allow the use of, the Software or Documentation on any project other than a project produced by Licensee (an “**Authorized Project**”) or to provide a service (whether or not any charge is made) to any third party; (f) allow or permit anyone (other than Licensee and Licensee’s authorized employees to the extent they are working on an Authorized Project) to use or have access to the Software or Documentation; (g) copy or install the Software or Documentation other than as expressly provided for in this EULA; or (h) take any action, or fail to take action, that could adversely affect the trademarks, service marks, patents, trade secrets, copyrights or other intellectual property rights of Foundry or any third party with intellectual property rights in the Software (each, a “**Third Party Licensor**”); (i) use the Software, or permit any third party to use the Software, for any illegal purpose. The Licensee shall maintain appropriate internal security, safeguards and controls which shall be at least as great as the measures the Licensee uses to keep Licensee’s own information secure (but in any case, using no less than a reasonable degree of care) to ensure that the Software is not used in any way which is in violation of this clause 3 or of this Agreement. For the avoidance of doubt, Licensee remains fully liable for any unauthorized use of the Software by its staff. For purposes of this clause 3, the term “Software” shall include any derivatives of the Software.

Notwithstanding clause 3(b) above, where the reduction of the Software to human readable form is necessary for the purposes of integrating the operation of the Software with the operation of other software or systems used by the Licensee in accordance with section 50B of the Copyright Designs and Patents Act 1988 (or any analogous legislation in other jurisdictions), prior to reducing the Software to human readable form (whether by reverse engineering, decompilation or disassembly), the Licensee shall notify Foundry in advance and allow Foundry a reasonable period to either carry out such action as is required to fulfil the integration or provide the information necessary to achieve such integration to the Licensee, and in either case the Licensee shall meet Foundry’s reasonable costs in doing so.

Unless Licensee has purchased an Individual License, a Team Login License or an Individual Login License, for other license types, if the Software is moved from one computer to another (or, in the case of an Offline Floating License, from one license server to another), the issuing of replacement or substituted license keys or approval of the new server location, is in Foundry’s sole discretion and is subject to and strictly in accordance with Foundry’s License Transfer Policy, which is available on Foundry’s website and which requires a fee to be paid in certain circumstances. Foundry may from time to time and at its sole discretion vary the terms and conditions of the License Transfer Policy.

The Software and/or any components thereof are not intended to be used for and Licensee shall not use the Software for (or modify the Software such that is suitable for): (i) prohibited practices under the European Union Artificial Intelligence Act (the “**EU AI Act**”) or similar legislation, (ii) any use that would result in the Software being considered a high-risk AI system or that otherwise qualifies as “high-risk” under the EU AI Act. Licensee may not remove, alter or obscure any metadata fields which are automatically applied to content generated by the Software which indicate that such content has been artificially generated or manipulated.

4. SOURCE CODE

Notwithstanding that clause 1 defines “Software” as an object code version and that clause 3 provides that Licensee may use the Software in object code form only:

4.1 if Foundry has agreed to license to Licensee (including by way of providing SDKs, upgrades, updates or enhancements/customization) source code or elements of the source code of the Software, the intellectual property rights in which belong either to Foundry or to a Third Party Licensor (“**Source Code**”), Licensee shall be licensed to use the Source Code as Software on the terms of this EULA and: (a) notwithstanding clause 3 (c), Licensee may use the Source Code at its own risk in any reasonable way for the limited purpose of enhancing its use of the Software solely for, where the Licensee is a business, its own internal business purposes or, where the Licensee is a consumer, domestic and private purposes and in all respects in accordance with this EULA; (b) Licensee shall in respect of the Source Code comply strictly with all other restrictions applying to its use of the Software under this EULA as well as any other restriction or instruction that is communicated to it by Foundry at any time during the Agreement (whether imposed or requested by Foundry or by any Third Party Licensor);

4.2 to the extent that the Software links to or itself incorporates any open source software and/or software libraries (“**OSS Components**”) that are provided to Licensee with or as part of the Software, then where such OSS Components are licensed on the terms of an open source software licence that requires Foundry to make the OSS Components available to the Licensee on specific terms (the “**OSS Licence Terms**”), those OSS Components are licensed to Licensee on, and subject to, the terms of the relevant OSS Licence Terms;

4.3 where Foundry is required by any OSS Licence Terms to make the source code of the relevant OSS Component available to the Licensee, Foundry will at any time during the three year period starting on the date of the Agreement, at the request of Licensee and subject to Licensee paying to Foundry a charge that does not exceed Foundry’s costs of doing so, provide Licensee with the source code of the relevant OSS Component (the “**OSS Source Code**”) in order that Licensee may modify the OSS Component in accordance with the relevant OSS Licence Terms, together (where appropriate) with certain object code of the Software necessary to enable Licensee to re-link any modified OSS Components to the Software (the “**Object**”); and

4.4 notwithstanding any other term of the Agreement, Foundry gives no express or implied warranty, undertaking or indemnity whatsoever in respect of the Source Code, the OSS Components, the OSS Source Code or the Object, all of which are licensed on an “as is” basis, or in respect of any modification of the Source Code, the OSS Components or the OSS Source Code made by Licensee (“**Modification**”). Licensee may not use the Object for any purpose other than its use of the Software in accordance with this EULA. Notwithstanding any other term of the Agreement, Foundry shall have no obligation to provide support, maintenance, upgrades or updates of or in respect of any of the Source Code, the OSS Components (save for any obligations Foundry may have in respect of any elements that form part of the Software as a whole), the OSS Source Code, the Object or any Modification. Licensee shall indemnify Foundry against all liabilities and expenses (including reasonable legal costs) incurred by Foundry in relation to any claim asserting that any Modification infringes the intellectual property rights of any third party.

5. BACK-UP COPY

Licensee may store one copy of the Software and Documentation off-line and off-site in a secured location within the Home Country that is owned or leased by Licensee in order to provide a back-up in the event of destruction by fire, flood, acts of war, acts of nature, vandalism or other incident. In no event may Licensee use the back-up copy of the Software or Documentation to circumvent the usage or other limitations set forth in this EULA.

6. OWNERSHIP

Licensee acknowledges that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation and all related intellectual property rights and other proprietary rights are and shall remain the sole property of Foundry and the Third Party Licensors. Licensee shall not remove, or allow the removal of, any copyright or other proprietary rights notice included in and on the Software or Documentation or take any other action that could adversely affect the property rights of Foundry or any Third Party Licensor. To the extent that Licensee is authorized to make copies of the Software or Documentation under this EULA, Licensee shall reproduce in and on all such copies any copyright and/or other proprietary rights notices provided in and on the materials supplied by Foundry hereunder. Nothing in the Agreement shall be deemed to give Licensee any rights in the trademarks, service marks, patents, trade secrets, confidential information, copyrights or other intellectual property rights of Foundry or any Third Party Licensor, and Licensee shall be strictly prohibited from using the name, trademarks or service marks of Foundry or any Third Party Licensor in Licensee's promotion or publicity without Foundry's prior express written approval.

Subject to clause 4.3, Foundry undertakes (the "**Undertaking**") to defend Licensee or at Foundry's option settle any claim brought against Licensee alleging that Licensee's possession or use of the Software or Documentation in accordance with the Agreement infringes the intellectual property rights of a third party in the same country as Licensee ("**Claim**") and shall reimburse all reasonable losses, damages, costs (including reasonable legal fees) and expenses incurred by or awarded against Licensee in connection with any such Claim, provided that the Undertaking shall not apply where the Claim in question is attributable to possession or use of the Software or Documentation other than in accordance with the Agreement, or in combination with any hardware, software or service not supplied or specified by Foundry. The Undertaking is conditional on Licensee giving written notice of the Claim to Foundry as soon as reasonably possible, cooperating in the defence of the Claim and not making any admission of liability or taking any step prejudicial to the defence of the Claim. If any Claim is made, or in Foundry's reasonable opinion is likely to be made, against Licensee, Foundry may at its sole option and expense (a) procure for Licensee the right to continue using the Software, (b) modify the Software so that it ceases to be infringing, (c) replace the Software with non-infringing software, or (d) terminate the Agreement immediately by notice in writing to Licensee and refund the License Fee (less a reasonable sum in respect of Licensee's use of the Software to

the date of termination) on return of the Software and all copies by Licensee. The Undertaking constitutes Licensee's exclusive remedy and Foundry's only liability in respect of any Claim.

7. LICENSE FEE

7.1 Licensee acknowledges that the rights granted to Licensee under this EULA are conditional on Licensee's timely payment of the license fee payable to Foundry in connection with the Agreement or, as the case may be, payable to Foundry's reseller (the "**License Fee**"). Except where the applicable invoice expressly sets out payment in instalments, the License Fee shall be payable in full as one single payment.

7.2 Licensee will be charged and agrees to pay to Foundry or Foundry's authorized reseller (as applicable): (a) the License Fee as notified by Foundry (or its reseller) at the time of the initial purchase of the License; and (b) in respect of any Subscription Auto-renewal Period for a Subscription License, the License Fee as notified by Foundry (or its reseller) on or about the applicable Renewal Date.

7.3 Unless stated otherwise, any Licence Fee notified to the Licensee by Foundry (or its reseller) is exclusive of sales tax, VAT and any other similar taxes, duties or levies, which shall be payable by the Licensee (and the Licensee agrees to pay) in addition to the Licence Fee.

7.4 In the cases of Non-Commercial Licenses or Trial Licenses or specific End of Life Software, the fact that no License Fee may be payable shall not be construed as a waiver by Foundry of any right or remedy available to it in relation to any breach by the Licensee of this EULA or the Agreement, or of any other right or remedy arising under applicable law, all of which are expressly reserved.

8. SUBSCRIPTION LICENSES AND AUTO-RENEWAL

8.1 If Licensee has purchased a Subscription Licence, the License shall be limited to the Initial Subscription Period and any/all Auto-renewal Periods (each as defined below) (together the "**Subscription Period**") after which it shall automatically expire.

8.2 The Subscription Licence shall begin as soon as Foundry accepts Licensee's order by issuing Licensee with a license key (the "**Subscription Start Date**") and shall continue for an initial period of twelve (12) months (the "**Initial Subscription Period**") unless earlier terminated in accordance the terms of this EULA, or unless otherwise communicated by Foundry.

8.3 Unless Licensee opts out of auto-renewal in accordance with clause 8.5 then upon the first anniversary of the Subscription Start Date and each subsequent anniversary (each a "**Subscription Renewal Date**"), Licensee's Subscription Licence shall renew automatically for a further twelve (12) months (each an "**Auto-renewal Period**"). Licensee's Subscription License will continue to auto-renew in this manner until Licensee opts out of auto-renewal or unless earlier terminated in accordance with the terms of this EULA.

8.4 Prior to each Subscription Renewal Date, Foundry shall send one (1) email to advise you that your Subscription License is approaching auto-renewal to the contact email address as provided by Licensee in accordance with clause 23. The reminder email will be sent not less than thirty (30) days prior to the relevant Subscription Renewal Date.

8.5 Opting Out of Auto-renewal and License Expiry. If Licensee wishes to opt out of auto-renewal then you must email licenses@foundry.com providing details of the Subscription Licence(s) which you wish to opt out not less than seventy-two (72) hours prior to the relevant Subscription Renewal Date. Provided that Licensee notifies Foundry in accordance with the provisions of this clause 8.5 then your Subscription License (s) will not auto-renew and shall expire at the end of the then-current Subscription Period.

8.6 Increases to the License Fee for Subscription Licenses. Foundry reserves the right to increase the License Fee for Subscription Licenses from time to time provided that it shall provide Licensee with not less than thirty (30) days' notice of any increase prior to the relevant Subscription Renewal Date.

9. CANCELLATIONS

9.1 Licensee may cancel a License within 14 days of the original purchase date to obtain a full refund and Licensee will no longer be able to use the Software from the cancellation date. Licensee's right to obtain a refund will be lost once the Software has been installed.

9.2 Refunds are not payable for cancellations made after such date. This includes Subscription licenses which are subject to the fixed twelve (12) month terms and for which the Licensee may opt out of auto-renewal in accordance with Clause 8.

9.3 Cancellations and requests for refunds can be made by contacting Foundry's Sales Support team at licenses@foundry.com.

10. END OF LIFE

10.1 From time to time, Foundry will retire Software product(s) (and any/ all versions) by withdrawing them from the market. Such Software product(s) will be designated as end of life ("**EoL**") and a corresponding announcement (the "**Announcement**") will be published on Foundry's website from time to time (the date of such announcement being the "**EoL Notification Date**"). The announcement will include an EoL date ("**EoL Date**") and other relevant information.

10.2 No new License(s) (including renewals, if applicable) for the EoL Software product(s) will be sold after the EoL Notification Date. Unless otherwise provided for in the Announcement, the EoL Date will be the last day that the EoL Software product(s) is sold, improved, maintained, updated, or supported (subject to the below).

10.3 Unless otherwise provided for in the Announcement, a Licensee with an active Subscription License(s) at the time of the EoL Date will receive certain limited support until the relevant Subscription Period expires. A Licensee with an active perpetual License(s) (purchased on or prior to 31 December 2023) for which it has renewed (or auto renewed) the associated Maintenance and Support Agreement prior to the EoL Notification Date, will receive certain limited support until the EoL Date, or for a period specified in the Announcement.

10.4 In the event of EoL, Foundry advises Licensee to install the latest product releases, security updates, fixes and patches available until the EoL Date, to remain as secure as possible. Older products may not meet security or performance requirements, and Foundry cannot confirm compatibility with future operating system updates after the EoL Date. To the maximum extent permitted by applicable law, Foundry shall not be liable for any form of loss, damage or disruption in respect of the EoL Software product(s) beyond the EoL Date.

10.5 Foundry retains all rights, including intellectual property rights, associated with the EoL Software product(s) beyond the EoL Date. Any use by Licensee of the EoL Software product(s) beyond the EoL Date will continue to be governed by the terms of this Agreement.

11. MAINTENANCE AND SUPPORT

11.1 Subject to Licensee's timely payment of the applicable License Fee, each Subscription Licence and each Rental License shall include access to certain maintenance and support services in accordance with the Maintenance and Support Agreement which is available on Foundry's website (the "**Maintenance and Support Agreement**").

11.2 If Licensee purchased a perpetual License on or prior to 31 December 2023 for which it has renewed (or auto-renewed) the associated Maintenance and Support Agreement, then such License shall include access to certain maintenance and support services for the relevant continuous Renewal Support Period(s) (as defined in, and in accordance with, the Maintenance and Support Agreement), after which Licensee shall not be entitled to purchase further maintenance and support for such License.

11.3 Foundry may vary the terms and conditions of the Maintenance and Support Agreement from time to time and at its sole discretion.

12. TAXES AND DUTIES

Licensee agrees to pay, and indemnify Foundry from claims for, any local, state or national tax (exclusive of taxes based on net income), duty, tariff or other impost related to or arising from the transaction contemplated by the Agreement.

13. LIMITED WARRANTY

13.1 Subject to clause 13.3, Foundry warrants that, for a period of ninety (90) days after Licensee first downloads the Software ("**Warranty Period**"): (a) the Software will, when properly used on an operating system for which it was designed, perform substantially in accordance with the functions described in the Documentation; and (b) that the Documentation correctly describes the operation of the Software in all material respects. If, within the Warranty Period, Licensee notifies Foundry in writing of any defect or fault in the Software as a result of which it fails to perform substantially in accordance with the Documentation, Foundry will, at its sole option, either repair or replace the Software, provided that Licensee makes available all the information that may be necessary to identify, recreate and remedy the defect or fault. This warranty will not apply to, and Foundry shall have no liability for, any defect or fault caused by: (a) unauthorised use of or any amendment made to the Software by any person other than Foundry; and/or (b) use of the Software in conjunction with third party technology. If Licensee is a consumer, the warranty given in this clause is in addition to Licensee's legal rights in relation to any Software or Documentation that is faulty or not as described.

13.2 Foundry does not warrant that the Software or Documentation will meet Licensee's requirements or that Licensee's use of the Software will be uninterrupted or error free.

13.3 If Licensee purchases a license of the Software that is of a fixed term duration, the Warranty Period in clause 13.1 shall apply only to Licensee's first purchase of such license and not to any subsequent renewal(s) even if a renewal involves another download.

14. INDEMNIFICATION

14.1 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use including but not limited to modification of the Software or Documentation by Licensee or any person connected with Licensee or any person provided with the Software by Licensee, infringes the intellectual property rights or other proprietary rights of any third party.

14.2 Licensee agrees to indemnify, hold harmless and defend Foundry, the Third Party Licensors and Foundry's and each Third Party Licensor's respective affiliates, officers, directors, shareholders, employees, authorized resellers, agents and other representatives from all claims, defence costs (including, but not limited to, legal fees), judgments, settlements and other expenses arising from or connected with any claim that any authorised or unauthorised use (including but not limited to modification) of the Software or Documentation, by Licensee, any person connected with Licensee or any person provided with the Software by Licensee, causes any of them or any third party (including but not limited to those viewing a product of the Software) to suffer harm or loss of any kind.

15. LIMITATION OF LIABILITY TO BUSINESS USERS

This clause applies where Licensee is a business user (and for these purposes any Licensee that is not a consumer shall be treated as a business user). Licensee acknowledges that the Software has not been developed to meet its individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation is supplied only for Licensee's internal use for its business, and not for any re-sale purposes or for the provision of the Software (whether directly or indirectly) to third parties. Foundry shall not under any circumstances whatever be liable to Licensee, its affiliates, officers, directors, shareholders, employees, agents or other representatives, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with the Agreement for loss of profits, sales, business, or revenue, business interruption, loss of anticipated savings, loss or corruption of data or information, loss of business opportunity, goodwill or reputation (in each case whether the loss is direct or indirect) or any indirect or consequential loss or damage. In respect of any other losses, Foundry's maximum aggregate liability under or in connection with the Agreement whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to the greater of US\$5,000 (five thousand USD) and a sum equal to the License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law. This EULA sets out the full extent of our obligations and liabilities in respect of the supply of the Software and Documentation. Except as expressly stated in writing in this EULA, there are no conditions, warranties, representations or other terms, express or implied, that are binding on Foundry. Any condition, warranty, representation or other term concerning the supply of the Software and Documentation which might otherwise be implied into, or incorporated in, the Agreement, whether by statute, common law or otherwise, is excluded to the fullest extent permitted by law.

16. LIMITATION OF LIABILITY TO CONSUMERS

This clause applies where Licensee is a consumer. Licensee acknowledges that the Software has not been developed to meet Licensee's individual requirements, and that it is therefore Licensee's responsibility to ensure that the facilities and functions of the Software as described in the Documentation meet such Licensee requirements. The Software and Documentation are only supplied for Licensee's domestic and private use. Licensee agrees not to use the Software and Documentation for any commercial, business or re-sale purposes, and Foundry has no liability to Licensee for any loss of profit, loss of business, business interruption, loss of business opportunity or loss or harm caused to property or data used for professional purposes. Foundry is only responsible for loss or damage suffered by Licensee that is a foreseeable result of Foundry's breach of the Agreement or its negligence but Foundry is not responsible for any loss or damage that is not foreseeable. Our maximum aggregate liability under or in connection with the Agreement, whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to a sum equal to the greater of US\$5,000 (five thousand USD) and a sum equal to the latest annual License Fee.

Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law.

17. TERM; TERMINATION

17.1 The Agreement is effective upon Licensee's download of the Software, and the Agreement will remain in effect until termination or expiry. Licensee may terminate the Agreement on written notice to Foundry if Foundry is in material breach of this Agreement and fails to cure the breach within 10 (ten) working days of receiving notice of such breach. If Licensee breaches the Agreement, Foundry may terminate the License immediately by notice to Licensee.

17.2 If the Agreement expires or is terminated, the License will cease immediately and Licensee will immediately cease use of any Software and Documentation and either return to Foundry all copies of the Software and Documentation in Licensee's possession, custody or power or, if Foundry directs in writing, destroy all such copies. In the latter case, if requested by Foundry, Licensee shall provide Foundry with a certificate confirming that such destruction has been completed within 10 days.

17.3 Foundry reserves the right to terminate and/or suspend the License as it deems reasonable in its sole discretion by notice to Licensee if it becomes aware that Licensee has failed to pay any sum due either to Foundry or to a reseller of Foundry either in connection with the Agreement or in connection with any other Software license to use any product(s) of Foundry, in connection with any other agreed terms between Foundry and the Licensee in connection with the Software, in connection with any Maintenance and Support Agreement or if the Licensee is otherwise in breach of or fails to comply with any term of the Agreement.

17.4 Foundry may also terminate this Agreement if Licensee becomes subject to bankruptcy proceedings, becomes insolvent, or makes an arrangement with Licensee's creditors. This Agreement, and for the avoidance of doubt the Licenses to which it relates, will terminate automatically without further notice or action by Foundry if Licensee goes into liquidation.

18. CONFIDENTIALITY

Licensee agrees that the Software (including, for the avoidance of doubt, any Source Code that is licensed to Licensee) and Documentation are proprietary to and the confidential information of Foundry or, as the case may be, the Third Party Licensors, and that all such information and any related communications (collectively, "**Confidential Information**") are confidential and a fundamental and important trade secret of Foundry and/or the Third Party Licensors. If Licensee is a business user, Licensee shall disclose Confidential Information only to Licensee's employees who are working on an Authorized Project and have a "need-to-know" such Confidential Information for the purposes of that Authorized Project, and shall advise any recipients of Confidential Information that it is to be used only as expressly authorized in the Agreement. Licensee shall not disclose Confidential Information or otherwise make any Confidential Information

available to any other of Licensee's employees or to any third parties without the express written consent of Foundry. Licensee agrees to segregate, to the extent it can be reasonably done, the Confidential Information from the confidential information and materials of others in order to prevent commingling. Licensee shall take reasonable security measures, which measures shall be at least as great as the measures Licensee uses to keep Licensee's own confidential information secure (but in any case using no less than a reasonable degree of care), to hold the Software, Documentation and any other Confidential Information in strict confidence and safe custody. Foundry may request, in which case Licensee agrees to comply with, certain reasonable security measures as part of the use of the Software and Documentation. This clause shall not apply to any information that is in or comes into the public domain (other than as a result of the Licensee's breach of its obligations under the Agreement), or was in Licensee's lawful possession before receipt or which Licensee develops independently and without breach of this clause. Licensee acknowledges that monetary damages may not be a sufficient remedy for unauthorized disclosure of Confidential Information, and that Foundry shall be entitled, without waiving any other rights or remedies, to such injunctive or other equitable relief as may be deemed proper by a court of competent jurisdiction.

19. INSPECTION AND INFORMATION

19.1 Unless Licensee is a consumer, Licensee shall advise Foundry on demand of all locations where the Software or Documentation is used or stored. Licensee shall permit Foundry or its authorized agents to audit all such locations during normal business hours and on reasonable advance notice.

19.2 The Software may include mechanisms to access and collect limited information from computer(s) on which it is installed and from any IT systems to which those computer(s) may be connected (including any system registry files) and transmit it to Foundry and/or its resellers, including the ability to locally cache such information on such computers. Such information (the "**Information**") may include details of relevant license(s) to Foundry products, details of computer and network equipment, details of the operating system(s) in use on such computer equipment, email domain relating to owners of such computer and network equipment, the location of the computer(s) on which the Software is installed and the profile and extent of use of the different elements of the Software and other Foundry software. Foundry may use the Information to (a) model the profiles of usage, hardware and operating systems in use collectively across its customer base in order to focus development and support, (b) to provide targeted support to individual customers, (c) to ensure that the usage of the Software by Licensee is in accordance with the Agreement and does not exceed any user number or other limits on its use, (d) to confirm the identity of Licensee, to identify unlicensed use of the Software (including use of pirated or other unlicensed copies of the Software) and to assist Foundry (and its resellers and any enforcement bodies) in contacting any unlicensed users of the Software and seeking to terminate unlicensed use of the Software, and (e) to advise Licensee about service issues such as available upgrades and maintenance expiry dates. To the extent that any Information constitutes personal data for the purposes of the General Data Protection Regulation (EU) 2016/679 ("**EU GDPR**") and the version of the EU GDPR retained in UK domestic law as further defined in the Data Protection Act 2018 ("**UK GDPR**"), in each case, as amended, superseded or replaced from time to time (as applicable, "**GDPR**") it shall be processed in accordance with the GDPR and with Foundry's Privacy Notice

(see <https://www.foundry.com/privacy-notice>), as may be updated by Foundry from time to time. Licensee undertakes to make all of users of the Software aware of the uses which Foundry will make of the Information and of the terms of Foundry's Privacy Policy.

19.3 By downloading or using the Software, you (i) warrant that you are entitled to control access to the computer(s) on which the Software is downloaded and any IT systems to which they may be connected, and (ii) irrevocably authorise Foundry (through the use of the Software) to access such computer(s) and IT systems (including any system registry files) and collect the Information from them and to transmit that Information to Foundry and its resellers (and any enforcement bodies) and use it for the purposes identified at clause 19.2(a) to (e) above.

20. U.S. GOVERNMENT LICENSE RIGHTS

All Software, including all components thereof, and Documentation qualify as "commercial items," as that term is defined at Federal Acquisition Regulation ("**FAR**") (48 C.F.R.) 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in FAR 12.212. Consistent with FAR 12.212 and DoD FAR Supp. 227.7202-1 through 227.7202-4, and notwithstanding any other FAR or other contractual clause to the contrary in any agreement into which this Agreement may be incorporated, a government end user will acquire the Software and Documentation with only those rights set forth in this Agreement. Use of either the Software or Documentation or both constitutes agreement by the government that all Software and Documentation are "commercial computer software" and "commercial computer software documentation," and constitutes acceptance of the rights and restrictions herein. The Software is the subject of the following notices:

* Copyright © 2001 - 2025 The Foundry Visionmongers Ltd. All Rights Reserved.

* Unpublished- rights reserved under the Copyright Laws of the United Kingdom.

21. SURVIVAL

Clause 6, clause 7 and clauses 12 to 25 inclusive shall survive any termination or expiration of the Agreement.

22. IMPORT/EXPORT CONTROLS AND SANCTIONS

To the extent that any Software made available under the Agreement is subject to restrictions upon export and/or re-export from any applicable jurisdiction (including the United States), Licensee agrees to comply with, and not act or fail to act in any way that would violate, applicable international, national, state, regional or local laws and regulations, including, without limitation, the U.S. Export Administration Act and the Export Administration Regulations, the regulations of the U.S. Department of Treasury Office of Foreign Assets Control, the International Traffic in Arms regulations, the United States Foreign Corrupt Practices Act, the UK Export Control Act 2002 and the UK Export Control Order 2008 (collectively, "**Export Laws**") as those laws

may be amended or otherwise modified from time to time, and neither Foundry nor Licensee shall be required under the Agreement to act or fail to act in any way which it believes in good faith will violate any such laws or regulations. Without limiting the foregoing, Licensee agrees that it will not sell, supply, transfer, export or re-export, directly or indirectly, Foundry's Software or related products and services, or any commodity, technology, technical data, software or service that incorporates, contains or is a direct product of Foundry's Software, products and/or services, (i) in violation of the Export Laws; (ii) to any country for which an export license or other governmental approval is required at the time of export, without first obtaining all necessary export licenses or other approvals; (iii) to any country, or national or resident of a country, to which trade is embargoed by the United States and/or sanctioned by the United Kingdom or the European Union; (iv) to any person or firm on any government agency's list of blocked, denied or barred persons or entities, including but not limited to the U.S. Department of Commerce's Denied Persons List and Entities List, and the U.S Treasury Department's Specially Designated Nationals List; or (v) to the Russian Federation and/or Belarus or for use in the Russian Federation and/or Belarus, or (vi) for use in any nuclear, chemical or biological weapons, or missile technology end-use. Licensee warrants and represents that it is not subject to any Export Law, sanction and/or other form of trading compliance restriction or regulation which may limit or prohibit its right to download, install and/or use the Software in accordance with this EULA.

23. MISCELLANEOUS

Unless Licensee is a consumer, the Agreement is the exclusive agreement between the parties concerning its subject matter and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning such subject matter. Licensee acknowledges that Licensee has not relied upon any representation or collateral warranty not recorded in the Agreement inducing it to enter into the Agreement.

The Agreement may be modified only in writing, by Foundry, at any time.

The failure of either party to enforce any rights granted under the Agreement or to take action against the other party in the event of any such breach shall not be deemed a waiver by that party as to subsequent enforcement of rights or subsequent actions in the event of future breaches.

The Agreement and any dispute or claim arising out of or in connection with it or its subject matter or formation (including, unless Licensee is a consumer, non-contractual disputes or claims) shall be governed by, and construed in accordance with English Law and the parties irrevocably submit to the non-exclusive jurisdiction of the English Courts, subject to any right that a consumer may have to bring proceedings or to have proceedings brought against them in a different jurisdiction.

If Foundry fails to insist that Licensee performs any obligation under the Agreement, or delays in doing so, that will not mean that Foundry has waived its rights.

If any provision or part-provision of this Agreement is or becomes invalid, illegal or unenforceable, it shall be deemed deleted, but that shall not affect the validity and enforceability of the rest of this Agreement.

Unless Licensee is a consumer, Licensee agrees that Foundry may refer to Licensee as a client or a user of the Software, may display its logo(s) for this purpose and may publish quotations and testimonials from Licensee, its directors, partners, officers or employees. Foundry agrees to promptly cease any such use on Licensee's written request.

Foundry and Licensee intend that each Third Party Licensor may enforce against Licensee under the Contracts (Rights of Third Parties) Act 1999 (the "Act") any obligation owed by Licensee to Foundry under this EULA that is capable of application to any proprietary or other right of that Third Party Licensor in or in relation to the Software. Foundry and Licensee reserve the right under section 2(3)(a) of the Act to rescind, terminate or vary this EULA without the consent of any Third Party Licensor.

Email Address for General Correspondence: Licensee shall notify Foundry of an email address for the provision of any correspondence in connection with the Agreement and shall notify Foundry via licenses@foundry.com of any change(s) to that email address. Please note, the email address you provide is important for the provision of information and notices to you, including in relation to the autorenewal of any Subscription License (if applicable). It is your responsibility to provide and maintain an up-to-date email address. Foundry shall store details of and may use the email address to notify you in accordance with the terms of this Agreement.

24. NOTICES

Any legal notices from the Licensee to Foundry should be sent via postal mail or international delivery to The Foundry Visionmongers Ltd, 5 Golden Square, London, W1F 9HT, addressed to The General Counsel. These notices will be effective upon receipt by Foundry. Additionally, a copy of the notice should be emailed to Foundry at legal@foundry.com. Unless otherwise agreed in writing, any notices from Foundry to the Licensee will be provided either (a) by email to the registered email address associated with the Licensee's account, or (b) in any other reasonable manner that involves specific notification to the Licensee. Notices from Foundry to the Licensee will be effective (i) one day after being sent by email, and (ii) five days after being posted or sent by international delivery service. The Licensee agrees that service of process may be effected on the Licensee by registered mail sent to the last known address on record with Foundry, if permitted by applicable law.

25. COMPLAINTS & ONLINE DISPUTE RESOLUTION PLATFORM

We hope that you are satisfied with any Software purchase made or service received from Foundry, but if you have a complaint, in the first instance, please contact us on licenses@foundry.com or through our Support Portal: <https://support.foundry.com/hc/en-us> (for technical support and bug reports), or you can request a call back from the Sales team here: <https://www.foundry.com/contact-us>.

Last updated 27 March 2025.

Copyright © 27 March 2025 The Foundry Visionmongers Ltd.

All Rights Reserved. Do not duplicate.