FLIX

GUIDE TO FLIX

Version 6.6.1

FOUNDRY.

Contents

Flix Install Guide

Installing Flix Server

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

RAM	At least 16 GB

> **Note:** The volume where the **/tmp** directory resides should have at least twice as much free space as the largest AAF imported into Flix. We recommend at least 32 GB, which is enough for most studios.

CPU	A 64-bit Intel processor @ 3.2 GHz, or higher
	8 cores, or more

Minimum 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
RAM	8 GB

> **Note:** The volume where the **/tmp** 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 5.7

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

> ⚠️ **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.

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:

> 📝 **Note:** 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.

Download Flix Server

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

   For Mac, a **.dmg** file is downloaded.

   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:

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:

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 5.7 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.

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 Flix Client for HTTPS.

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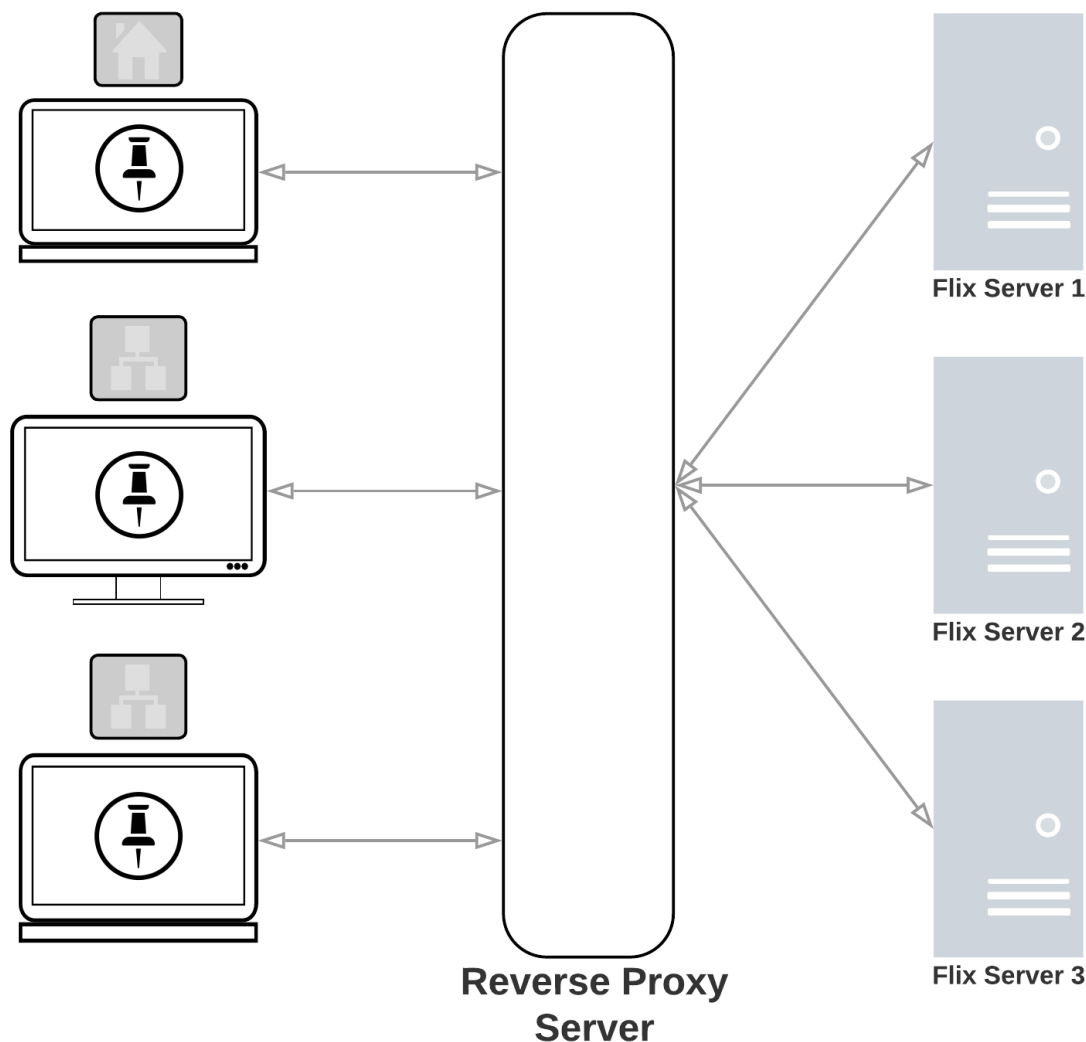
The reverse proxy server handles server load bearing and hides
your Flix Servers for added security.

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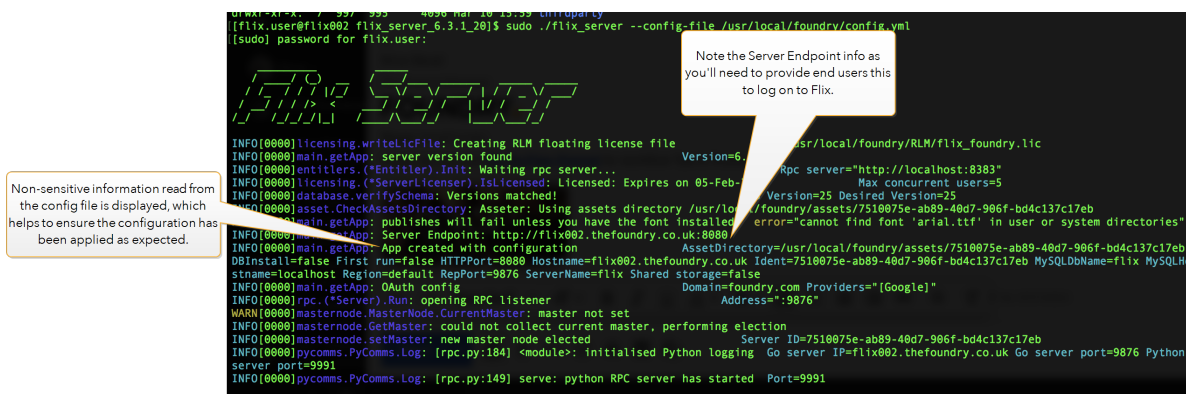
If your server config file is located in the same directory as your Flix Server binary, you can run the server as follows:

```
./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.

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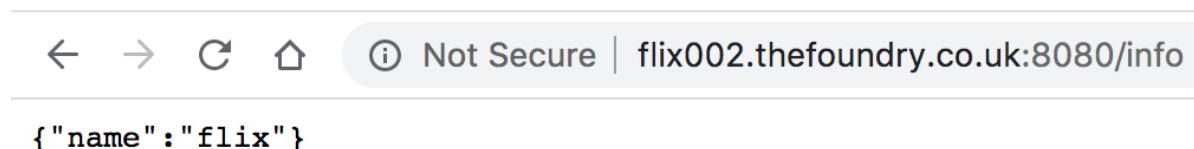
    If you receive a connection error in the web browser, try the following:

      • 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:

```
[[flix.user@flix002 foundry]$ pwd
 /usr/local/foundry
[[flix.user@flix002 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
 ├── flix_server_6.0.5
 │   ├── flix_server
 │   ├── flix_server.log
 │   ├── rpc-linux
 │   └── thirdparty
 ├── flix_server_6.1.2_23
 │   ├── flix_server
 │   ├── flix_server.log
 │   ├── flix-server-utility
 │   └── thirdparty
 ├── flix_server_6.2.2_30
 │   ├── flix_server
 │   ├── flix_server.log
 │   ├── flix-server-utility
 │   └── thirdparty
 └── flix_server_6.3.1_20
     ├── flix_server
     ├── flix-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.getTables: 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.

6. 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
6.6.0 - 6.6.1	58
6.5.0 - 6.5.1	56
6.4.0 - 6.4.1	42
6.3.7	29
6.3.0 - 6.3.2	25
6.3.3 - 6.3.4	26
6.3.5 - 6.3.6	28
6.2.0 - 6.2.2	20
6.1.0 - 6.1.2	12
6.0.0	4
6.0.1 - 6.0.5	5

> **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
```

7. Flix asks if you want to backup the database. Type **Y** (yes) or **N** (no).
8. Enter the desired database version to roll back to. Please refer to the table above for reference.
9. 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.

Manually Installing the Photoshop Plug-in for End Users

If story artists do not have the required admin privileges to install the Photoshop plug-in via Flix Client, system administrators can install it manually using these steps.

> **Note:** The following steps assume your Flix Client app is installed in **/Applications/Flix** on Mac OS and **C:\Program Files\Flix** on Windows.

1. Locate the **flix.zxp** file bundled inside the Flix Client app, at the following location:
   - Mac: **Flix.app/Contents/Resources/flix.zxp**

- Windows: **resources\flix.zxp**

2. Rename this file's extension to **.zip**, so you can extract the contents.

   The file should now be called '**flix.zip**'.

3. Unzip the **flix.zip** file.

   A folder called '**flix**' is created.

4. Rename this folder to '**com.foundry.FLIX**'.

5. Move this folder to the correct Photoshop CEP location:

   - Mac: **/Library/Application Support/Adobe/CEP/extensions**
   - Windows: **C:\Program Files (x86)\Common Files\Adobe\CEP\extensions**

The Photoshop plug-in is now installed.

6. To allow artists to map keyboard shortcuts (or 'hotkeys') to Flix commands in Photoshop, copy the **hotkeys** directory into the **Presets**/**Scripts** directory of your artists' Photoshop version folder.

   For example:

   - Mac: copy the **hotkeys** folder from **/Library/Application Support/Adobe/CEP/extensions** > **/Applications/Adobe Photoshop <version>/Presets/Scripts/**
   - Windows: copy the **hotkeys** folder from **C:\Program Files (x86)\Common Files\Adobe\CEP\extensions** > **C:\Program Files\Adobe\Adobe Photoshop <version>\Presets\Scripts**

> ⚠️ **Warning:** Do not delete or move the **hotkeys** directory from the original path, otherwise the Photoshop plug-in will not function correctly.

Copy the Hotkeys directory into the Scripts directory of the Photoshop folder.
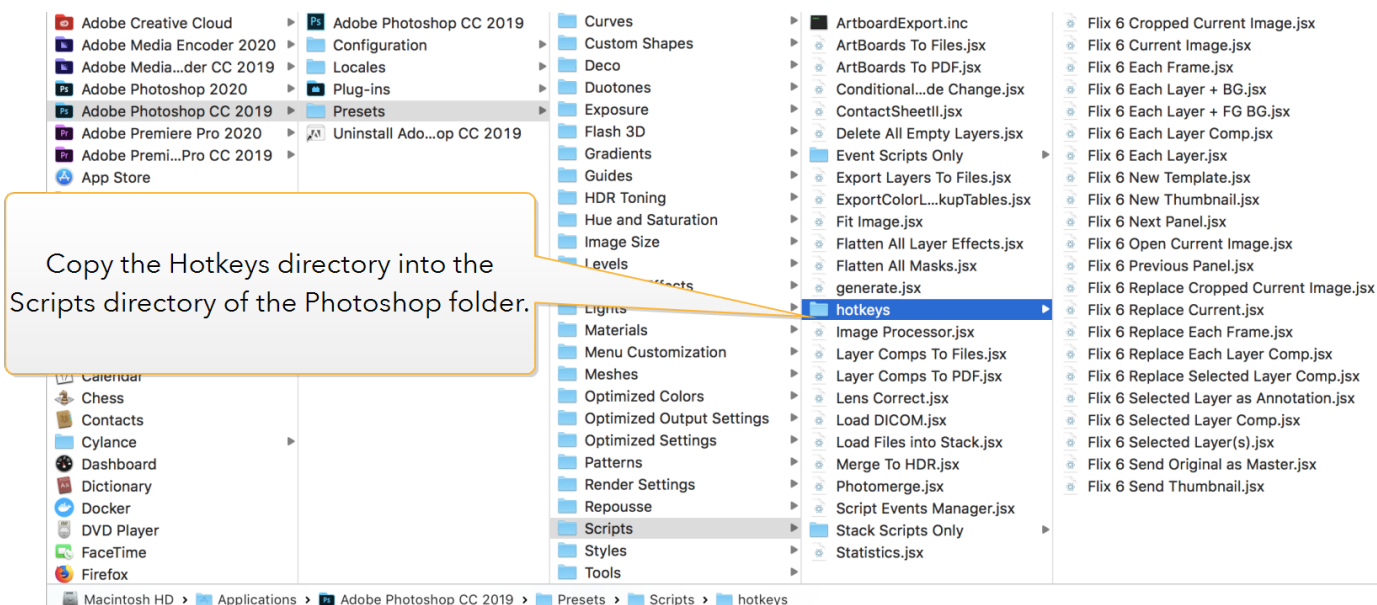
Adding Servers

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	
Operating System	• 11.x (Big Sur) • 12.x (Monterey) • 13.x (Ventura)
RAM	4 GB
Processor	ARM-based Apple Silicon M1 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	
Operating System	Windows 10 or 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.
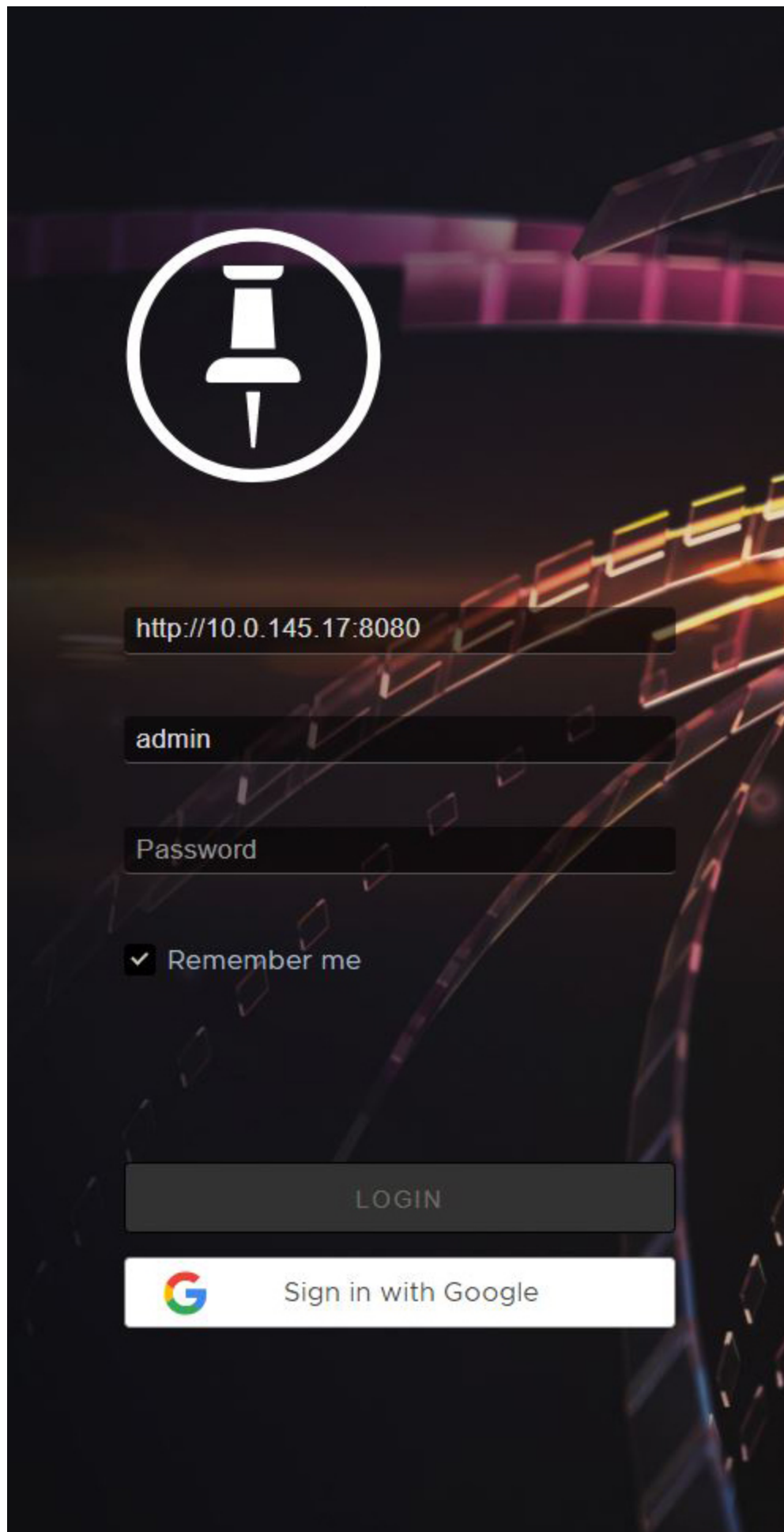
2. In the **Server Hostname** field, enter the server address. These credentials can be obtained from your System Administrator. See Installing Flix Server.

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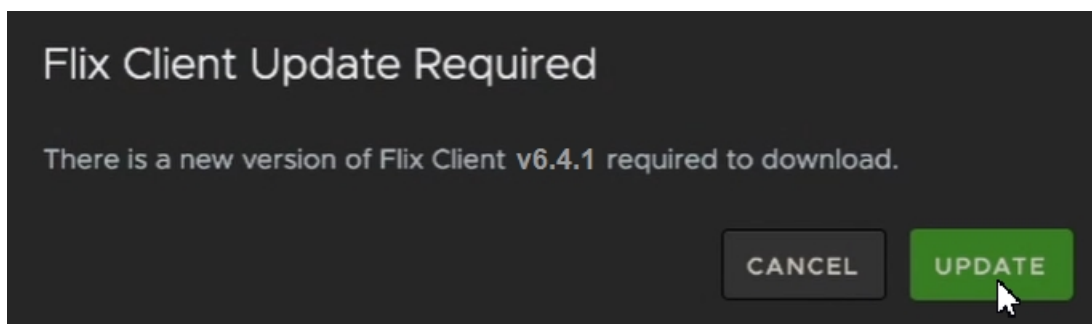
Logging in successfully opens Flix at the **Shows** level.

> **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 Flix Client for HTTPS

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 TSL 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 TSL 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.

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.

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 **distingishedName**, **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.

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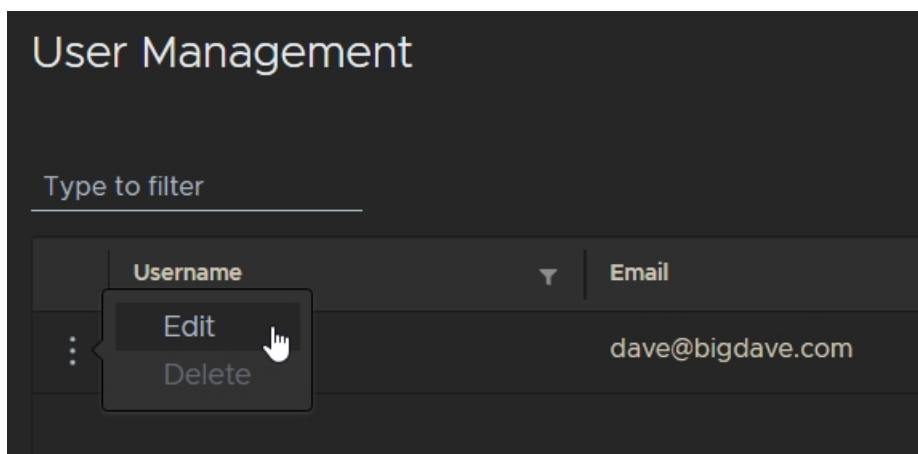
3. Make the required changes in the **Edit user** dialog.

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 <img> button and select **Delete**.
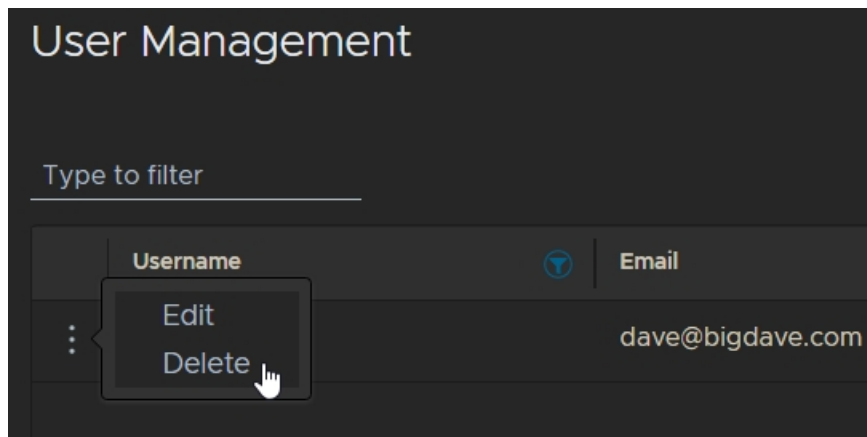
3. A warning is displayed because accounts are not recoverable if they are deleted.
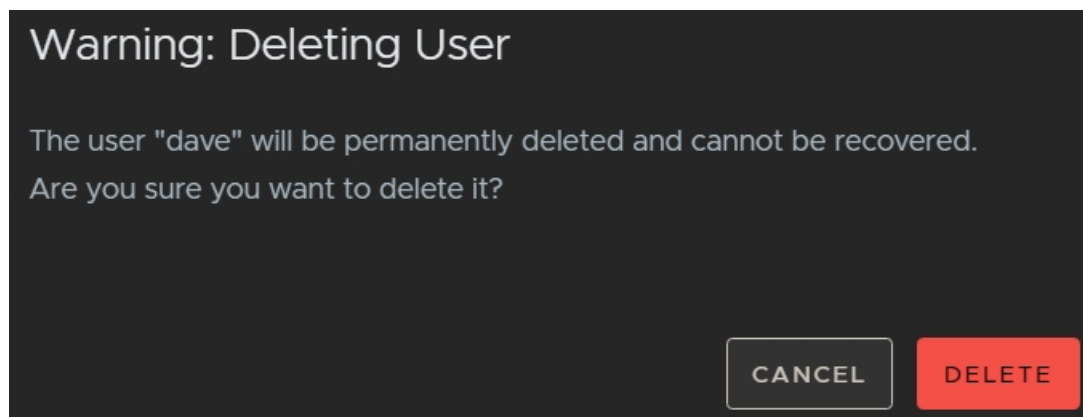
4. Click **Delete** to remove the account permanently.

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.

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.

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.

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 you own as well. See Add Custom Roles for more information.

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

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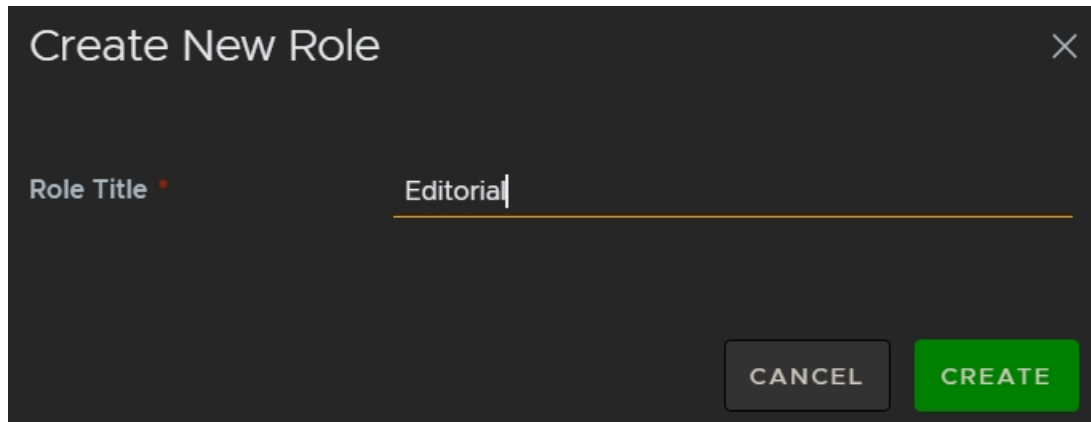
You can create your own roles in addition to the defaults if you need more control over who can access what in Flix.

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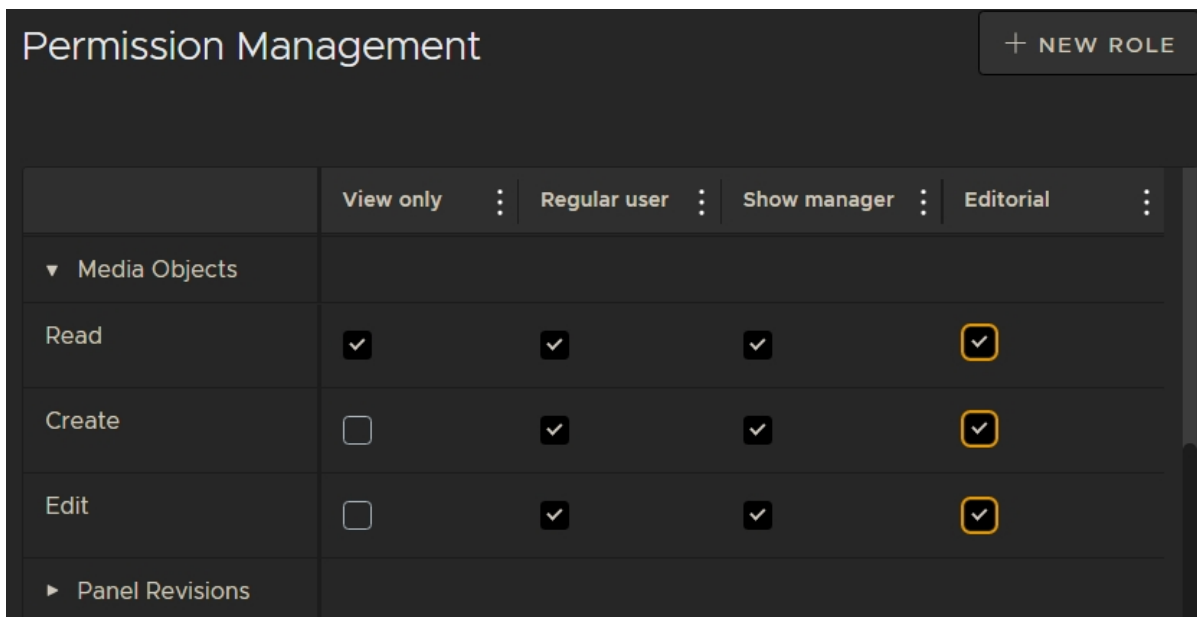
3. Enter a **Role Title** and click **Create** to save the new role.

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.

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

Flix User Guide

Getting Started

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.

Launching Flix

Once Flix Server is installed, all you need to do is launch Flix Client, the desktop application.

1. Set up a server whether it is at your studio or in the Cloud. SeeInstalling Flix Server.

2. Download and unzip the **.zip** file provided by Foundry.

3. In the unzipped folder, double-click or open the following executable file:
   • Windows: **Flix.exe**
   • Mac: **Flix.app**

   This opens the login page.
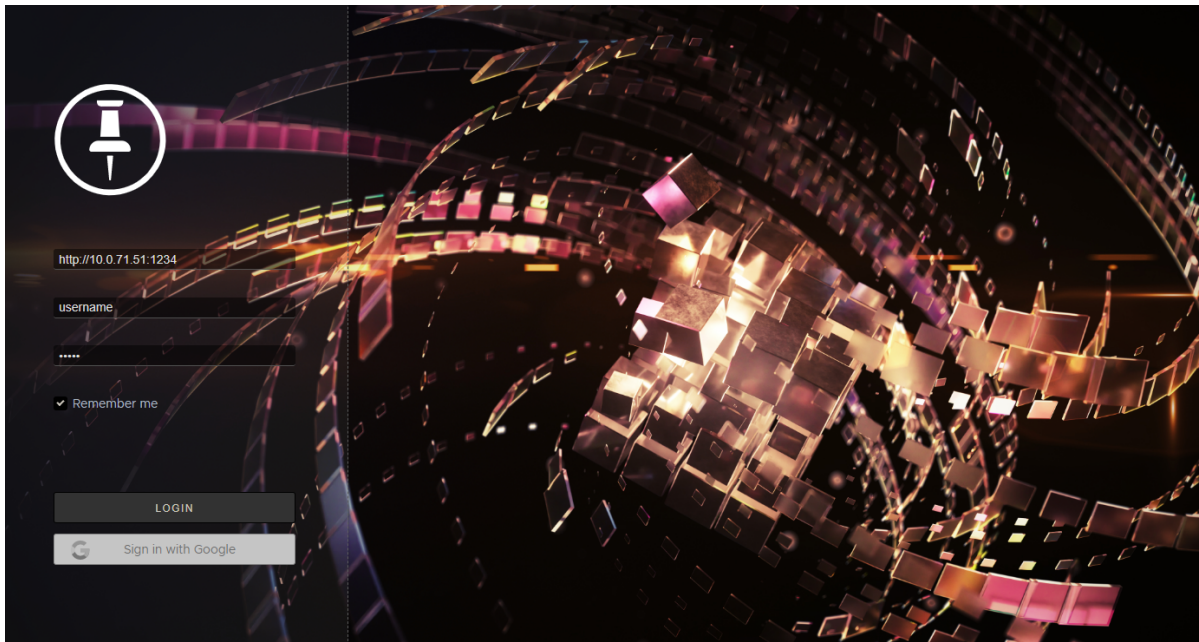
4. In the **Server Hostname** field, enter the server address.

   The server address format is **http://[server_hostname_or_IP]:[port]**

5. 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**.

   > **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.
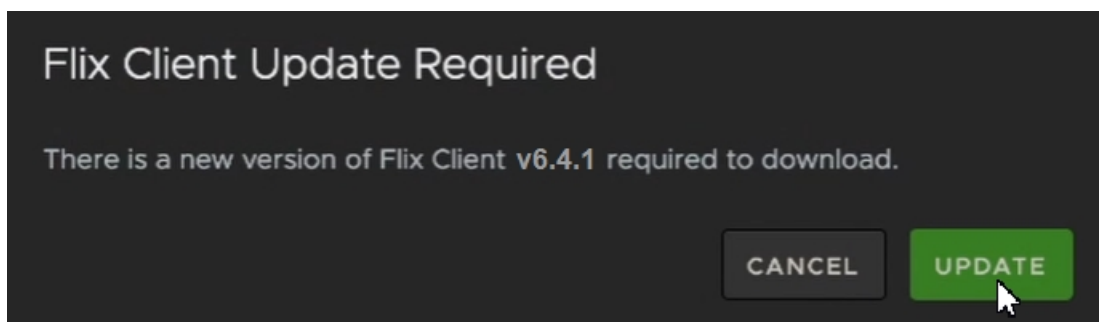
Flix opens at the **Show** level. From here you can open an existing show, create a new one or set user preferences for shows.
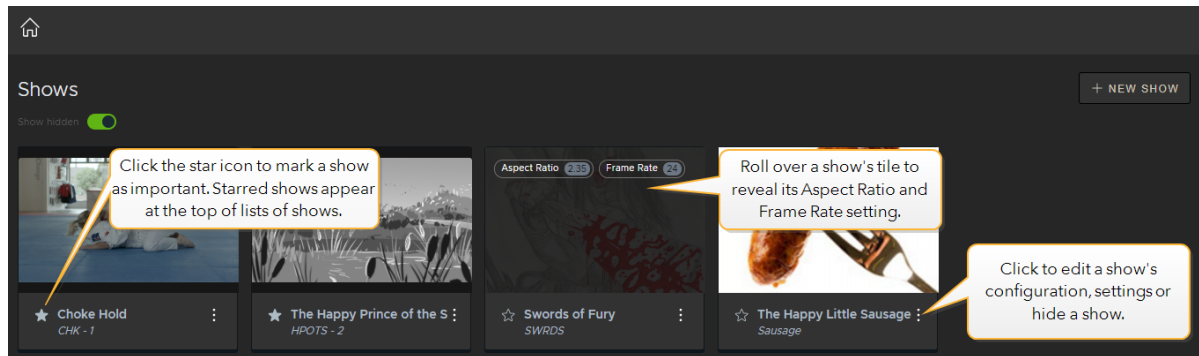
Navigating Through Existing Projects

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.
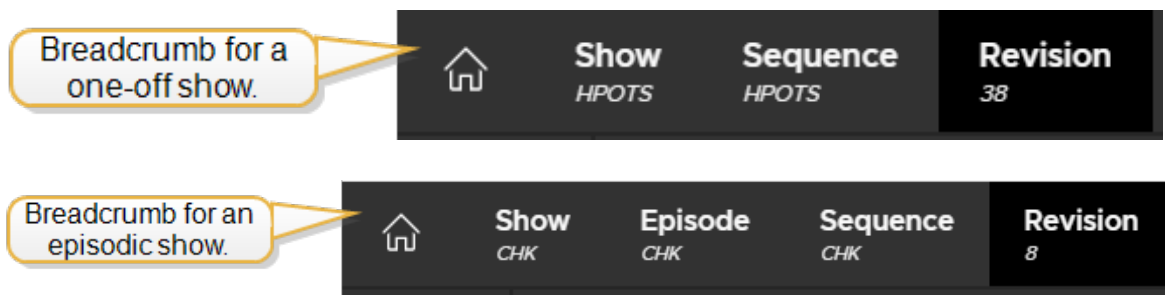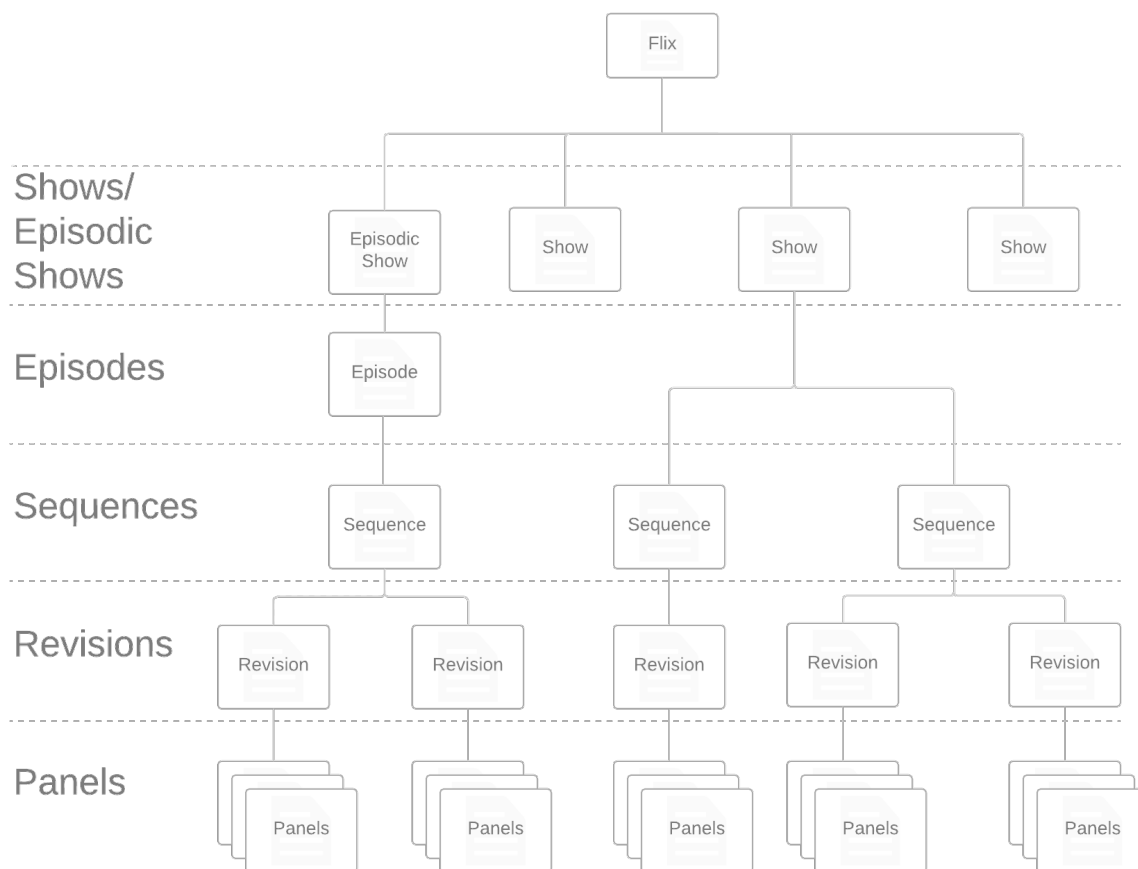
Loading an Existing Sequence

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.

Create a new edit.

Latest revision of a sequence.
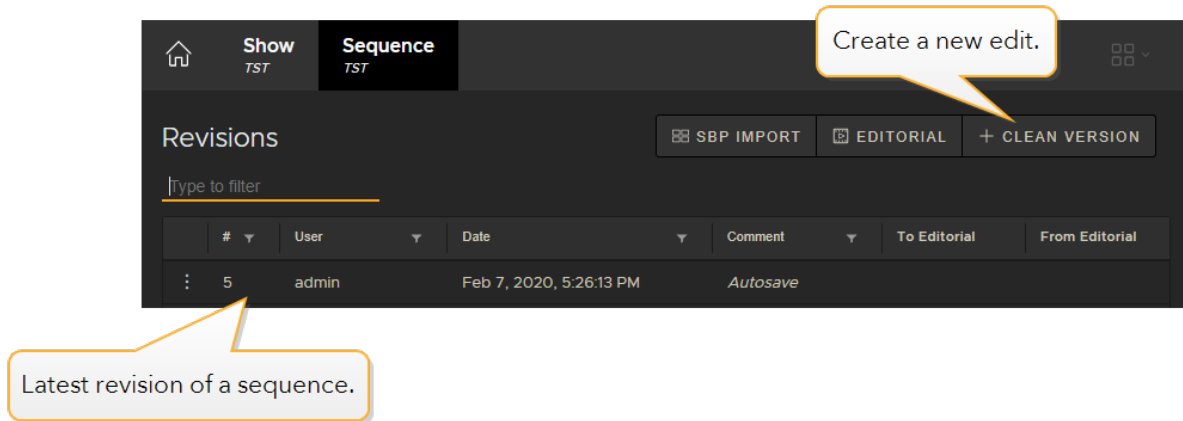
Creating a Clean Version of an Edit

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

1. 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.

Flix for Production

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 more options ⋮ button of 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.
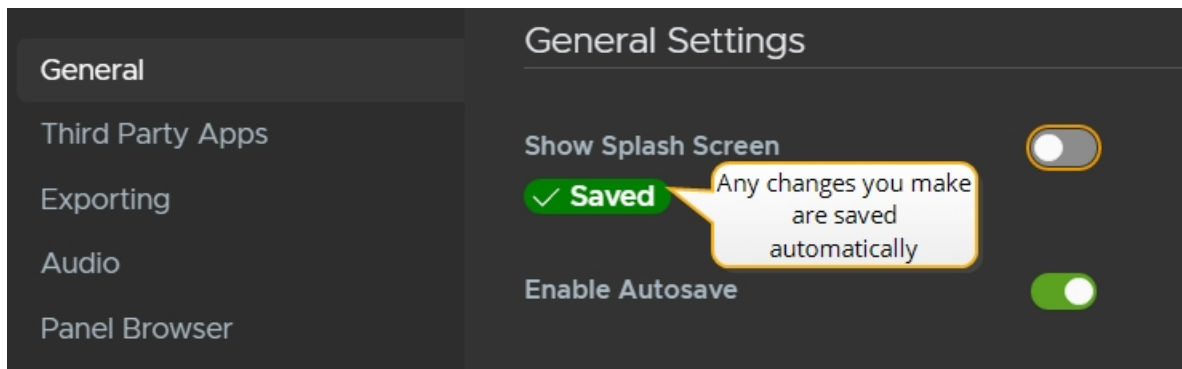
Setting Naming Conventions

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.

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**.

Details	
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	
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.
Aspect Ratio	Sets your show's aspect ratio. Choose from the following ratios: • 1.0:1 • 1.77:1 • 1.85:1 • 2.2:1 • 2.35:1 • 2.39:1
Permissions	
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.

3. Click **Create**.

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

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.
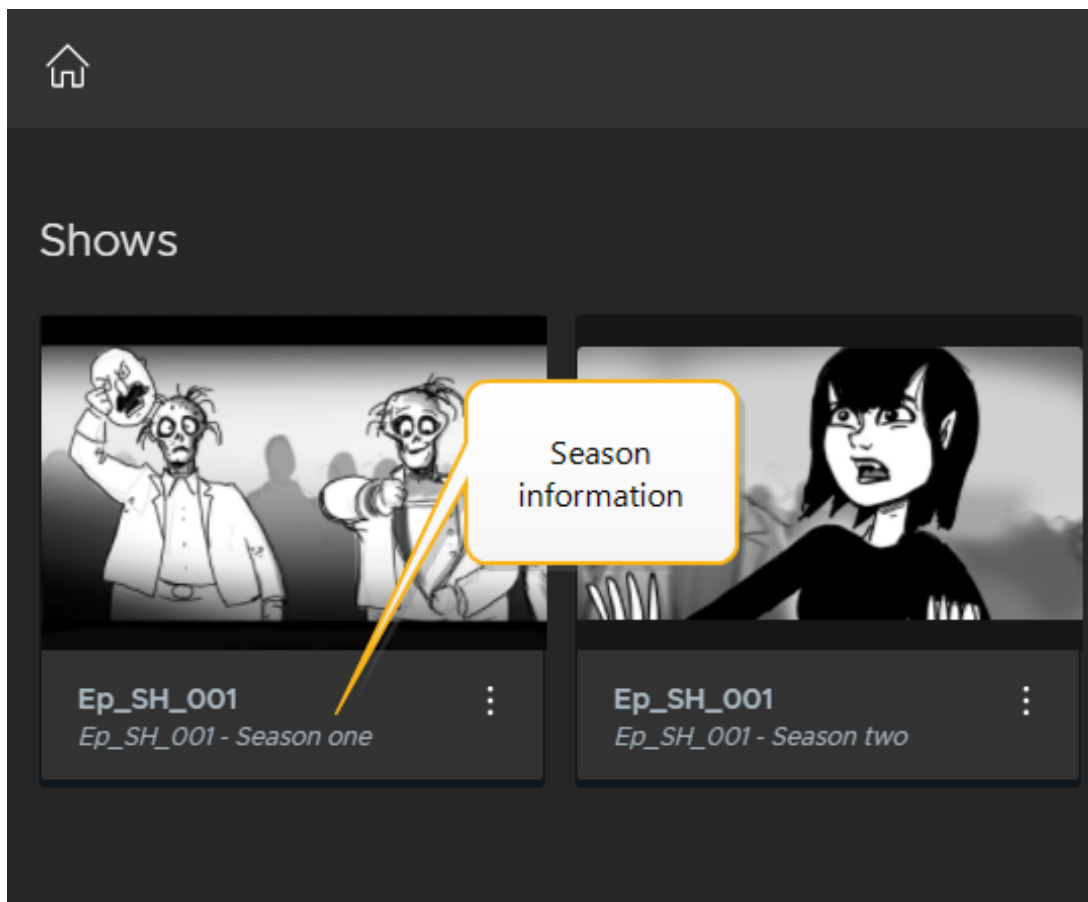
2. Click the more options button and select **New Season**.
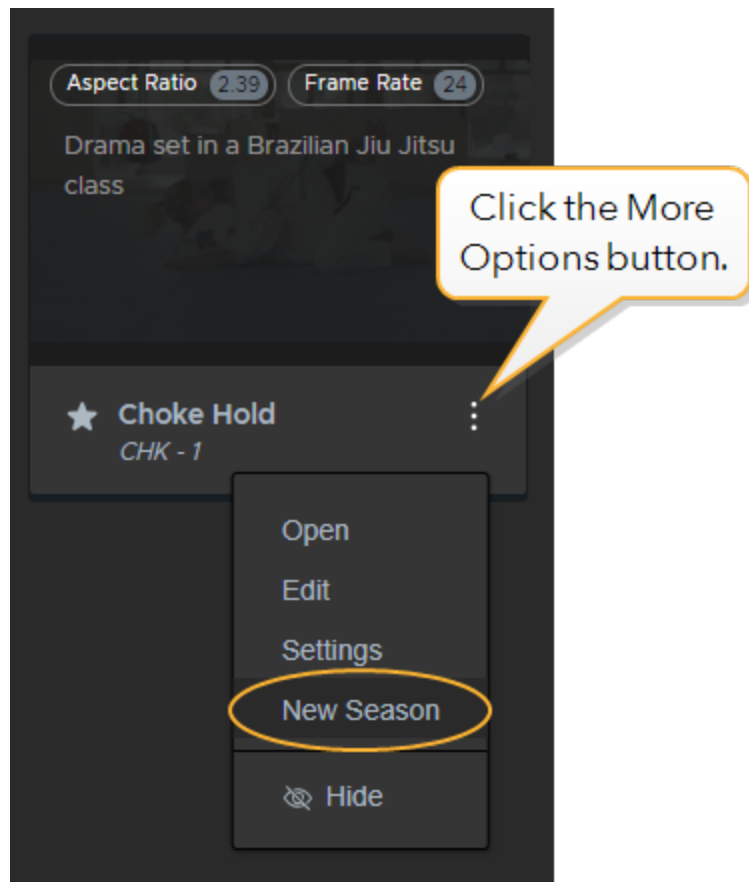
This opens the **Create Show** dialog. The details are pre-filled with the information you provided for the previous season.

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.

Editing a Show

1. Navigate to the **Shows** level in the breadcrumb, click the more options ⁝ button and select **Edit**.
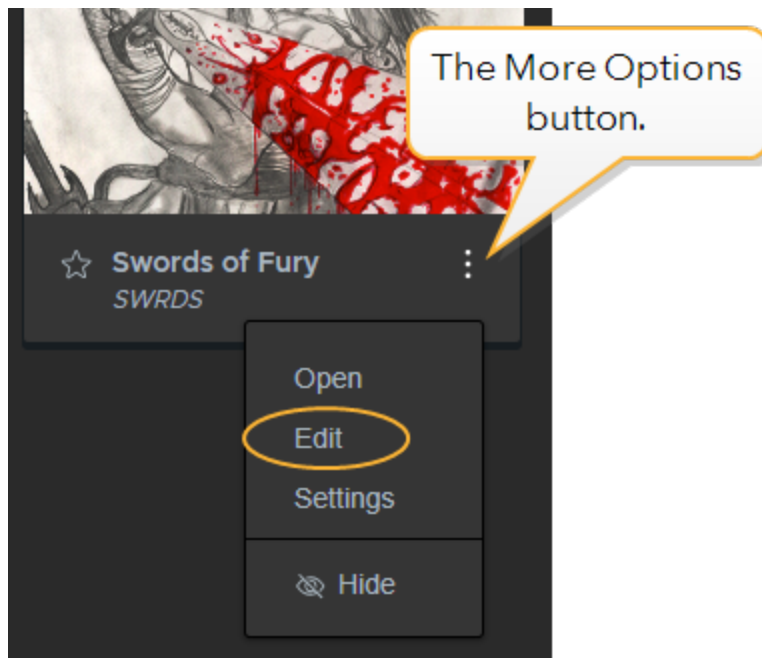
This opens the **Edit Show** dialog.

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 more options ⋮ button and select **Hide**.
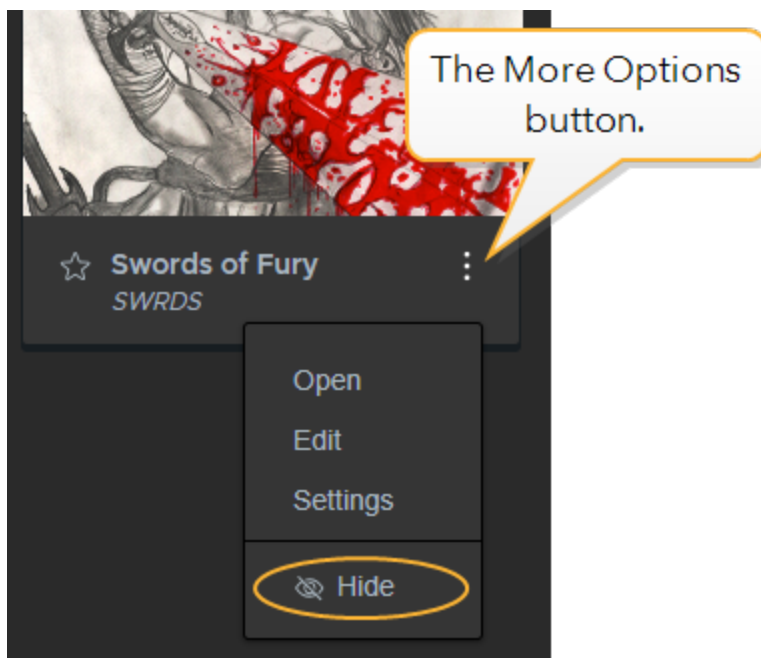
The show is invisible at the Shows level.

> **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 more options ⋮ button 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.

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.

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.

Editing an Episode

1. Using the breadcrumb, navigate to the **Episodes** level and click the more options ⋮ button, then Select **Edit**.
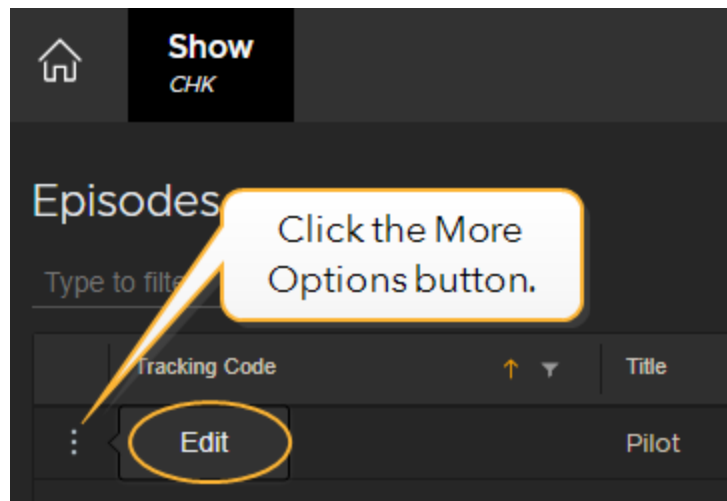
This opens the **Edit Episode** dialog.

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

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**.
The **New Sequence** menu opens.

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.

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.

Editing a Sequence

1. Using the breadcrumb, navigate to the **Sequences** level and click the more options ⋮ button, then Select **Edit**.
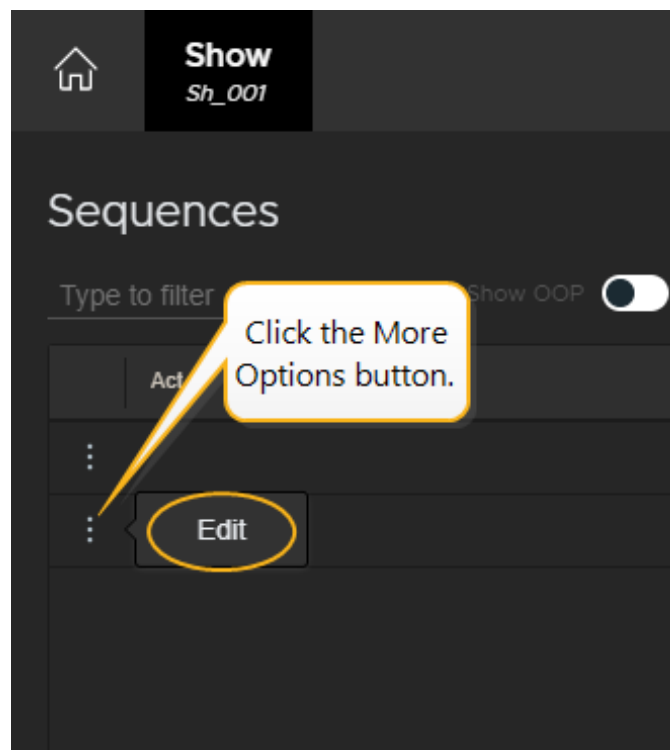
This opens the **Edit Sequence** dialog.

2. Enter the new information for your sequence.

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

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 more options button of the required sequence and select **Edit**.
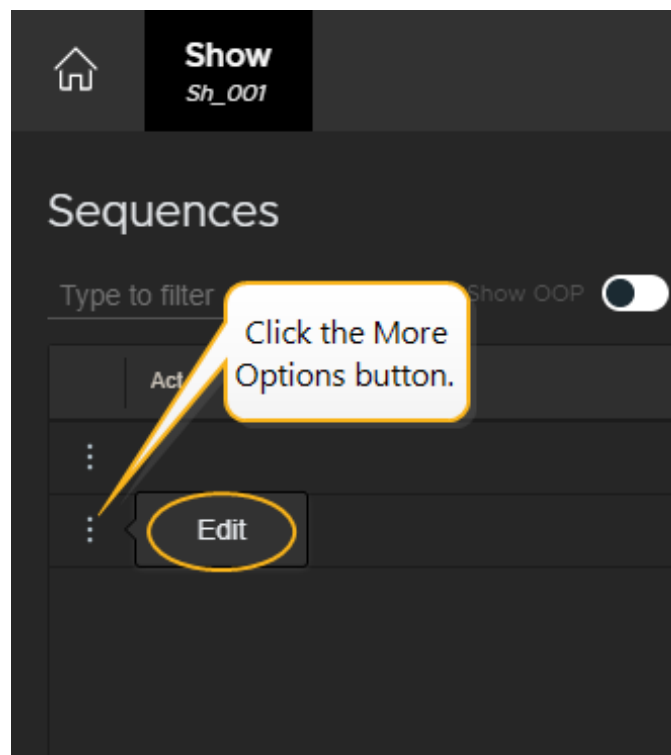
This opens the **Edit Sequence** menu.

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.
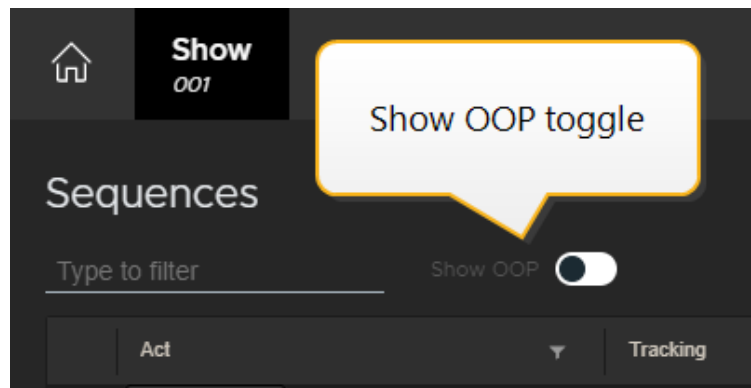
Using the Panel Browser

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.

• 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 [icon] **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 [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.
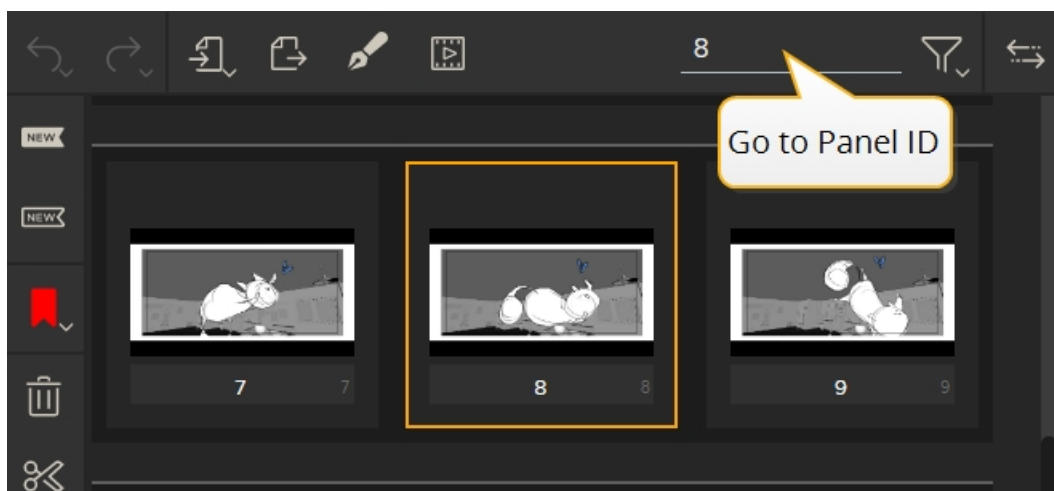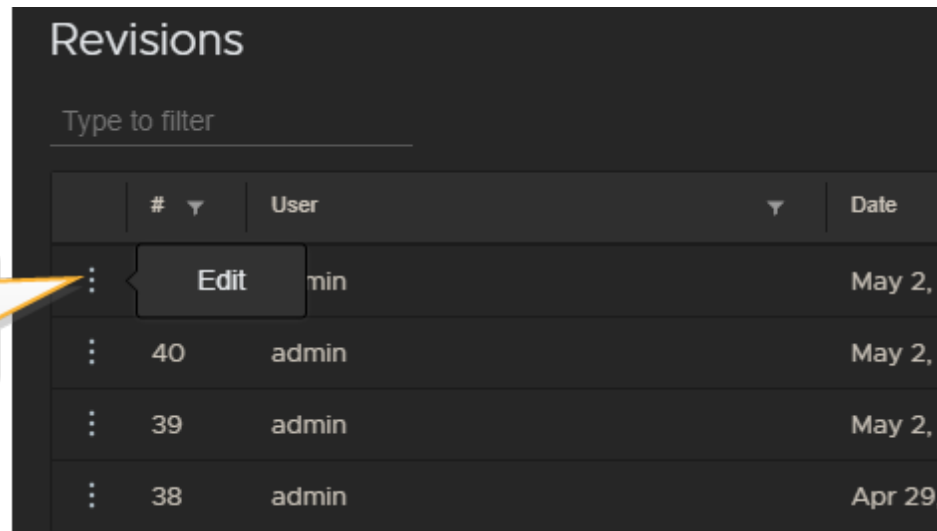
Keeping Edits Organized

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 more options ⠿ button and select **Edit**.

3. Type in the comment "Use for screening" and click **Save**.
4. 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.

Column filters are useful for targeted searches.
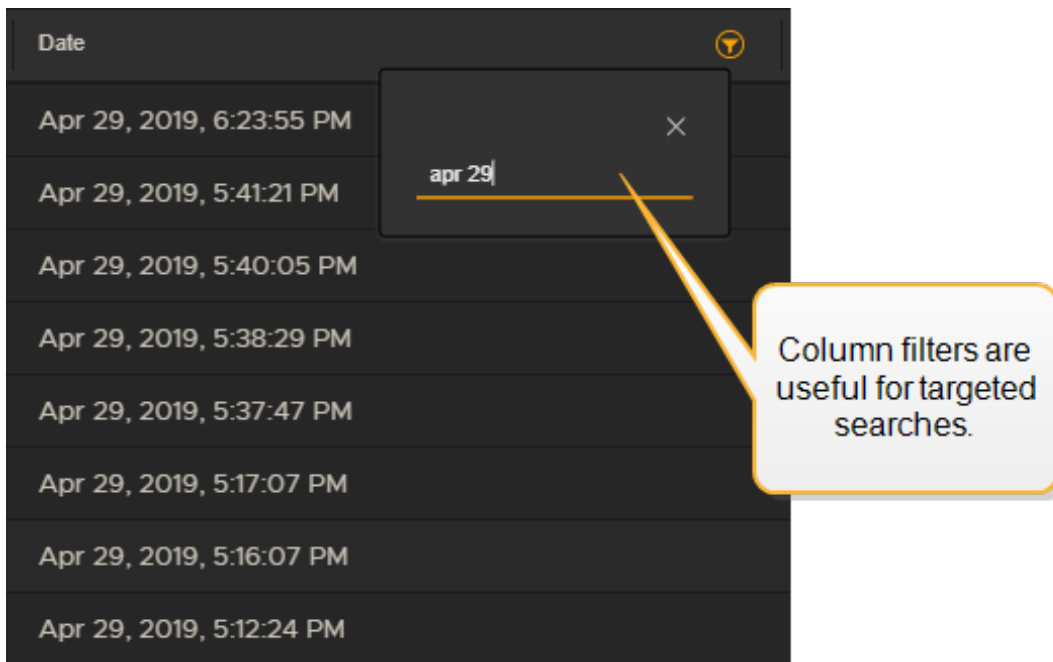
Highlighting Panels

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.
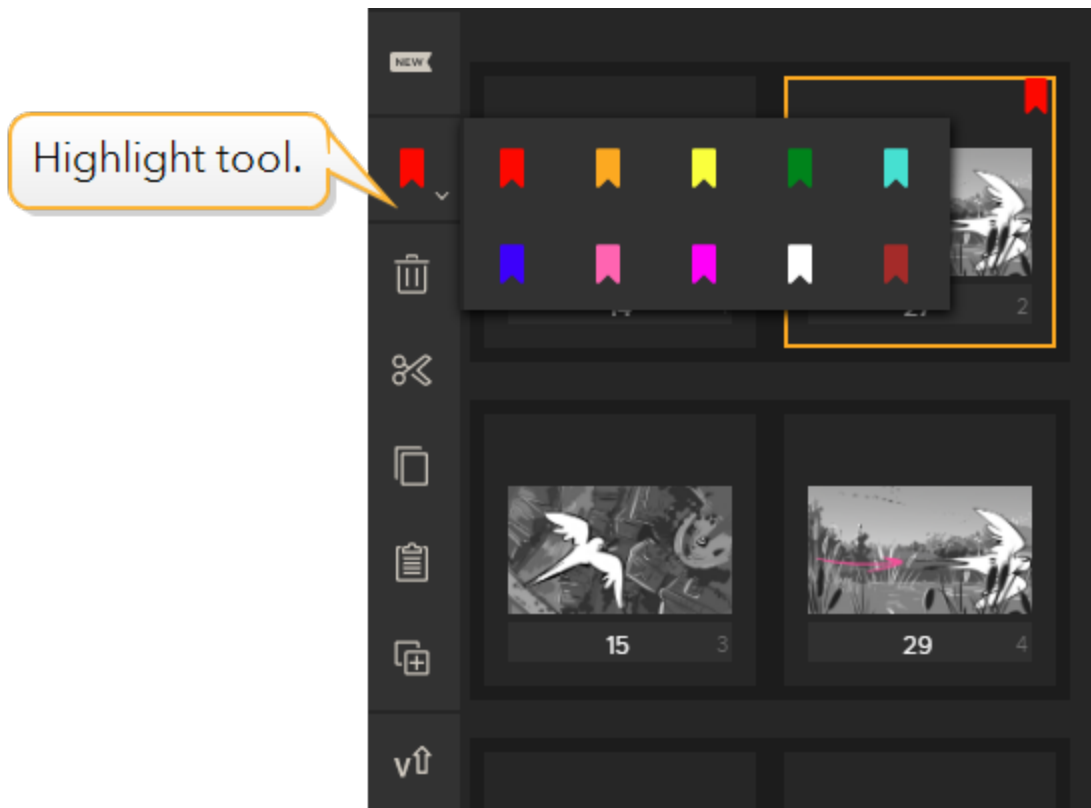
Highlight tool.

To remove highlights:

1. Select the highlighted panels.

2. In the left toolbar, click the Highlight button.

   This removes the highlight from the selection.

Using Markers in Flix

Markers, or locators, are usually added in editorial to delimit shots. These are carried over to Flix in AAFs from Avid and XML files from Premiere, and can be displayed in the panel browser.

Sometimes markers are added to the sequence in Flix. Several panels are often used to make up a single shot, usually to show a character motion or camera move, so a good way to show where a shot begins is to use a marker. If a panel displays a marker, that panel is the start of a new shot and the previous panel is the end of the previous shot.

Configuring Flix Markers

You can configure the naming convention for markers created in your Flix sequence to show information such as shot number and sequence title on a shot.

Setting Marker Names

1. Navigate to **Preferences** > **Panel Browser**.
2. Click on the placeholders below the **Marker Name Format** field to create a naming convention. Click in the **Marker Name Format** field to edit the marker name.

> 📝 **Note:** Your marker name format must include the **[shot_number]** in order to display the shot number.

Setting Shot Number Format

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

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.

Configuring Markers for Adobe Premiere

Markers created in Flix can be configured to be sent to Adobe Premiere as either Clip markers or Timeline markers.

To configure markers for Adobe Premiere:

1. Navigate to **Preferences** > **Third Party Apps** > **Adobe Premiere**.
2. Choose between **Clip** and **Timeline** from the **Marker Type** dropdown menu.

> **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.

• **Clip** markers apply to a whole clip within a sequence and appear at the beginning of the clip.

• **Timeline** markers apply to a particular timestamp in the sequence and appear on the timeline.

> **Note:** Markers created in either Adobe Premiere or Avid Media Composer display in the Flix panel browser at the beginning of the corresponding shot.

> **Note:** If markers are set to display in the panel browser, scene numbers show as markers on sequences imported from Storyboard Pro.

Adding Markers

To display markers in the panel browser:

1. In the panel browser, select a panel and click the **New Marker** button.

> **Note:** If the **New Marker** button is not displayed in the panel browser, navigate to **Preferences** > **Panel Browser** and click the **Markers** option to enable markers.

A marker appears on the selected panel.

2. Double-click on a marker to edit its name directly.
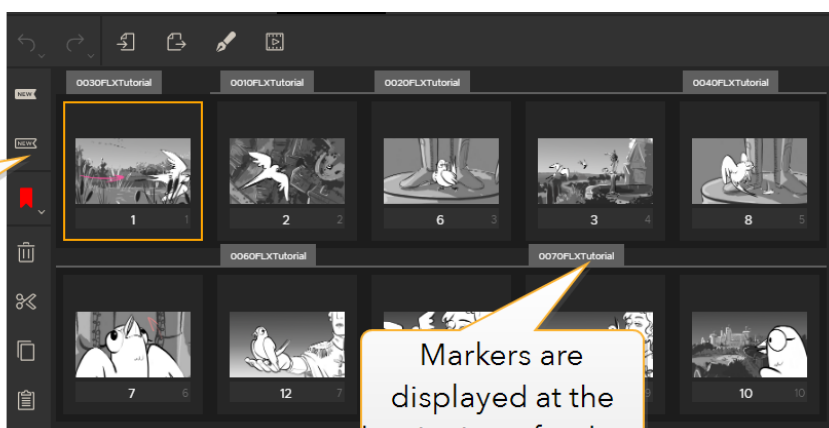
Comparing Edits

1. In your currently loaded edit, in the main toolbar, click the **Compare Tool**.

This opens the **Compare revision** dialog.

2. Select the revision to compare your current revision with and click **Compare**.
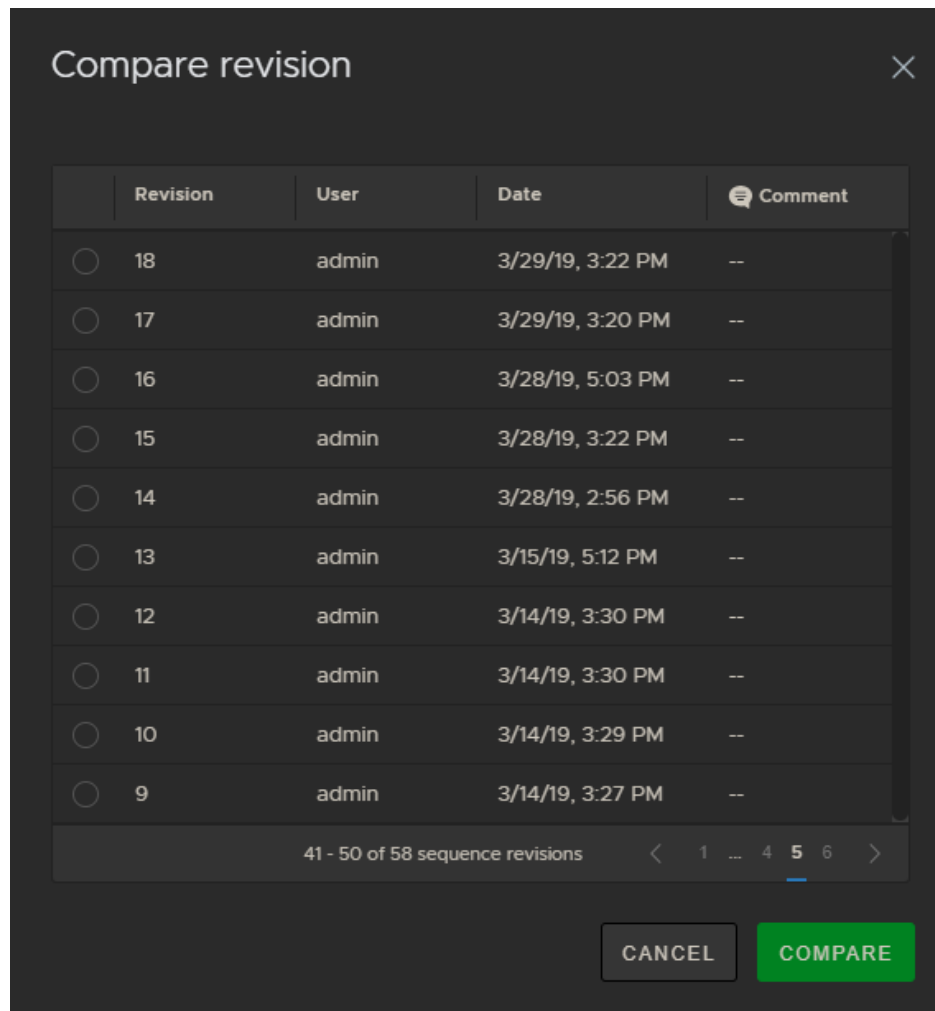
Panels that have been added display a ⊕ icon and panels that have been deleted display a ⊖ icon.

> **Note:** ⊖ icons, which indicate that panels have been deleted, only display when comparing an old sequence revision to a newer one, for example comparing v4 with v10.

3. Click the **Compare Tool** again to stop displaying the icons.

Locating Panels in the Library

You can use the Library to find old panels:

1. In the Panel Properties pane, select the **Library** tab.
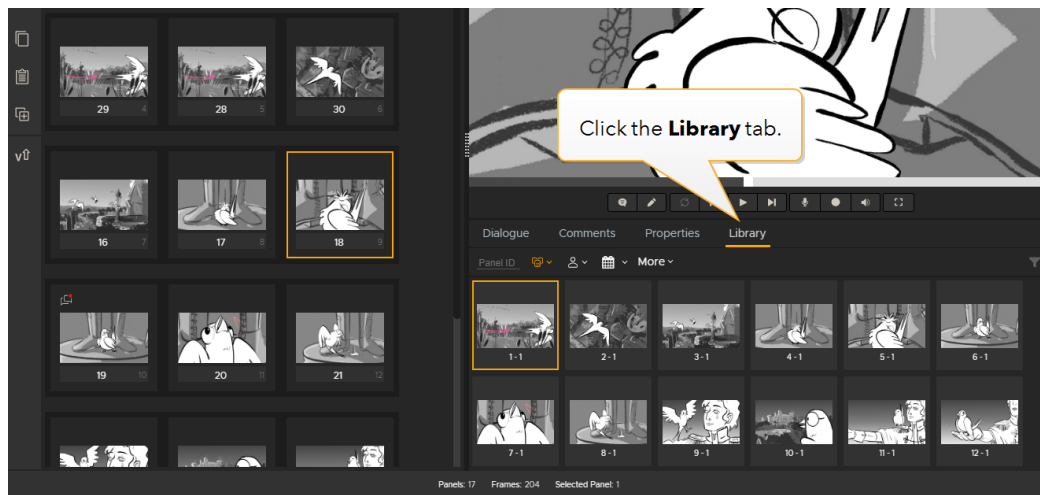
2.   Click the Filter by Sequence button and select a specific sequence or episode.

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.
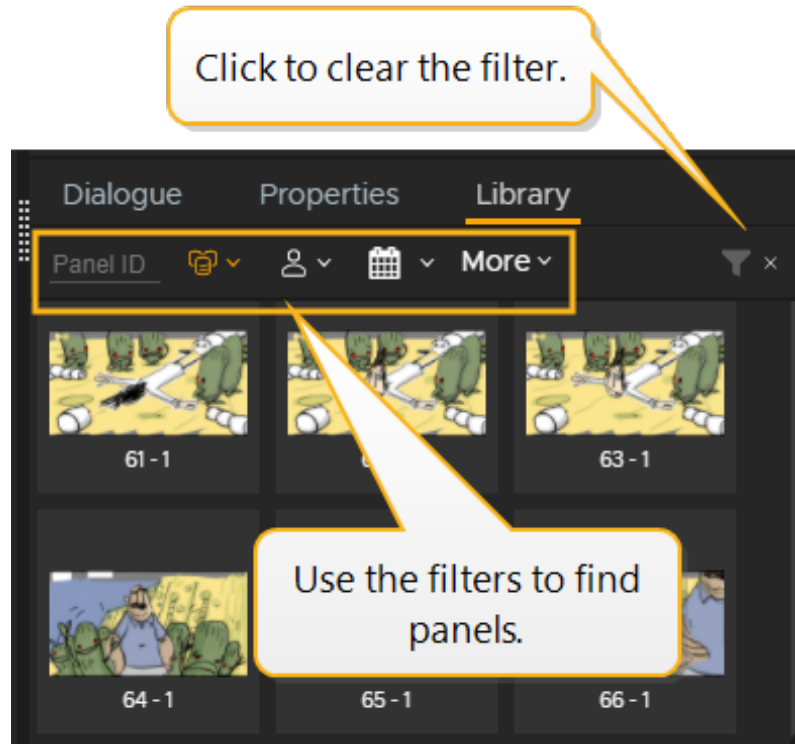
Panel Properties

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

1. Select the panel.
2. Click on the **Properties** tab under the Player.
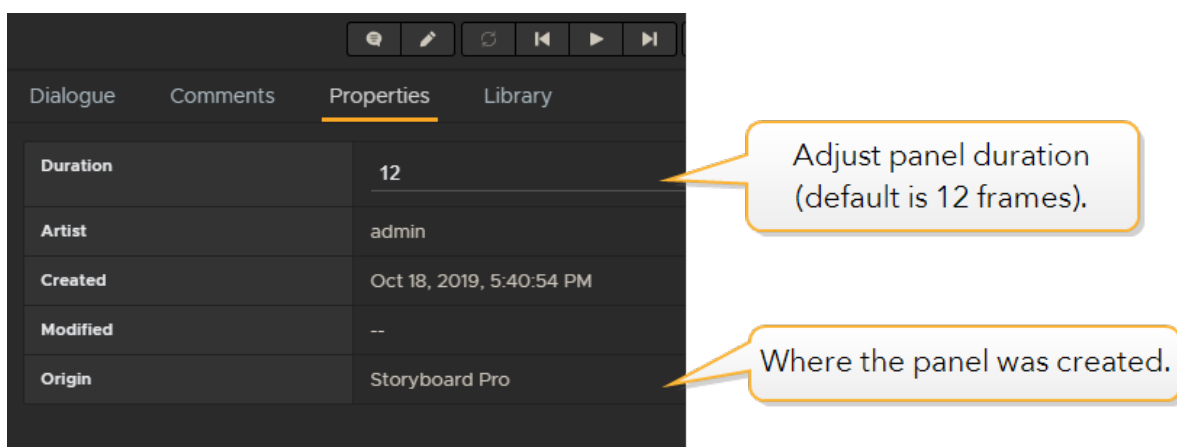
Copying Panels from Another Edit

If you'd like to copy panels from another edit into your current one, you can either open another Flix window and copy the panels to your current edit or drag them from the Library.

> **Note:** Copying panels from another edit retains the dialogue and duration from the original edit, unlike copying panels from the Library.

Method 1 - Using Another Flix Window

1. Navigate to **File** > **New Window**.

   This opens a second Flix window.
2. In the second Flix window, navigate to the required edit and select the panel(s) you want.
3. Click the Copy button.
4. In the original Flix window, click the Paste button.

   The panel(s) are copied after the selected panel.

Method 2 - Using the Library

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, see the previous section, Locating Panels in the Library.
4. Drag the required panels into the edit (sequence revision).

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.
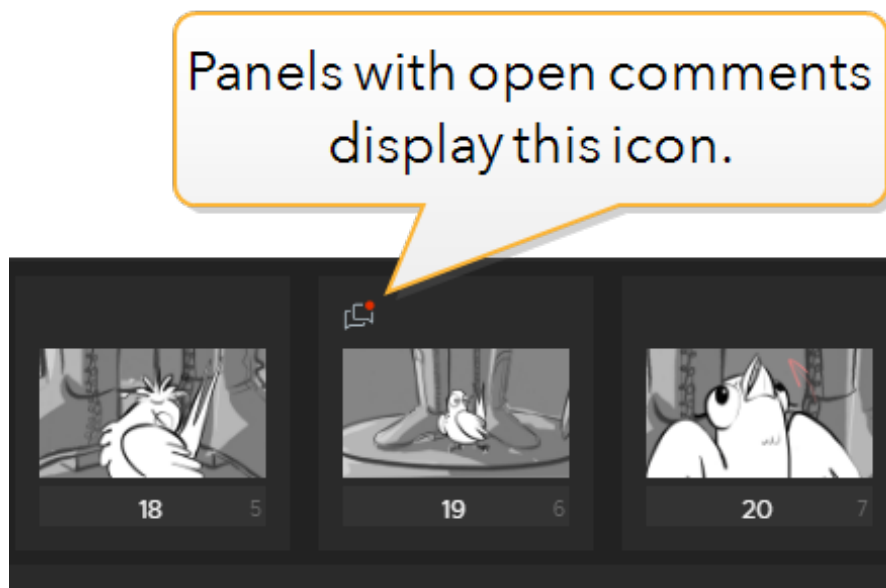
Your comment appears in a feed with any other comments other users have added to that panel.
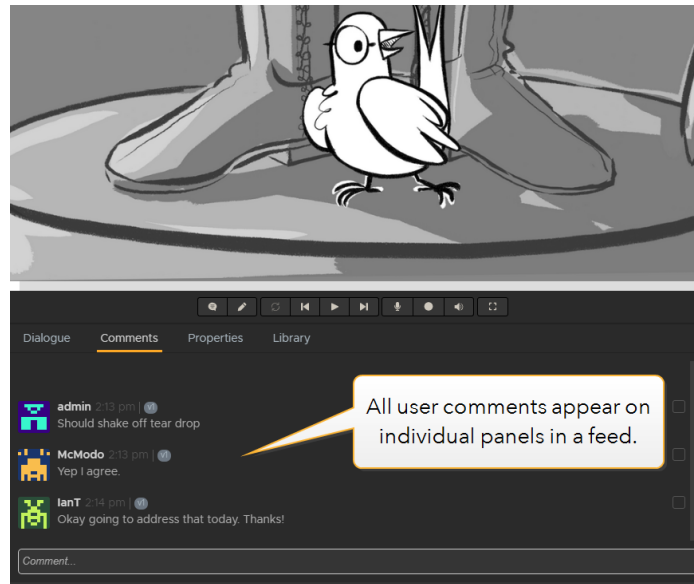
Every comment has a checkbox which allows users to mark as resolved. For example, once feedback in a comment has been addressed, the user would tick the comment so everyone in the production knows that feedback has been actioned.
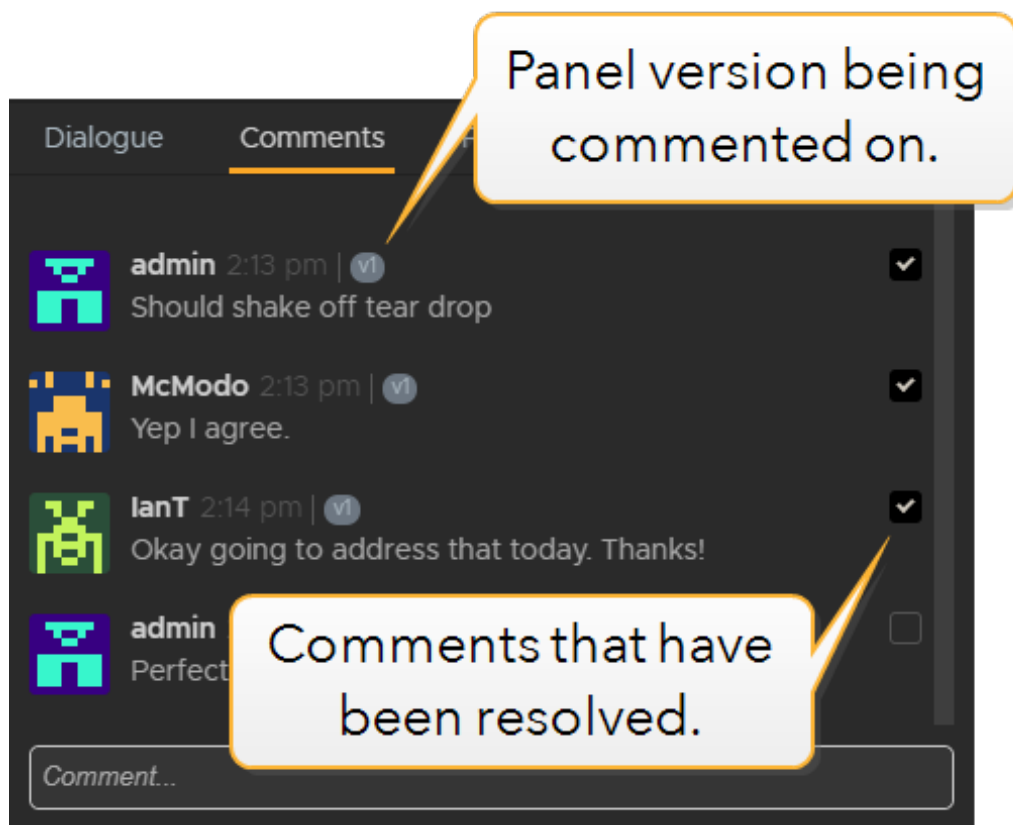
> **Note:** Once all the comments in a feed have been resolved, the comment icon disappears from the panel.

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.
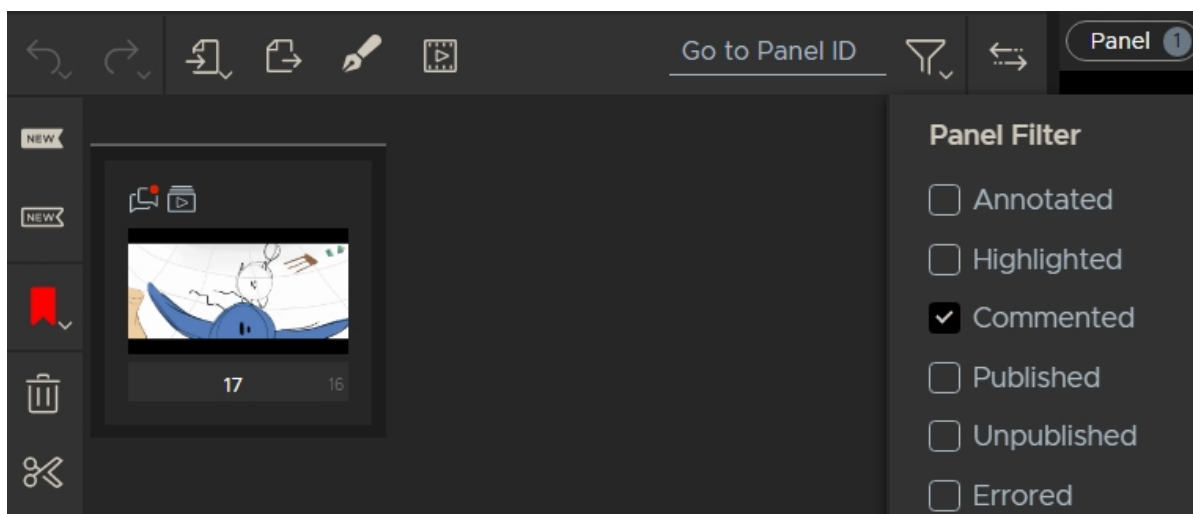
Exporting an Edit as a QuickTime

> **Tip:** Before exporting, navigate to **File** > **Preferences** > **Exporting** to set the directory path and the filename structure for your exported files. See Flix Preferences.

1. Ensure your edit is saved to include all changes in the panels to export.

2. Select the panels you want to export. Don't make any specific selection if you want to export the whole edit.

3. In the main toolbar, click the **Export from Flix** button.

Export from Flix button.

This opens the **Export** dialog.

4. Select whether you want to export **All panels** or **Selected panels**, if you have selected panels in your edit.

5. Select **QuickTime**.
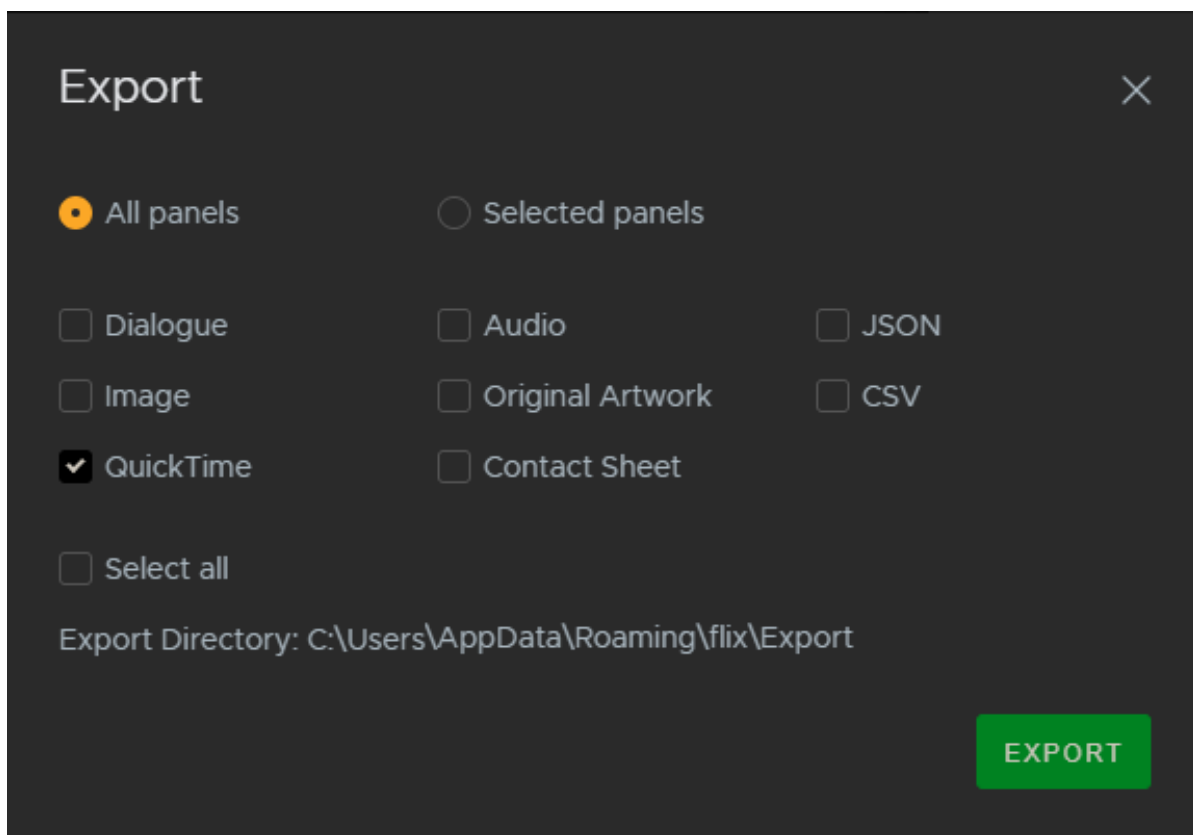
6. Click **Export**.

A file browser opens in the directory containing your exported QuickTime file.

7. Click the **X** button to close the **Export** dialog.

Exporting Dialogue as Subtitles to QuickTime

You can also export your dialogue as subtitles in QuickTime if you have the **Include Dialogue** option turned on in **File** > **Preferences** > **Exporting** > **QuickTime Export**.

> **Tip:** To view the subtitles in your QuickTime export, make sure closed captions are enabled in QuickTime.

Exporting Panels out of Flix

> **Tip:** Before exporting, navigate to **File** > **Preferences** > **Exporting** to set the directory path and the filename structure for your exported files. See Flix Preferences.

1. Ensure your edit is saved to include all changes in the panels to export.

2. Select the panels you want to export. Don't make any specific selection if you want to export the whole edit.

3. In the main toolbar, click the **Export from Flix** button.

Export from Flix button.

This opens the **Export** dialog.

4. Select whether you want to export **All panels** or **Selected panels**, if you have selected panels.

5. Select **Original Artwork** to export the original files imported into Flix and/or **Images** to export the thumbnails visible in Flix.
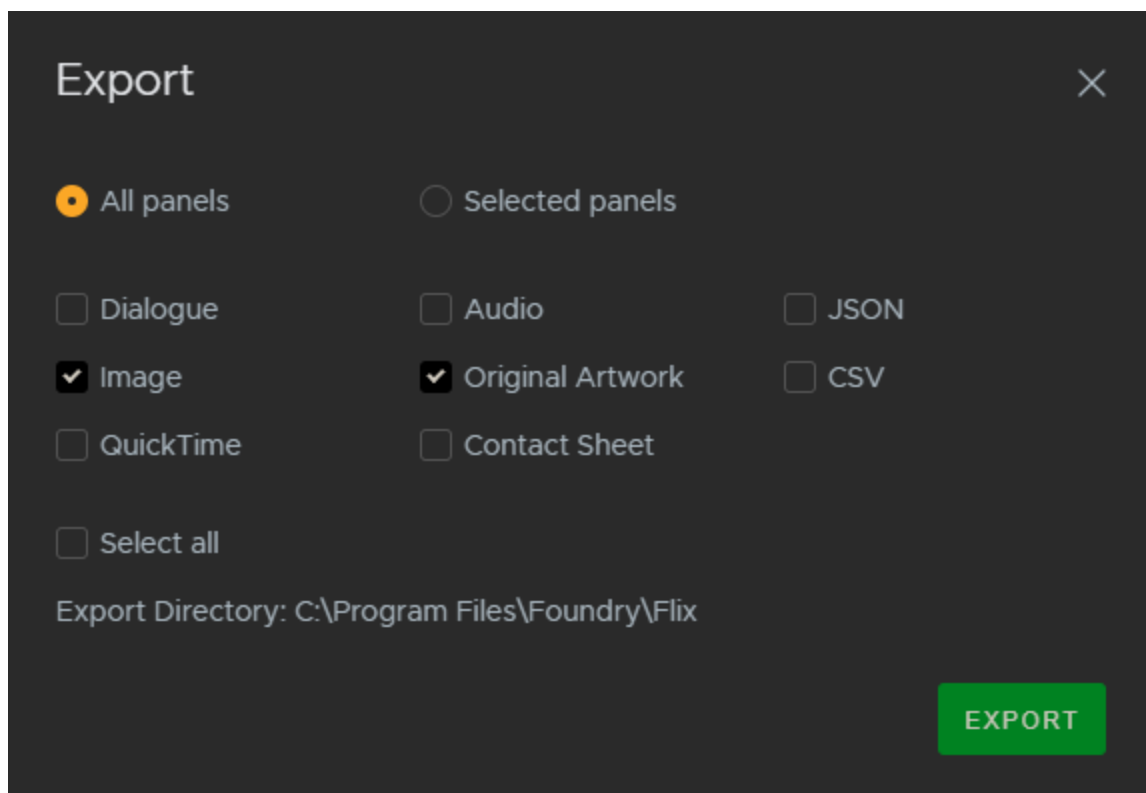
6. Click **Export**.

   A file browser opens in the directory containing your exported panels.

7.   Click the **X** button to close the **Export** dialog.

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.

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 for Story

Top Five Things to Onboard New Story Artists

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

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.

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 **JPG**s, **PNG**s and **TIFF**s you've already created, and **PSD**s from

Photoshop or **.sboard** and **.sbpz** (packaged project) 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.
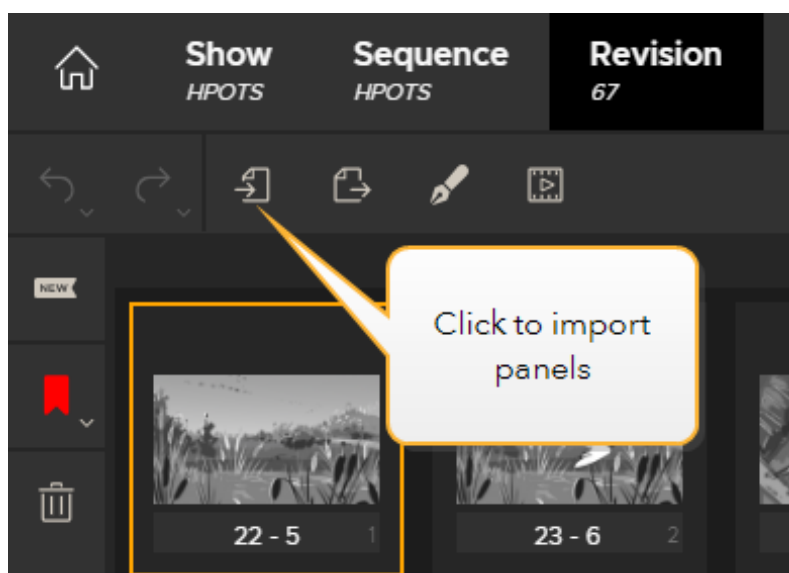
Importing PSD Files

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 & Photoshop for more information or check out the Flix & Photoshop online video course.

> 📝 **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.
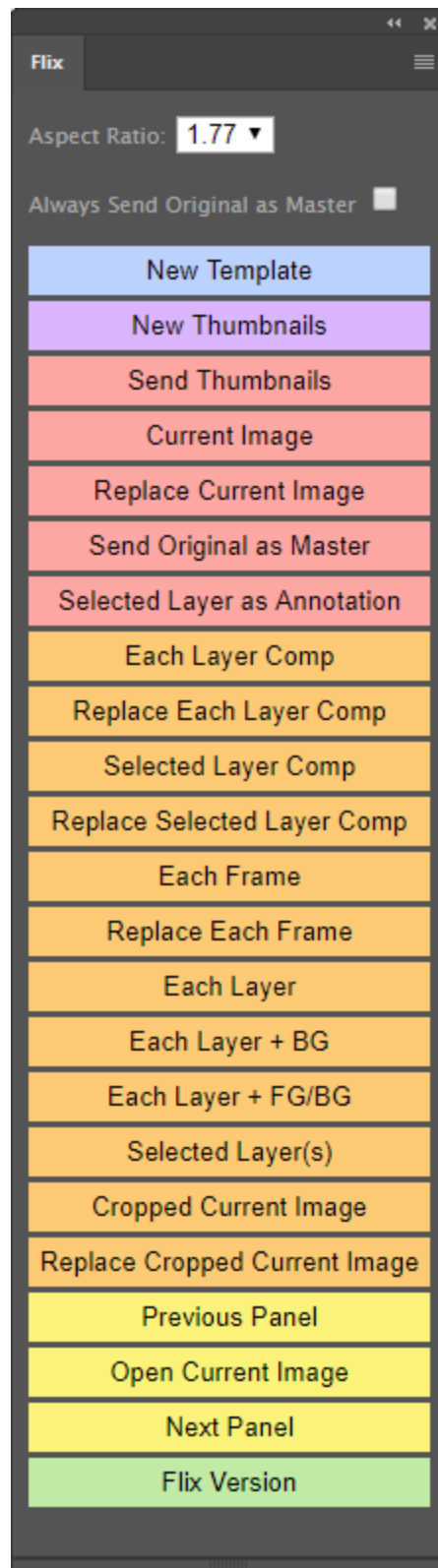
Flix/Photoshop toolset

Check out the video below for a brief overview of how Flix and Photoshop work together.

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.

> **Note:** Importing **.sbpz** files provides an optimized experience using less storage and reducing backup issues or loss of data.

See Flix & Storyboard Pro for more information or check out the Flix & Storyboard Pro online video course. For more information on importing panels and Flix workflows with Photoshop and Storyboard Pro, see https://learn.foundry.com/flix#story.

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** > **General** 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.

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** > **General** 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**.
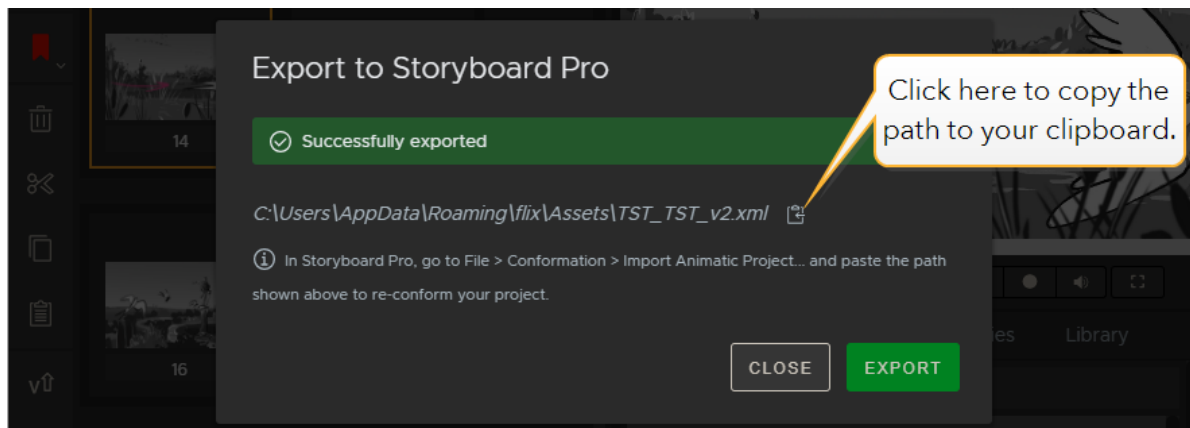
You can manually import the **XML** into Storyboard Pro, which reconforms the sequence with all the latest changes you made in Flix.

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.
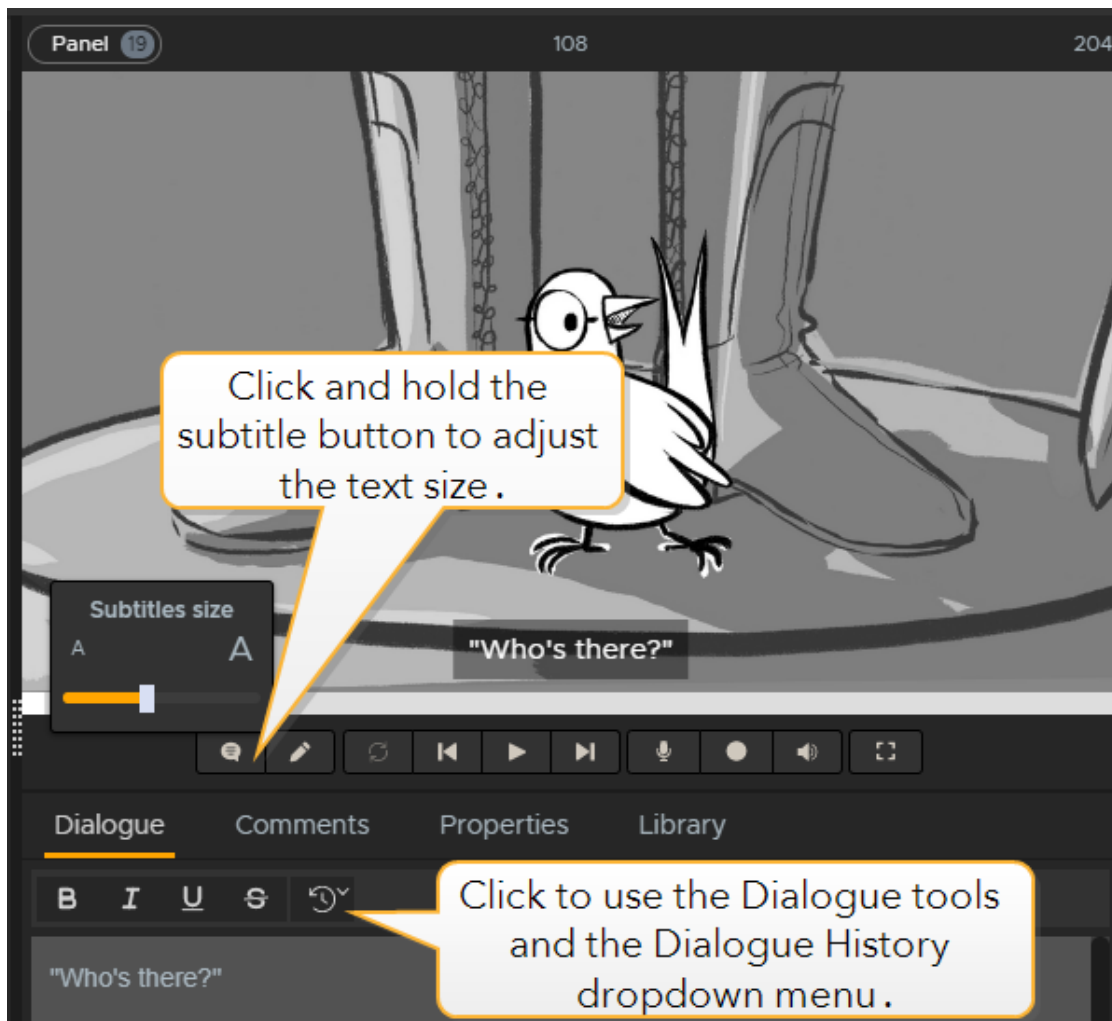
Note: You can also select multiple panels and enter text in the Dialogue tab to have that text applied across all selected panels.

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**.

   Flix plays through the sequence.

2. Record your audio in time with the sequence revision.
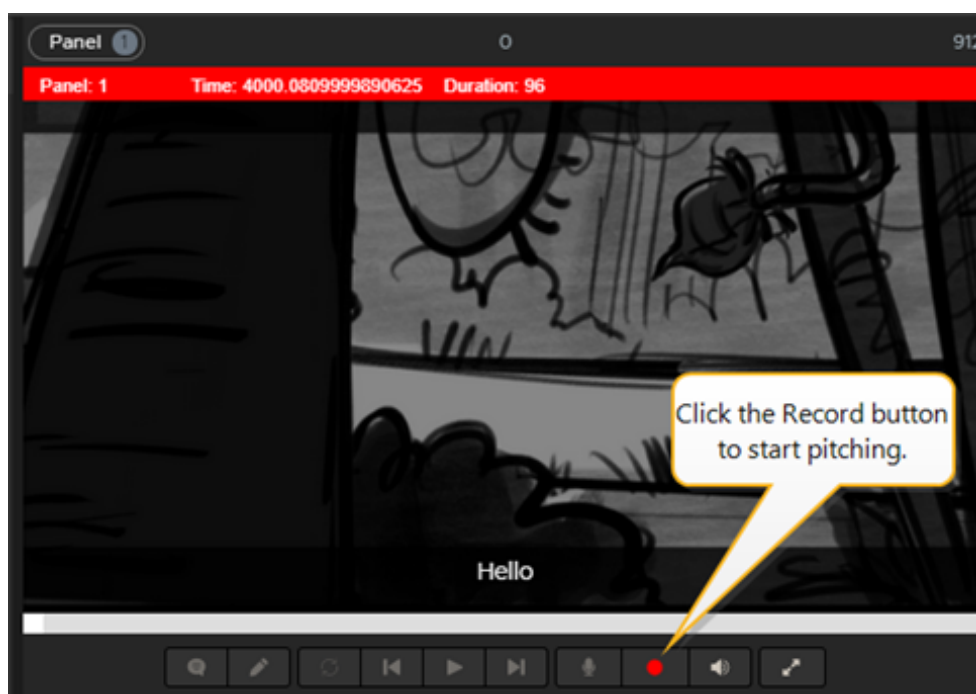
> 📝 **Note:** You can use pre-recorded audio to add background music or other sound effects by importing **MP3** or **WAV** files, which are added in time with the panels.

> 💡 **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.

Check out the video below for a demonstration on using Flix's pitch mode.

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.
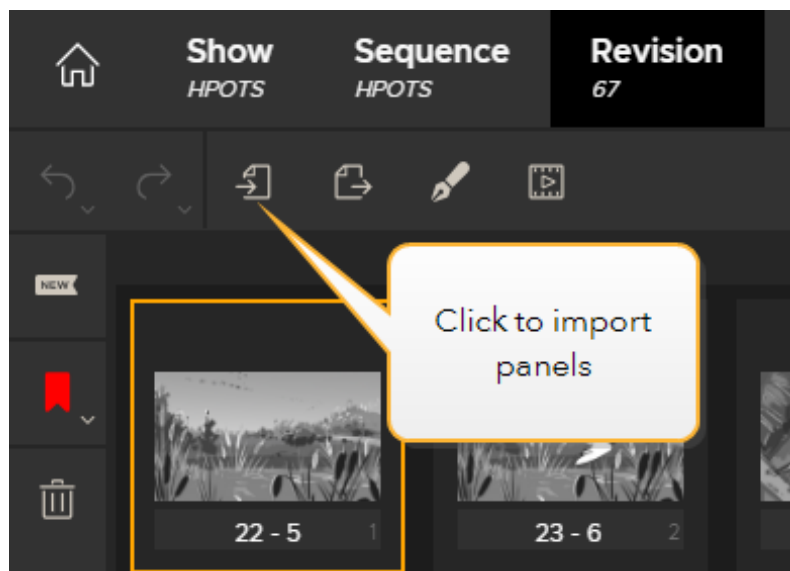
This opens a file browser.

2. Select one or multiple files.

> 📝 **Note:** You can import the following files: **.png**, **.psd**, **.jpeg**, **.jpg**, **.mov**, **.mp3**, **.ogg, .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.

> ⚠️ **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 process the new ones.

When audio, such as **.mp3** or **.wav** files are imported, Flix displays an audio icon in the Status bar.
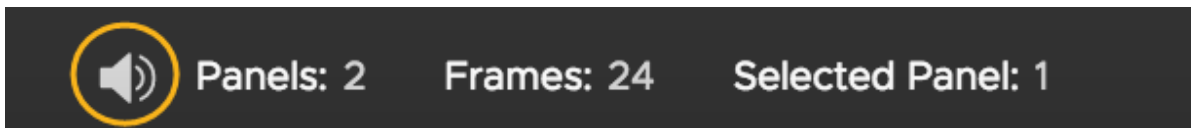
> 📝 **Note:** If you import a **.mov** file to the panel browser, any embedded audio contained in the file is not retained. Please see Flix for Editorial for details on importing movie files from editorial with audio.

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.
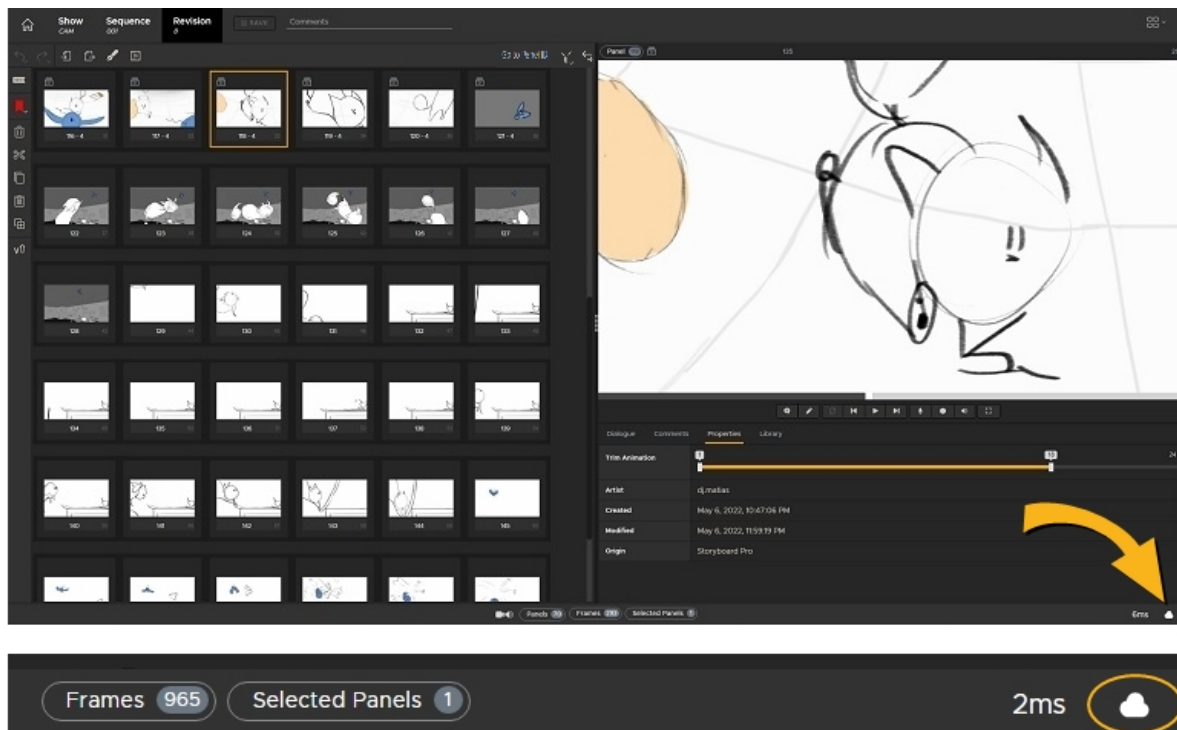
The **Total Transfer** progress is shown when the Transfer Queue is closed, giving you an overall picture of the state of your file transfers.
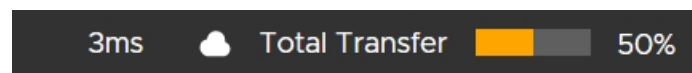
Clicking the ☁ cloud icon opens the queue interface. At the bottom of the window, the **Expand/Collapse** allows you to quickly check on the status of running and failed imports or collapse all entries in the queue.
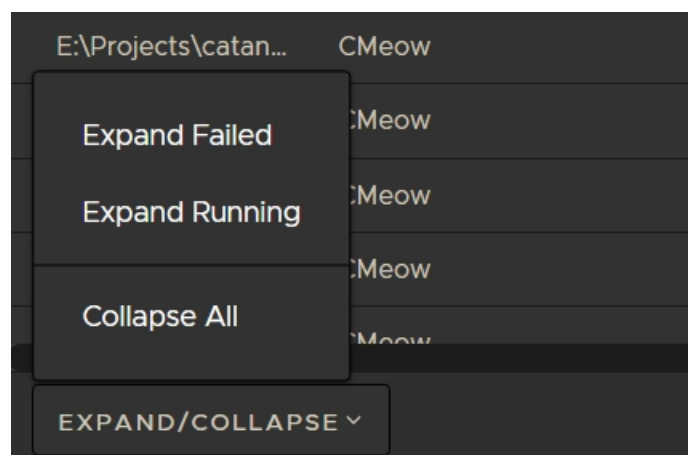
Expanding entries displays detailed information on individual files in the transfer, such as **Show**, **Panel ID**, **Progress**, and **Status**.

> 💡 **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:
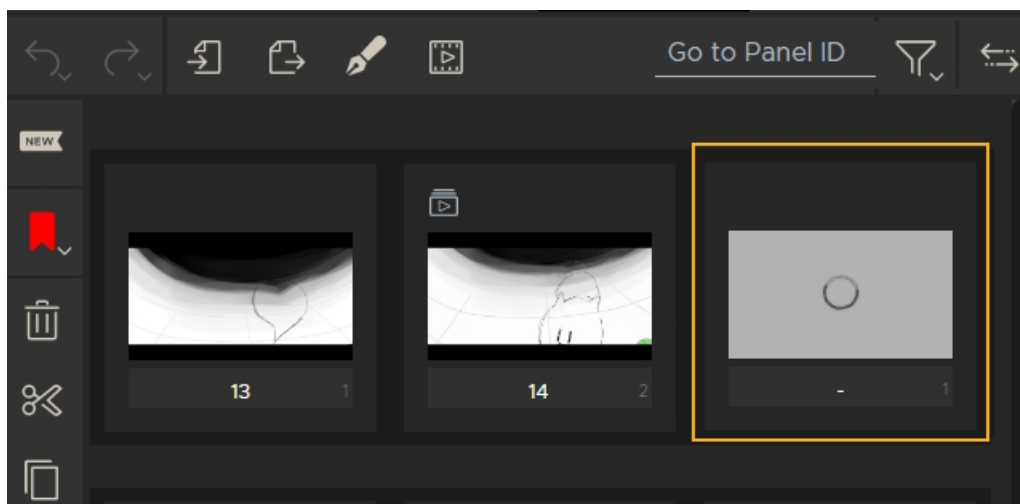
Transfers that display a **Status** of **Failed** are displayed in the panel browser with an exclamation point and are highlighted red in the **Panels** count at the bottom of the Flix Client.
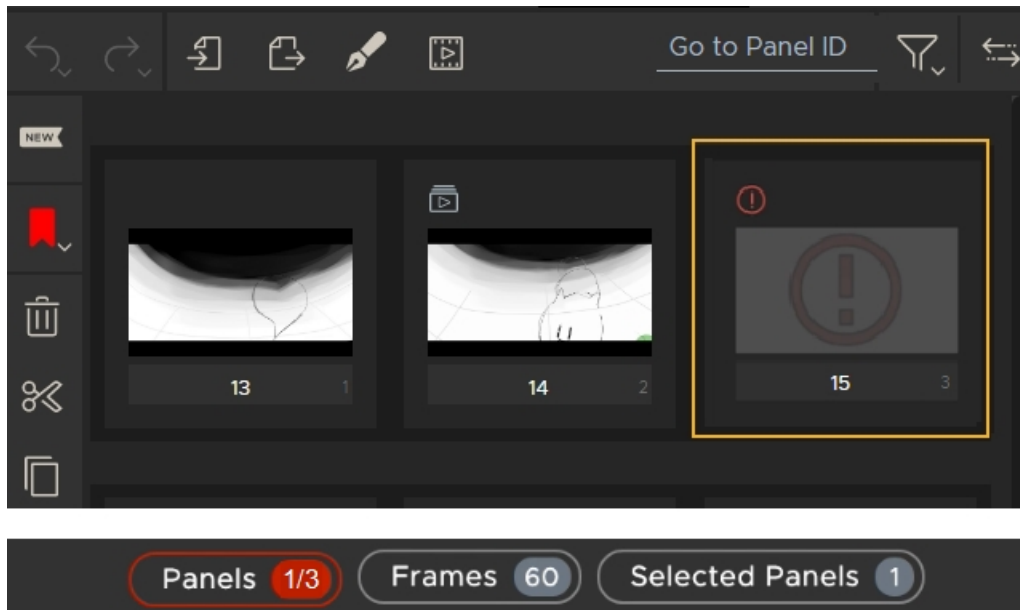
If a panel fails to import correctly, you can delete the panel and restart the import or look at the logs to determine the cause of the failure. See Transfer Logs for more information.

> 💡 **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:

Function	Conditions
Workspace	

Resizing, menu operation, Preferences, and so on	Always enabled.
Panel Browser	
Reorder panels	Always enabled.
Add panels	Always enabled.
Markers	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. **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 Storybaord 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	
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	
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.

Warning: Possible data loss

This will cancel any ongoing transfers, and discard any unsaved changes.
Are you sure you want to leave?

STAY   LEAVE
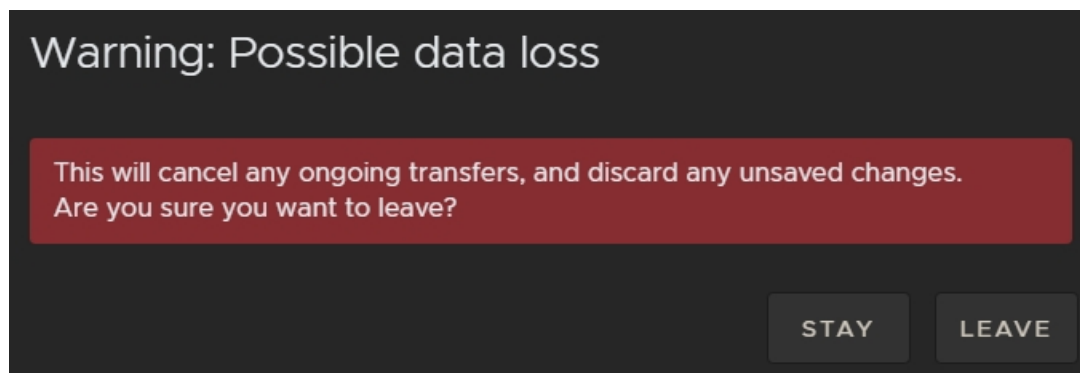
- Reloading the Revision to which files are being transferred, and

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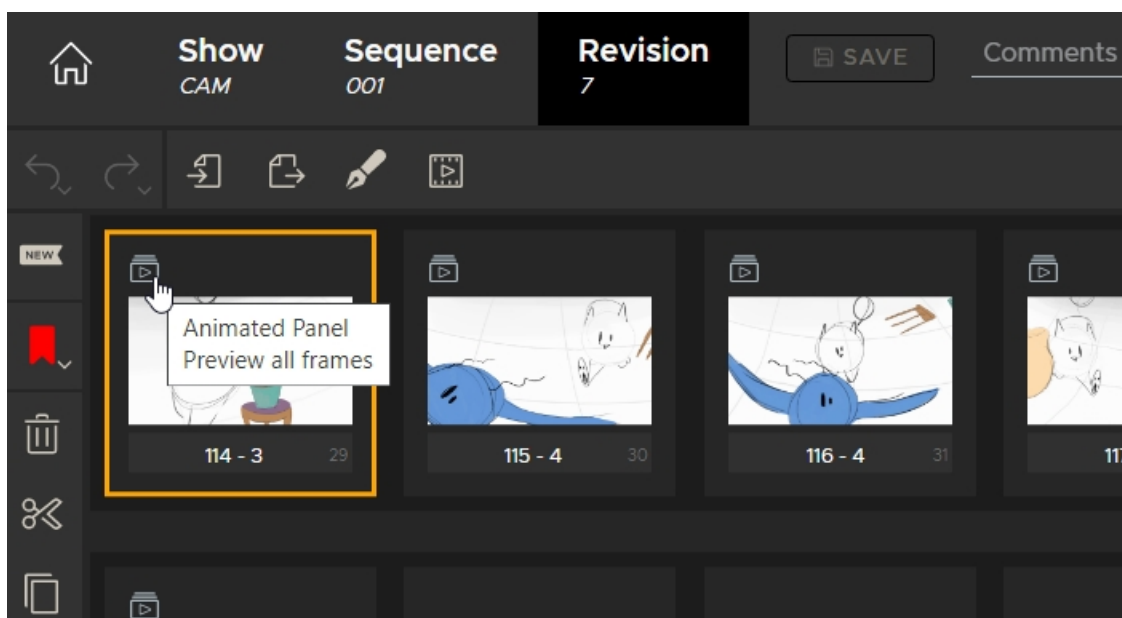
- **Preview all frames** - Animated panels show all available frames in the panel viewer.
- **Preview first frame** - Animated panels only show the first frame in the panel viewer and the **Trim Animation** control is disabled.

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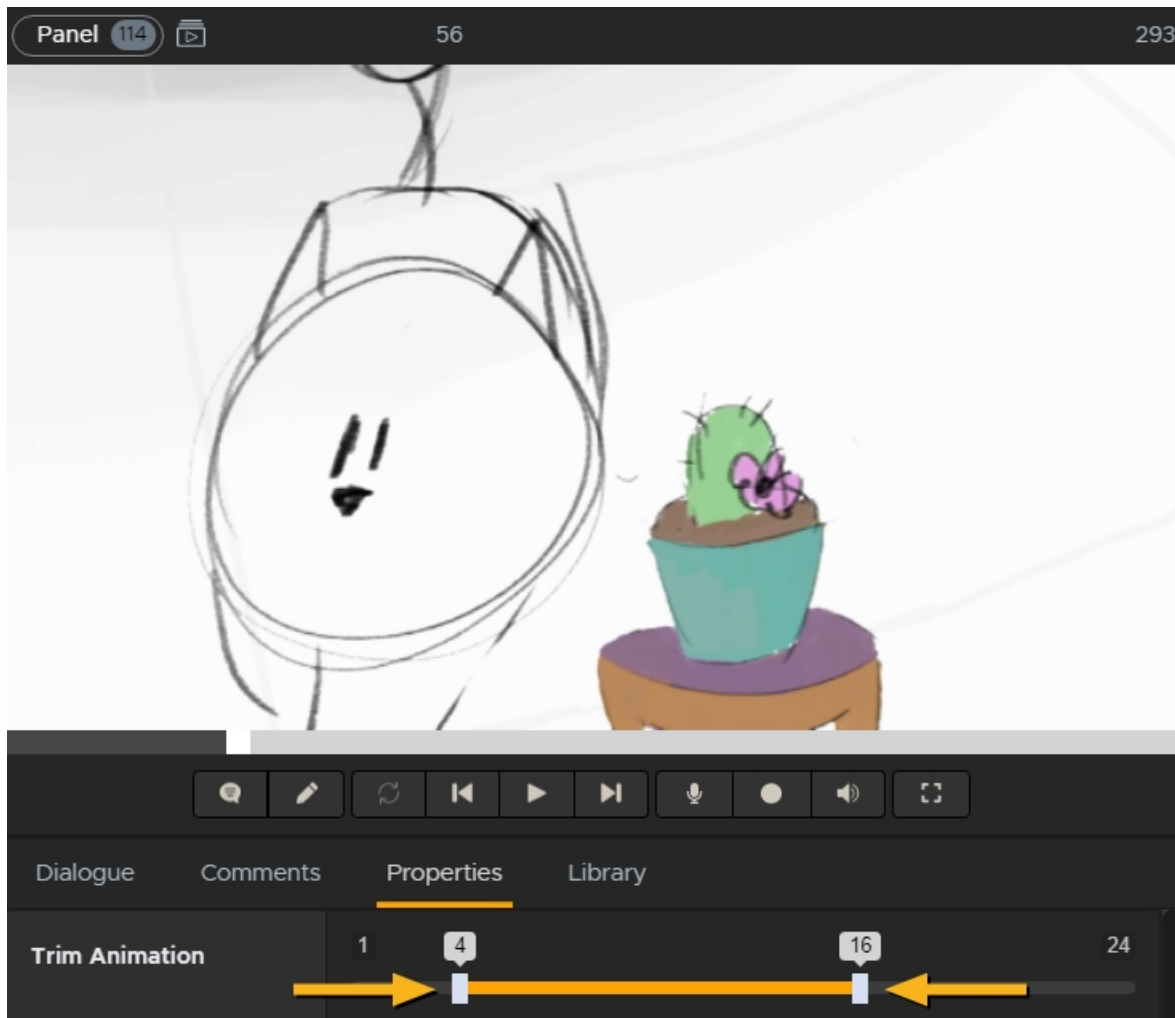
This trimmed animated panel only plays between frames 4 and 16.

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.

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.

• 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 Annotations.

• Adding and versioning dialogue.

> **Tip:** Duplicate selected panels with the [icon] **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 & Photoshop 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.
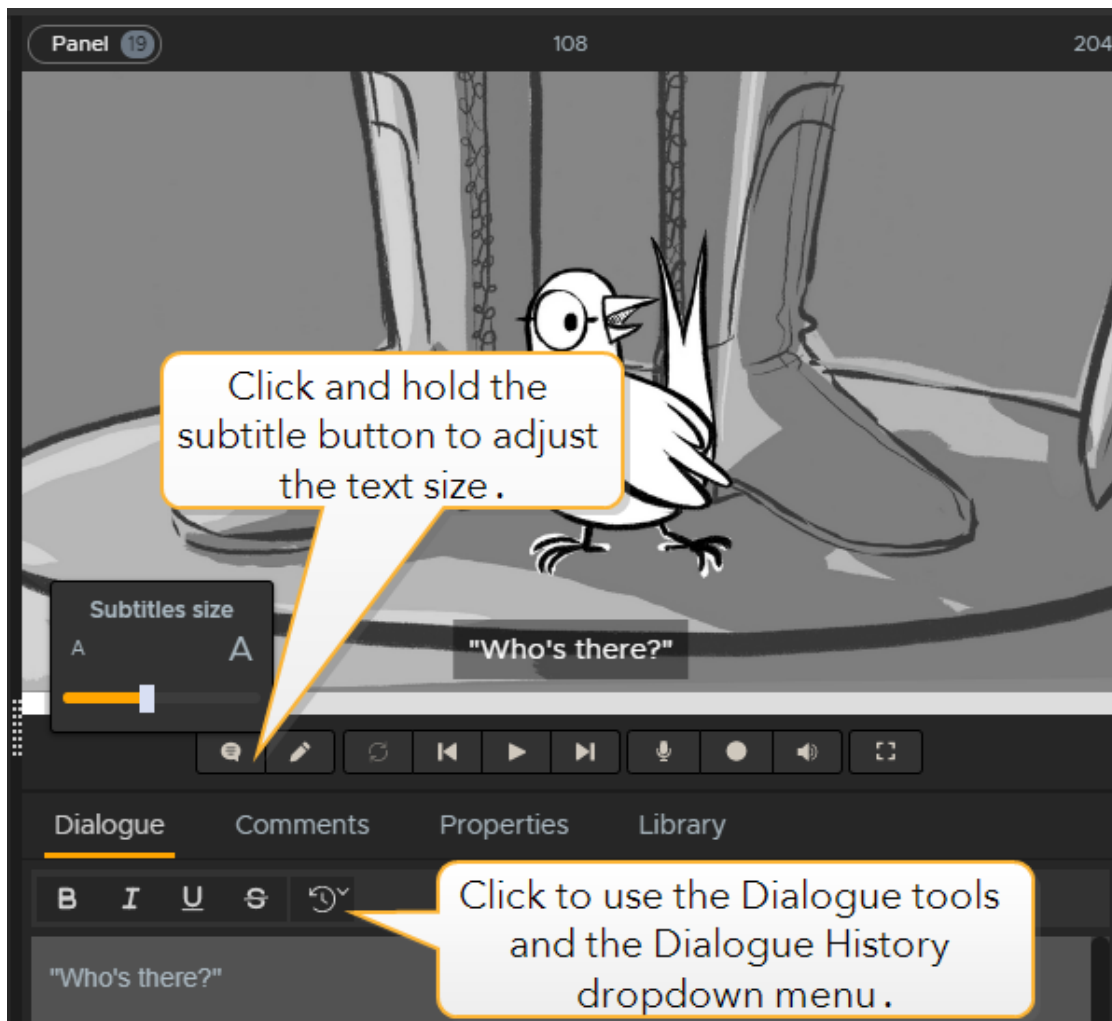
💡 **Tip:** Use the Undo and Redo buttons, located under the breadcrumb, to undo or redo your recent actions.

4. Click **Save** to save all new dialogue.
5. Keep track of the dialogue history by clicking on the Dialogue History dropdown menu.
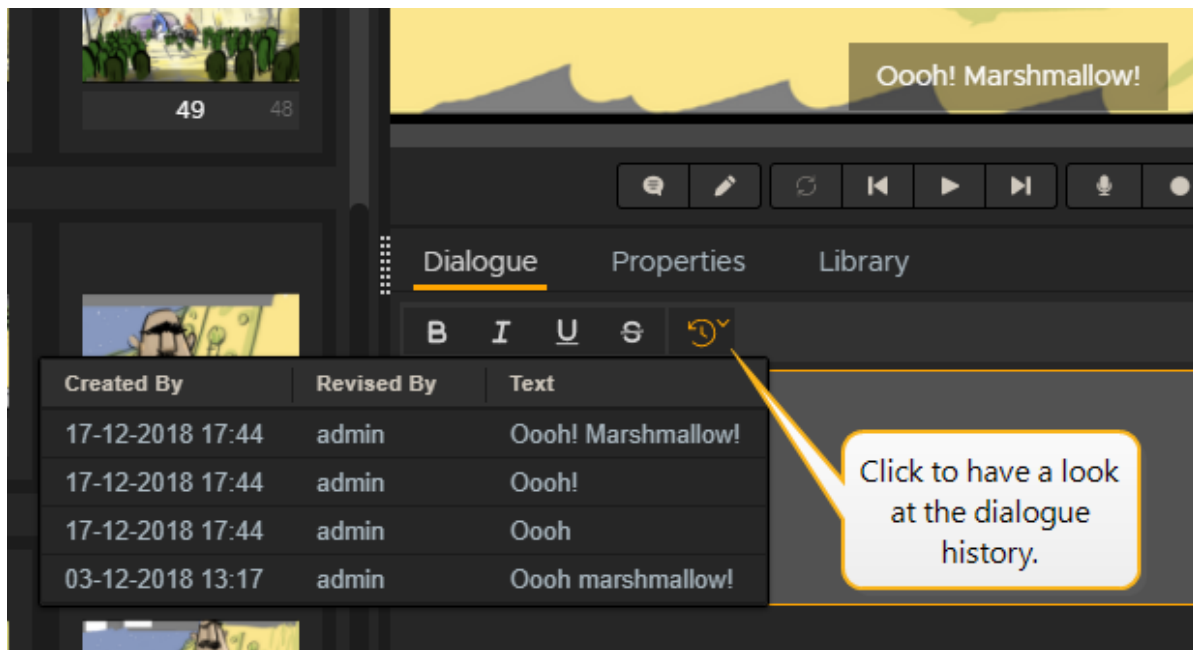
Method 2

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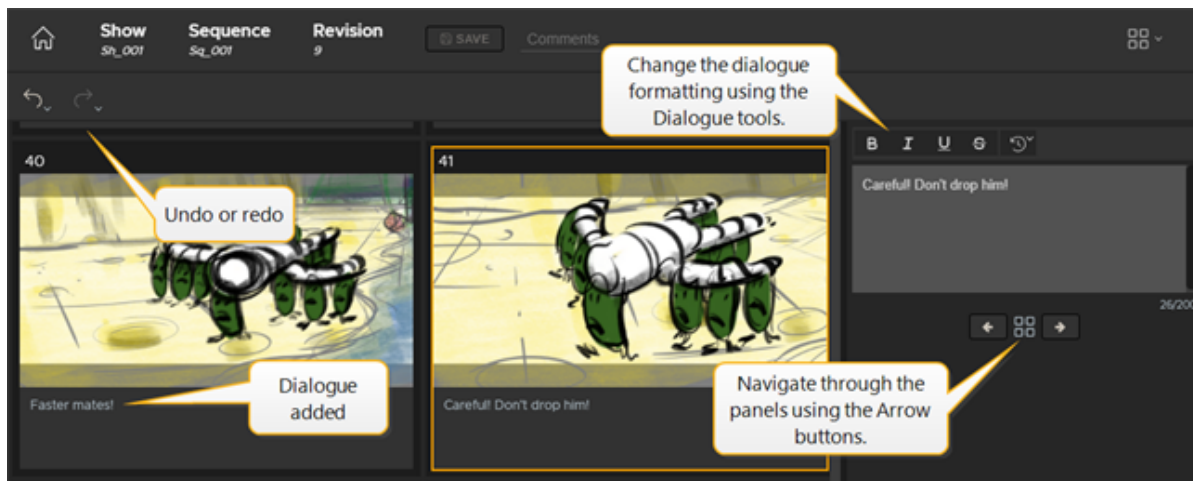
3.   **Save** your sequence revision to save all new dialogue.

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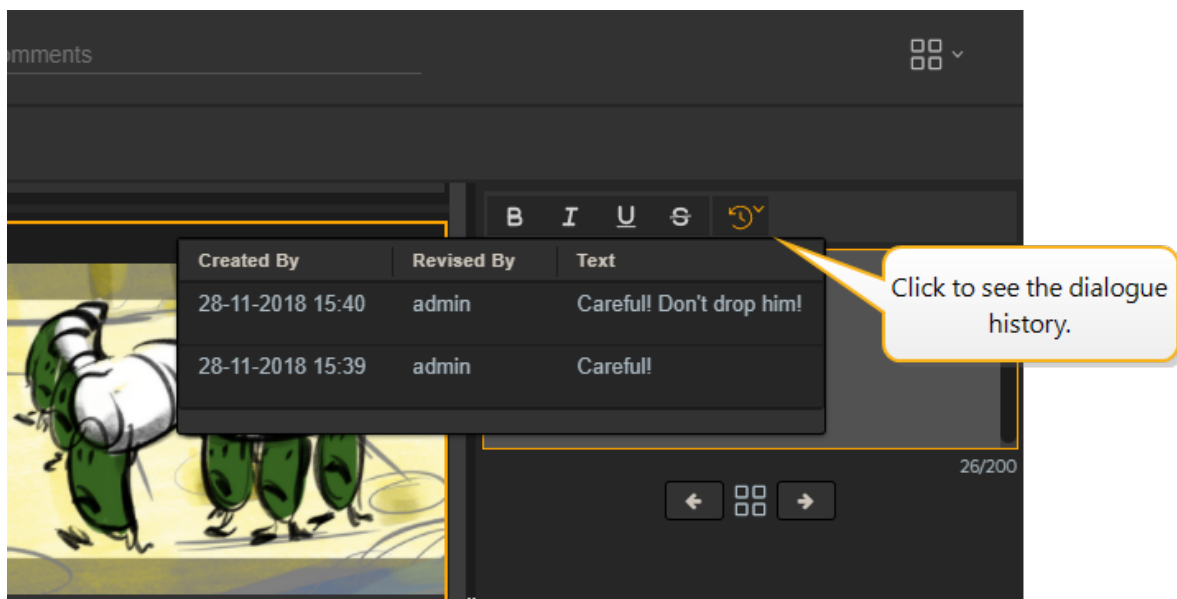
Annotations

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 Annotating a Panel Using Photoshop.

1. In the Panel Properties pane, toggle the annotations on by clicking the Pencil button at the bottom of the Player.
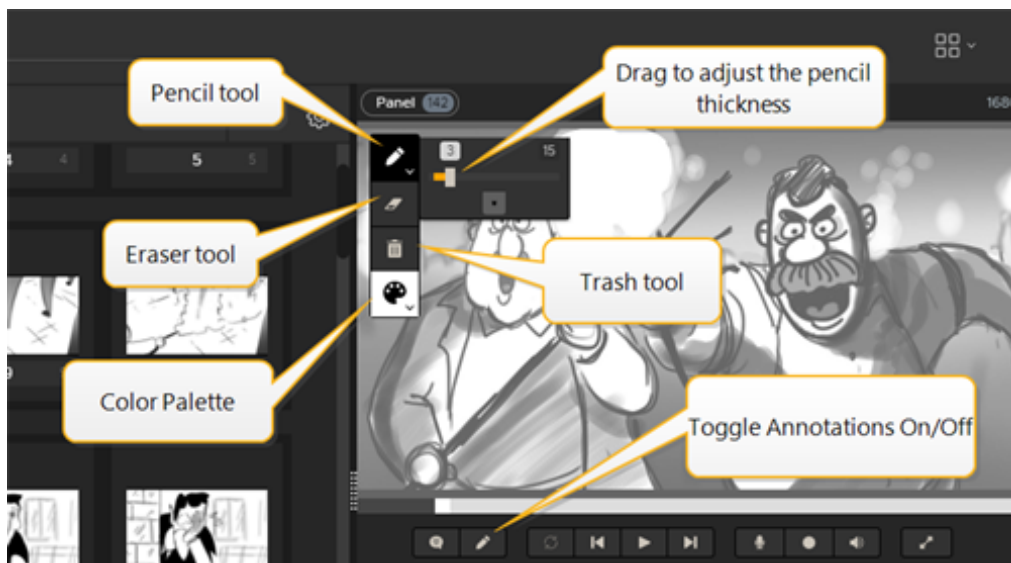
2. Click the Color Palette button then click again to select a color.
3. Click and hold the Pencil button then drag the slider to adjust the thickness of the pencil.
4. Make annotations on your panel.
5. Use the Eraser tool to partially erase your annotations or use the Trash tool to completely remove 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 ![filter icon] button and enable the **Annotations** filter.

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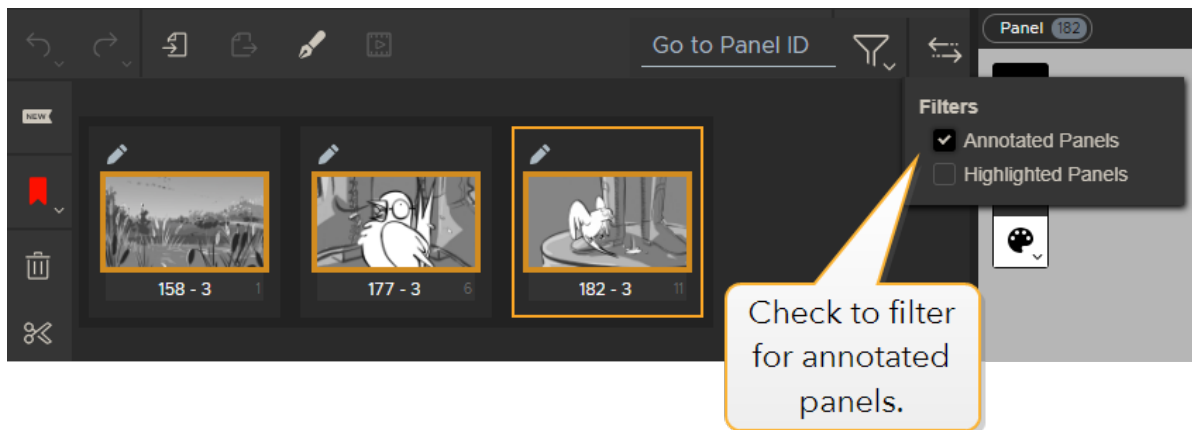
> **Tip:** By default, annotations appear as an extra layer in PSD files open in Photoshop. You can disable this by going to **File** > **Preferences** > **Third Party Apps** > **Adobe Photoshop** and disabling the **Send Annotation as Layer** preference.

Panel Comments

You can add comments to individual panels, which allows Flix 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 post it.

    The panel on which you've commented now displays a comment icon. Hover over the icon to see the latest open comment.
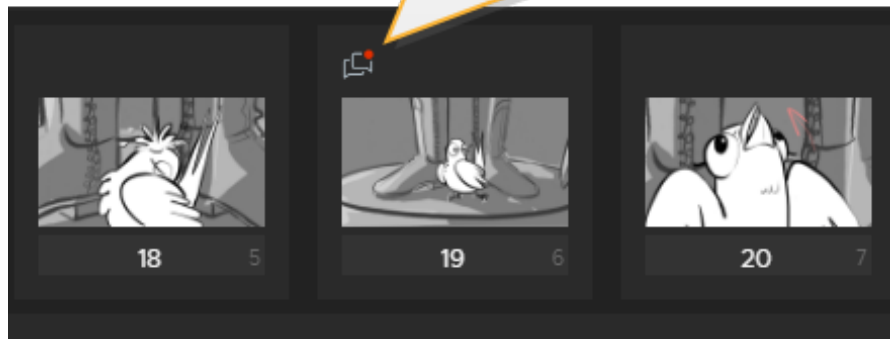
Your comment appears in a feed with any other comments other users have added to that panel.
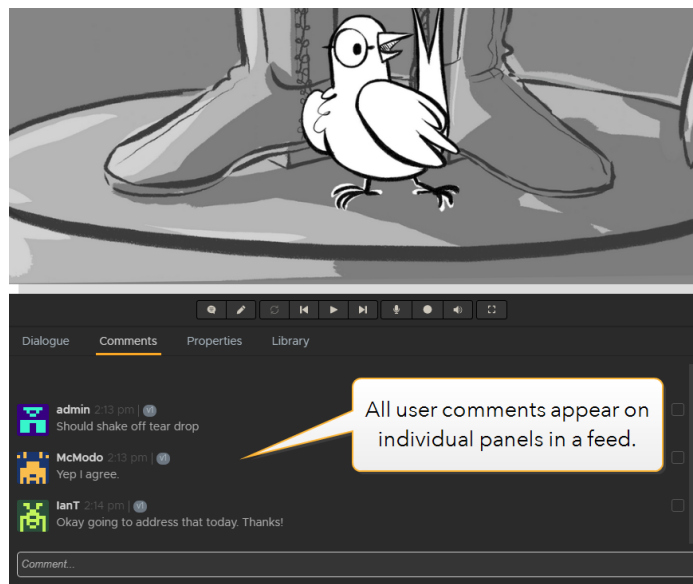
Every comment has a checkbox which allows users to mark as resolved. For example, once feedback in a comment has been addressed, the person approving the change would tick the comment so everyone in the production knows that feedback has been actioned.
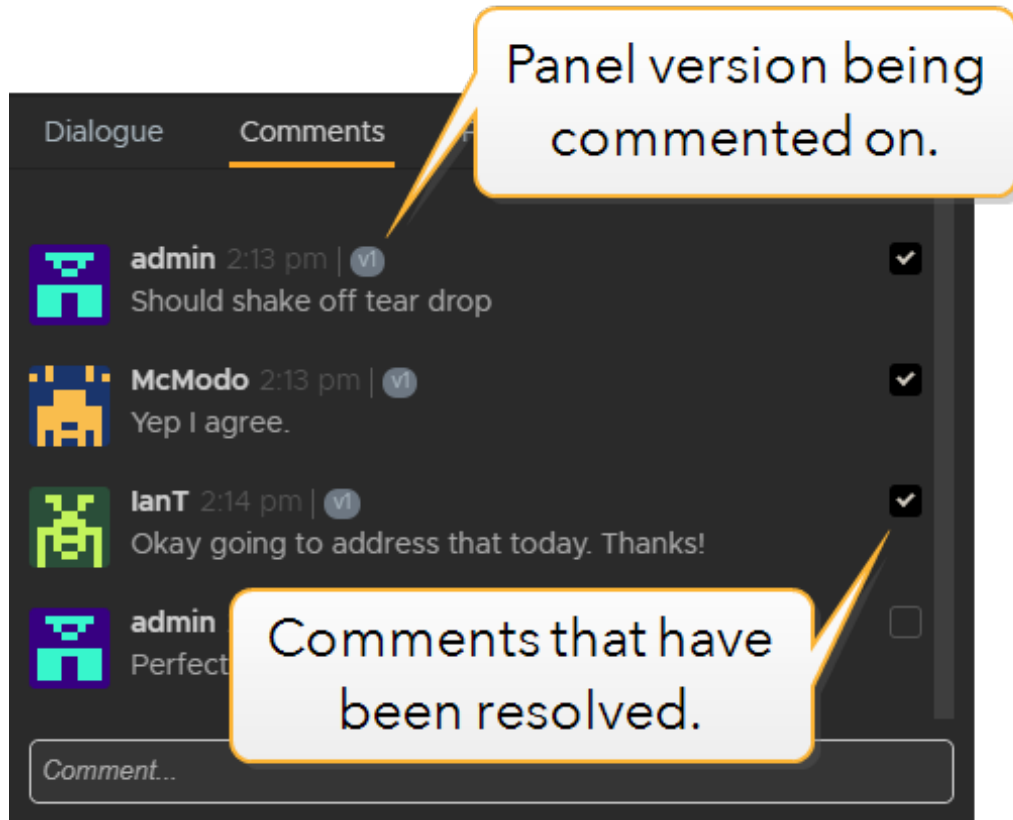
> **Note:** Once all the comments in a feed have been resolved, the comment icon disappears from the panel.

Filtering a Sequence by Comment

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

To filter by comment:

1. Click on the [filter icon] filter icon in the panel browser.

2. Select **Commented**.

   Flix displays only panels with open comments on them.
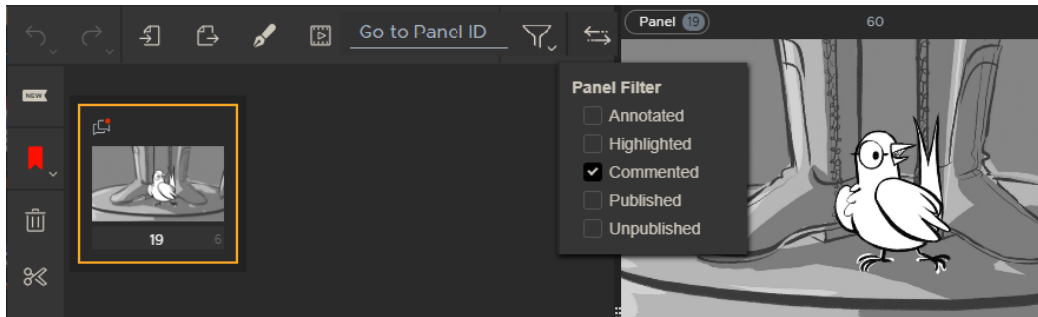
Recording a Pitch

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.
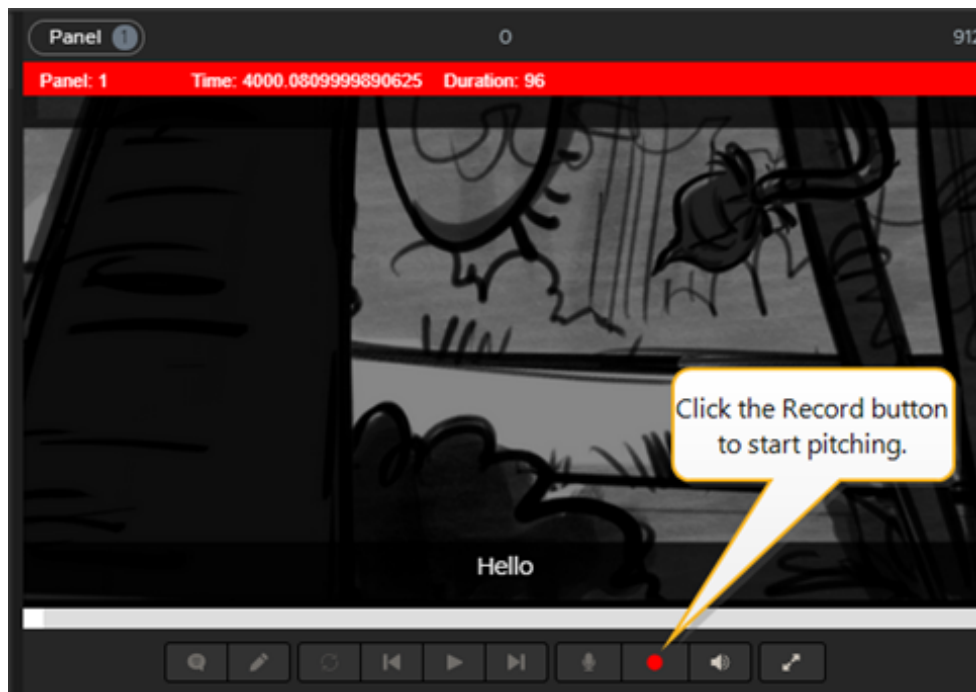
3. As soon as you reach the required duration, press the right arrow key to move to the next panel.

   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 & Photoshop

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 first need to select Photoshop as your sketching tool then set up the required version and install the plug-in.

1. Ensure Photoshop is closed while setting preferences in Flix.

2. Navigate to **File** or **Flix** > **Preferences** > **Third Party Apps** > **General**.

3. Under **Sketching**, select **Photoshop** from the dropdown menu.

4. In the **Adobe Photoshop** tab:

   • set the **Executable** preference by browsing to the version of Photoshop you want to use with Flix.

   • set the **Panel Open Behavior**:

       • **Open as Separate PSD** - Opens the selected panels in Photoshop as separate **.psd** files.
       • **Open in Layer Comps** - Opens the selected panels in Photoshop as layer comps.
       • **Open in Timeline** - Opens the selected panels in Photoshop in the Timeline.

   > 💡 **Tip:** If you use the **Open in Timeline** option, make sure in Photoshop the **New Layers visible in All Frames** option is off. It is available in the hamburger menu in the timeline.

   • enable or disable **Always Open Master Image** - This opens the **.psd** file in Flix as a master image, which means that all information is saved, such as hidden layers and empty groups.

   • enable or disable **Send Annotation as Layer** - This option sets annotated panels to open in Photoshop with their annotations shown on an additional layer.

   • click **Install Plugin**. This installs the Photoshop scripts to run the Photoshop actions.

   > ⚠️ **Warning:** If you're using Photoshop 2022, refer to Installing Plug-ins for Photoshop 2022.

   > 🗒️ **Note:** Installing the Photoshop plug-in may require admin privileges. Ask your system administrator for assistance, as they can install these manually if needed. Steps for this can be found in Manually Installing the Photoshop Plug-in for End Users.

5.   To open the Photoshop actions, in Photoshop, navigate to **Window** > **Extensions** > **Flix**.

The Photoshop Actions open in a new tab called **Flix**.

Installing Plug-ins for Photoshop 2022

Photoshop 2022 support requires some extra set up before you install the plug-in.

> **Article:** Have a look at Knowledge Base article Q100649 for more detailed information on installing Photoshop plug-ins.

macOS

1.   Launch Photoshop 2022 through the Creative Cloud application **before** you start the Flix Client.

2.   Go to **Photoshop** > **Preferences** > **Plugins** and enable:
     • **Allow Extensions to Connect to the Internet**, and
     • **Load Extension Panels**.
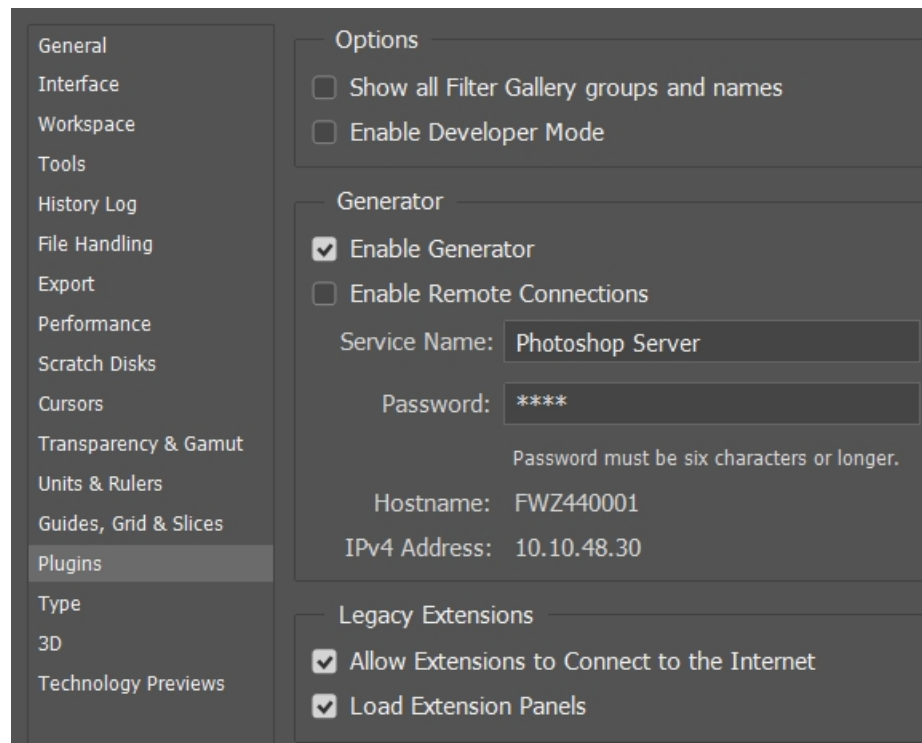
3. Then, in the Flix Client, navigate to **Flix** > **Preferences** > **Third Party Apps** > **Adobe Photoshop**.

4. In the **Adobe Photoshop** tab, click **Install Plugin**.

   The Photoshop scripts to run Photoshop actions are installed.

Windows

1. Launch Photoshop 2022 through the Creative Cloud application **before** you start the Flix Client.

2. Go to **Edit** > **Preferences** > **Plugins** and enable:
   • **Allow Extensions to Connect to the Internet**, and
   • **Load Extension Panels**.
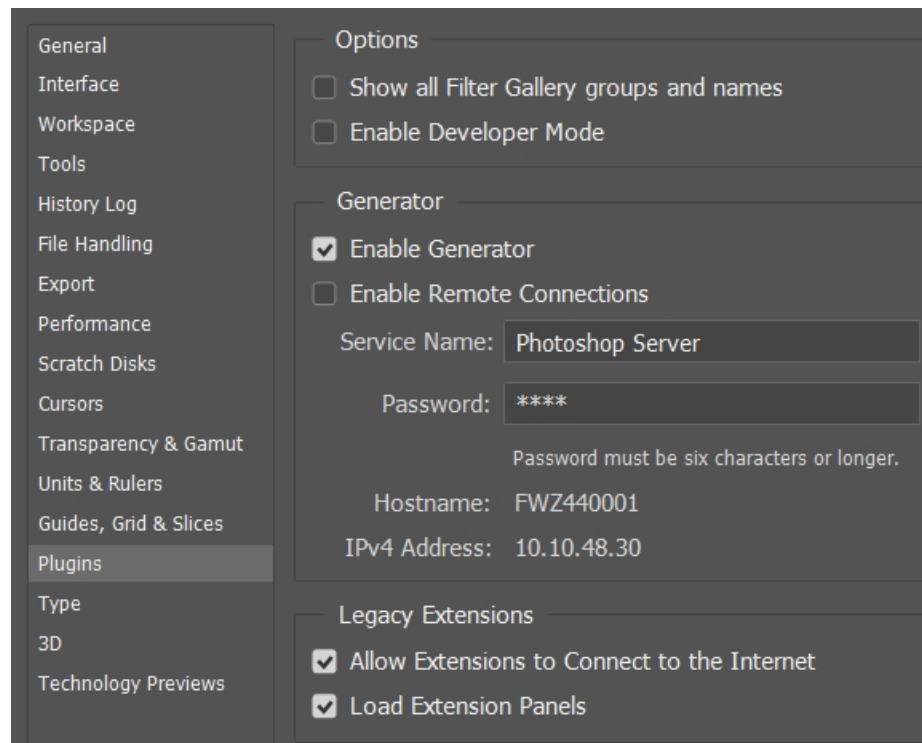
> ⚠️ **Warning:** You must run the Flix Client with Administrator permissions to install the plug-in.

3. Right-click the Flix desktop shortcut.

4. Click **Run as administrator**.
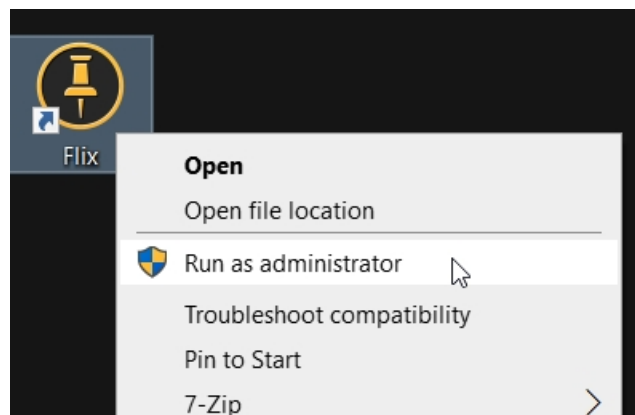
The Flix Client starts up as normal.

5. Navigate to **File** > **Preferences** > **Third Party Apps > Adobe Photoshop**.

6. In the **Photoshop** tab, click **Install Plugin**.

The Photoshop scripts to run the Photoshop actions are installed.

Assigning Keyboard Shortcuts to Flix Commands

From Flix 6.3.3 onwards you can assign keyboard shortcuts or 'hotkeys' to the list of Flix commands in Photoshop. The video below shows you how.

> 📝 **Note:** If you do not have administrator privileges, you will need to contact your IT department to help you with the following steps.

To set up Flix keyboard shortcuts in Photoshop:

1. In Photoshop, click the hamburger menu on the Flix toolset.
2. Select **Enable Flix Hotkeys**.

An alert with instructions appears and two windows open automatically.

3. Click **OK** to close the alert.
4. Copy the **Flix 6 Hotkeys** folder from the temp directory to the **Scripts** folder in Photoshop's directory.

> **Note:** We recommend copying rather than cutting or moving the **Flix 6 Hotkeys** folder from the temp directory. If you mistakenly delete or misplace the folder, restart the process from step 4, and the folder is recreated in the temp directory.

5. Restart Photoshop.

    You can now assign keyboard shortcuts to Flix commands in Photoshop. If you're already familiar with assigning keyboard shortcuts in Photoshop, you can skip the rest of these steps.

6. In Photoshop, navigate to **Edit** > **Keyboard Shortcuts...**

    The **Keyboard Shortcuts and Menus** window opens.

7. In the **Application Menu Command** column, expand the **File**section and scroll down to the **Scripts** subsection.
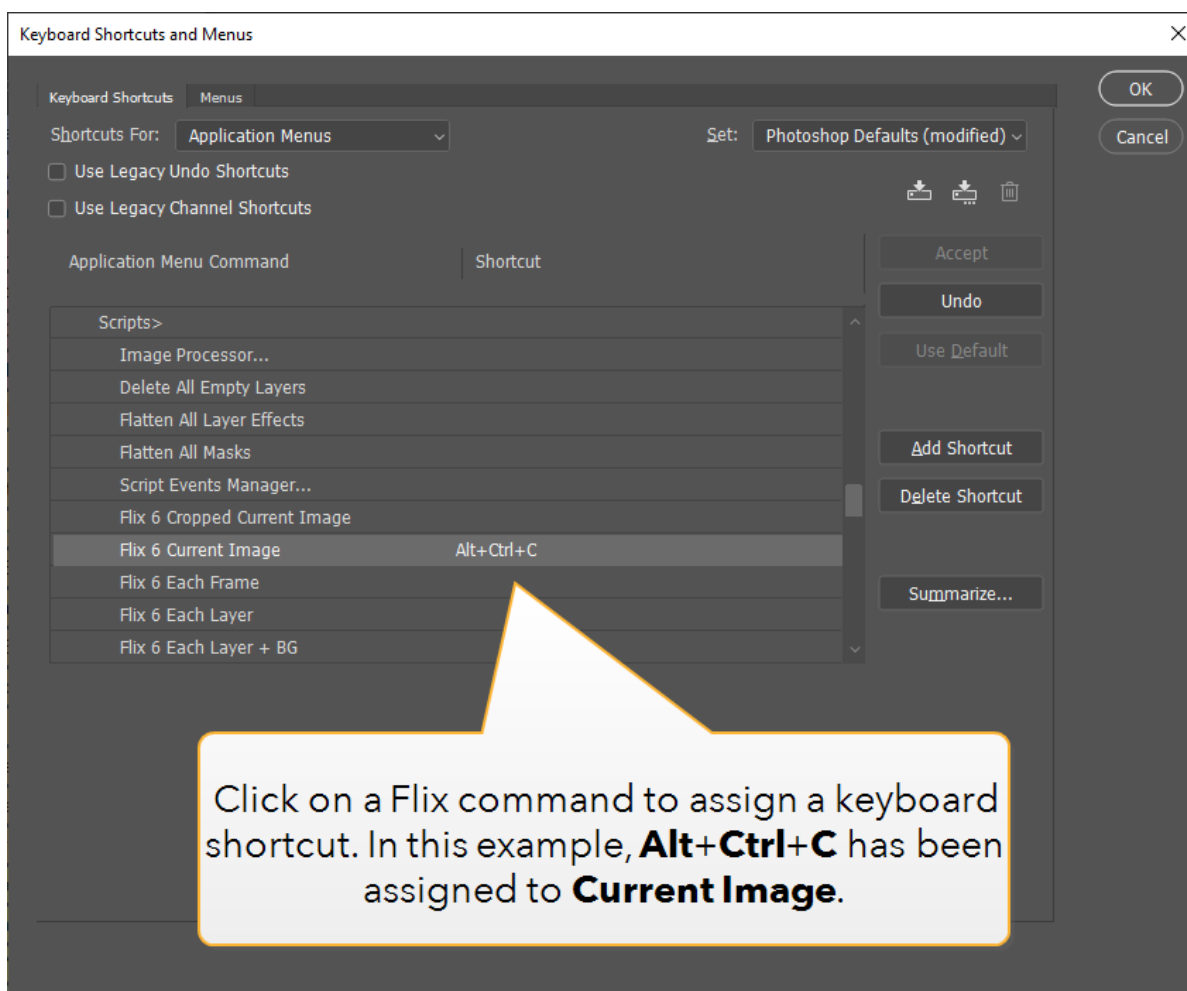
8. Click on a Flix command to add a shortcut. Click **Accept** once you've entered the shortcut commands you want to use.

> **Note:** If you assign a shortcut to a pre-existing one, Photoshop alerts you to a conflict. You can overwrite the original shortcut with your new one by clicking **Accept**, or click **Accept and Go to Conflict** to assign a new shortcut to the pre-existing one.

9. Click **OK** to close the **Keyboard Shortcuts and Menus** window.

> **Tip:** You can check the scripts that have been installed in Photoshop by going to **File** > **Scripts**. Any assigned hotkeys are also listed here.

Creating a New Panel and Sending it 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.

1. In Photoshop, in the **Flix** tab, select the aspect ratio at the top of the list to match the one of your show:
   - 1.0:1
   - 1.77:1
   - 1.85:1
   - 2.2:1
   - 2.35:1
   - 2.39:1

> **Note:** Select whether you want to **Always Send Original as Master**. This sends the original file, unedited, as a Master file to Flix. This does not insert anything in the edit (sequence revision).

2. Select **New Template**.
3. Draw on the panel.
4. Select **Current Image**.

Flix sends the panel to the edit (sequence revision). The panel is inserted after the currently selected panel

Creating Thumbnails and Sending them to Flix

1. In Photoshop, open the **Flix** tab and select the aspect ratio at the top of the list to match the one of your show.

OPTIONAL: Select whether you want to **Always Send Original as Master**. This sends the original file, unedited, as a Master file to Flix. This does not insert anything in the edit (sequence revision).

2. Click **New Thumbnails**.

3. Draw on the thumbnails.

4. Click **Send Thumbnails**.

   Flix sends the thumbnails to the edit. The thumbnails are inserted as nine new individual panels after the currently selected panel, if any.

> 📝 **Note:** Make sure the **File** > **Preferences** > **Always Open Master Image** preference is disabled, otherwise when you open one of the nine thumbnails from Flix to Photoshop it will open the original panel containing the nine 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 **Selected Layer as Annotation**.

4. In Flix, click the annotations button under the player to display the new annotation.

Working with Flix and Photoshop's Layer Comps

OPTIONAL: Select whether you want to **Always Send Original as Master**. This sends the original file, unedited, as a Master file to Flix. This does not insert anything in the edit (sequence revision).

• In Photoshop, with your **.psd** file open, open the **Flix** tab and click **Each Layer Comp** or **Selected Layer Comp**.

Photoshop sends to Flix each (selected) layer comp as a single panel.

Making Revisions to Layer Comps

If a master image is available for your layer comps:

1. Ensure the **File** > **Preferences** > **Third Party Apps > Photoshop > Always Open Master Image** is enabled.

2. Select the panel you want to work on.

3. In the main toolbar, click the Open in Sketching app button.

    In Photoshop, this opens the master image with all information including hidden layers and empty groups.

    In Flix, this opens the Master file that the panel is linked to in Photoshop.

4. 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 the first selected to the last, going from the first layer comp to the last.

    This video demonstrates how Master Images are created and where Flix stores them.

OR

If you haven't started your work with layer comps or don't have a master image:

1. Navigate to **File** > **Preferences** > **Third Party Apps > Photoshop > Panel Open Behavior** and select **Open in Layer Comps.**

2. In your edit (sequence revision), select the required panels you want to open as layer comps.

3. In the main toolbar, click the Open in Sketching app button to send your panels to Photoshop.

   This opens one **.psd** file containing a layer comp for each selected panel, each layer comp containing the layers associated to the selected panels.

4. After you are done working on your layer comps, save the **.psd** file

   This automatically replaces the opened panels with the updated layer comps.

Working with Flix and Photoshop's Frame Timeline

> 📝 **Note:** Select whether you want to **Always Send Original as Master**. This sends the original file, unedited, as a Master file to Flix. This does not insert anything in the edit (sequence revision).

• In Photoshop, with your **.psd** file open, open the **Flix** tab and click **Each Frame**.

  Photoshop sends each frame in the **Frame Timeline** to Flix as a single panel.

Making Revisions to Frames

If a master image is available for your frames:

1. Ensure the **File** > **Preferences** > **Third Party Apps > Photoshop > Always Open Master Image** is enabled.

2. Select the panel you want to work on.

3. In the main toolbar, click the Open in Sketching app button.

   In Photoshop, this opens the master image with all information including hidden layers and empty groups.

   In Flix, this opens the Master file the panel is linked to in Photoshop.

4. After you are done working on your master image, save the **.psd** file and select **Replace Each Frame** from the Photoshop actions.

   In Flix, this updates panels from the first selected to the last, going from the first frame to the last.

OR

If you haven't started your work with frames or don't have a master image:

> **Note:** When you are working in a Frame Timeline without any original master image, in the Frame Timeline, click the Hamburger button and disable **New Layers Visible in All Frames**. This displays all the frames in the Frame Timeline, otherwise all the panels are the same.

1. Navigate to **File** > **Preferences > Third Party Apps > Photoshop > 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 one **.psd** file containing one frame for each selected panel, each frame containing the layers associated to the selected panels.
4. After you are done working on your frames, save the **.psd** file

   This automatically replaces the opened panels with the updated frames.

Working with Flix and Photoshop's Layers/Groups

OPTIONAL: Select whether you want to **Always Send Original as Master**. This sends the original file, unedited, as a Master file to Flix. This does not insert anything in the edit (sequence revision).

In Photoshop, with your **.psd** file open, open the **Flix** tab and click any of the following:

> **Note:** Make sure the layers or groups you want to send are visible otherwise they will be ignored.

• **Each Layer** - Sends to Flix each visible layer in an image as a separate panel.

• **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.

• **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.

• **Selected Layer(s)** - Sends to Flix only the selected layers of the image as a new panel.

Making Revisions to Layers and Groups

If a master image is available for your layers:

1. Ensure the **File** > **Preferences** > **Third Party Apps > Photoshop >Always Open Master Image** is enabled.

2. Select the panel you want to work on.

3. In the main toolbar, click the Open in Sketching app button.

   In Photoshop, this opens the master image with all information including hidden layers and empty groups.

   In Flix, this opens the Master file the panel is linked to in Photoshop.

4. After you are done working on your master image, save the **.psd** file and select any layer actions (see above) from the Photoshop actions.

   In Flix, this updates panels from the first selected to the last, going from the first layer or group to the last.

OR

If you haven't started your work with layers or groups or don't have a master image:

1. Navigate to **File** > **Preferences** > **Third Party Apps > Photoshop > 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 one **.psd** file containing a layer or group for each selected panel, each layer or group containing the layers associated to the selected panels.

4. After you are done working on your layer comps, save the **.psd** file

   This automatically replaces the opened panels with the updated layers or groups.

Flix & Storyboard Pro

Setting Up Flix to Work with Storyboard Pro

To set Storyboard Pro as your Sketching Tool:

1. Navigate to **File** > **Preferences > Third Party Apps > General**.
2. Under **Sketching**, select **Storyboard Pro** from the dropdown menu.
3. In the **Storyboard Pro** tab:
   - 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\Toon Boom Animation\Toon Boom Storyboard Pro
     <version>\StoryboardPro <version>.exe

     /Applications/Toon Boom Storyboard Pro <version>/Storyboard Pro
     <version>.app
     ```

   - Under the **Import** section, 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.
   - Under the **Export** section, choose the behavior for exporting Flix sequences to Storyboard Pro.
     - **Export Path** - Set the path for your exported Flix sequence.
     - **Export Camera Moves** - Toggle ON to update Camera Moves in Storyboard Pro.
     - **Export Markers** - Marker updates are not currently supported in Storyboard Pro.
4. In the **Panel Browser** tab, 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 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.

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: Check that you have QuickTime installed on your machine or imports from Storyboard Pro will fail.

1. Save your Storyboard Pro project.
2. In Flix, navigate to the Revisions page, either by clicking on your sequence or creating a clean version.
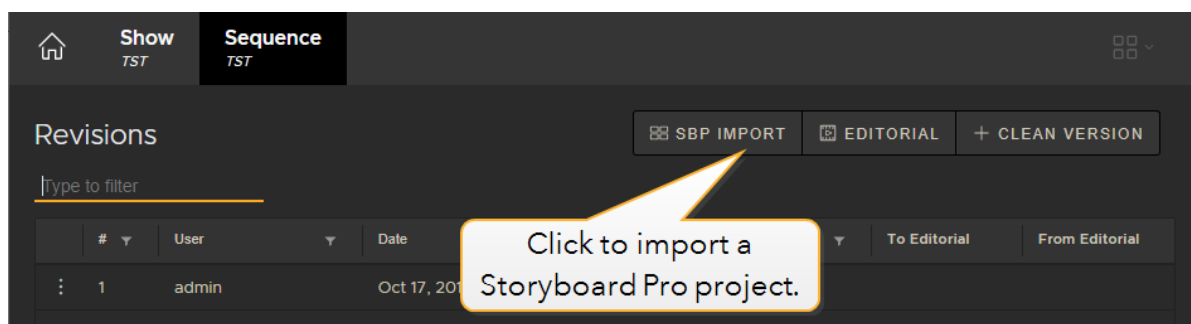
3. Click on the **SBP Import** button.

   The Storyboard Pro Import page opens.

> **Note:** If Storyboard Pro is not installed on the machine running Flix client or has not been set in your Preferences, the **SBP Import** button is disabled.

4. 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:
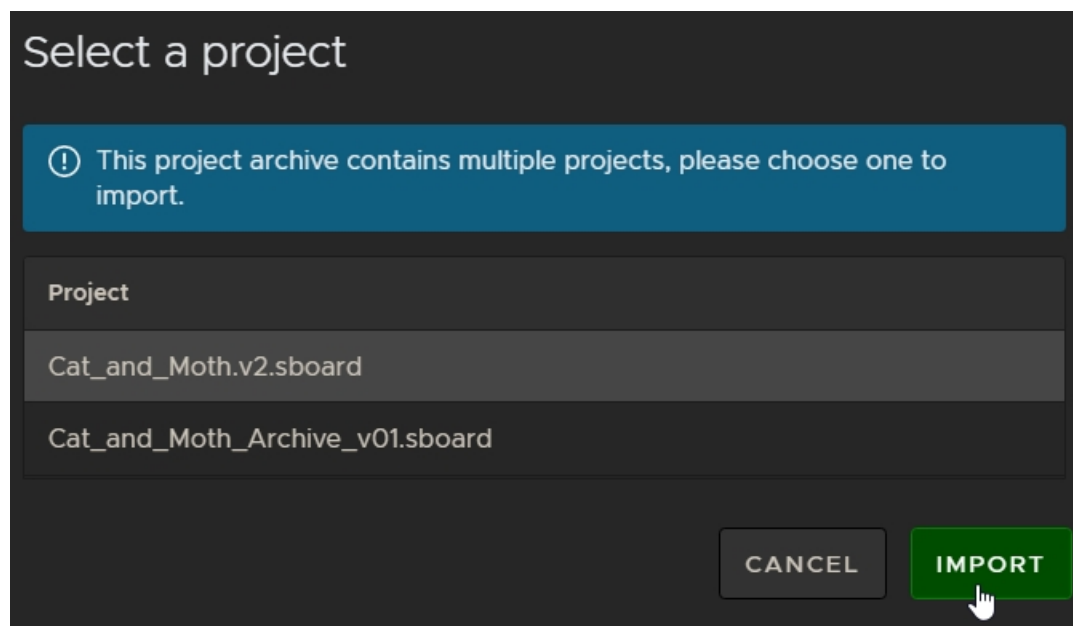
   The path to the project file appears in the **Project Path** when you've chosen the required file.

5. 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.

> 📄 **Note:** The **Store Source Files** option is grayed-out if you browse to a regular **.sboard** file.

> 💡 **Tip:** The **Store Source Files** option is enabled by default in the **Preferences** under **Third-Party Apps** > **Storyboard Pro** > **Import** > **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.
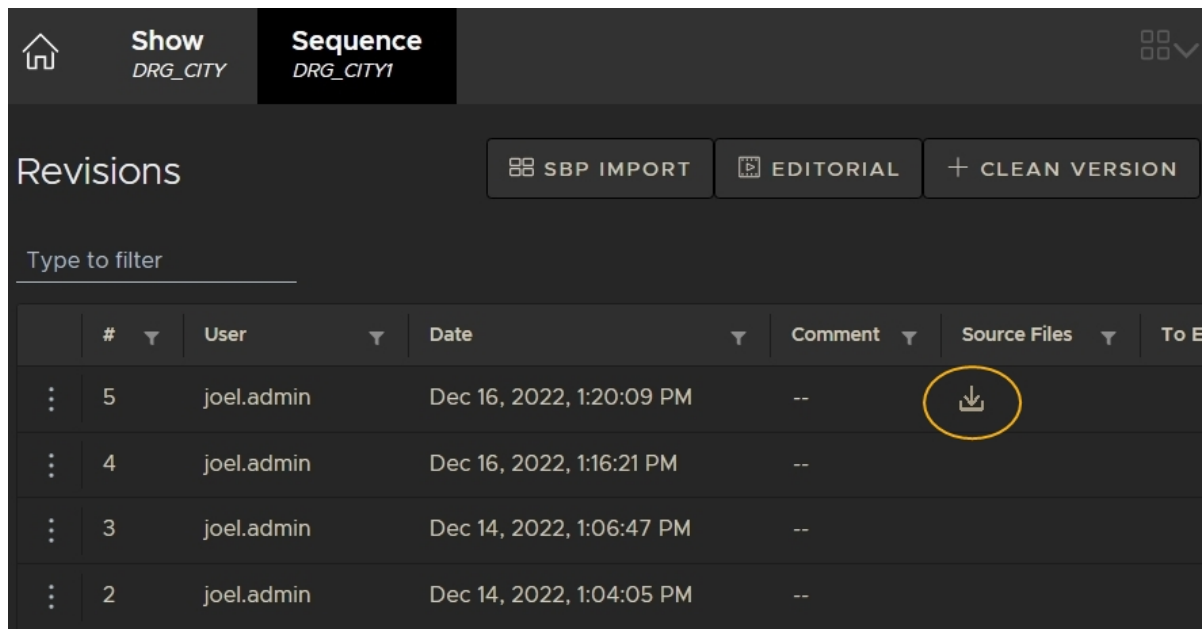
6. Enter a comment if you want it to appear on the new sequence revision, then click **Import**.

   Flix reports that the import was successful.

   > **Note:** Depending on the size of the project file, the import might take time.

7. Click on **Sequence** in the breadcrumb to return to the Revisions page.

   A new sequence revision has been created from your Storyboard Pro import. If markers are set to display in the panel browser, scene numbers show as markers on sequences imported from Storyboard Pro.

   > **Note:** In Flix 6.4, transitions between two panels in Storyboard Pro caused the duration of the panels either side of the transition to change when imported back into Flix. Flix incorrectly updated the marker position of the first panel to the start of the transition between panels.
   > In Flix 6.5, 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.
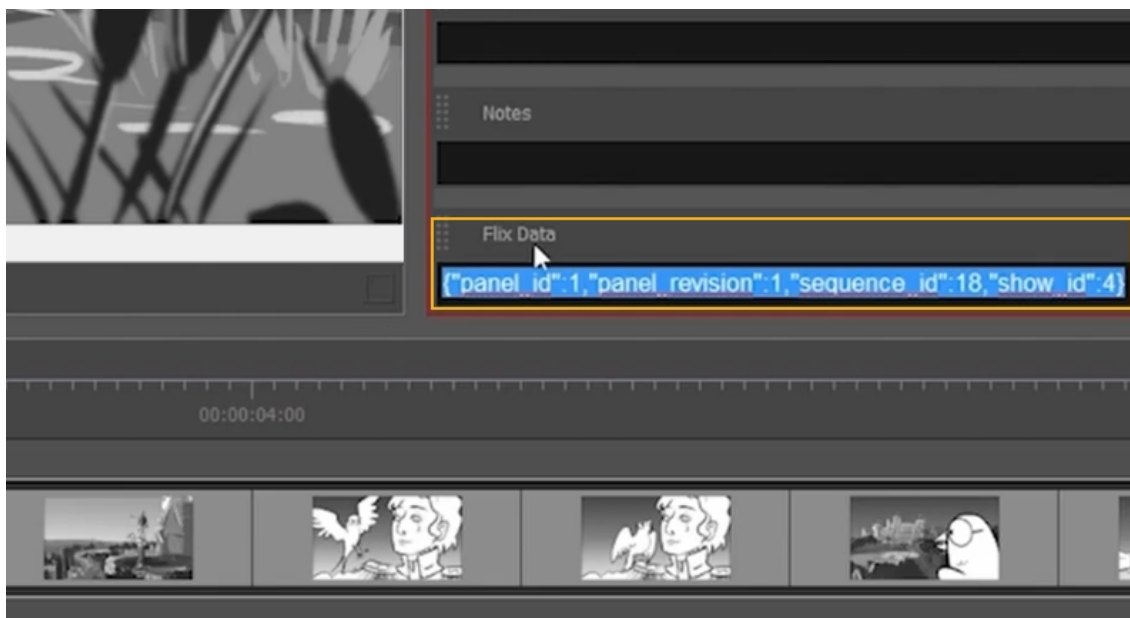
The same is true of panels in Flix. Each panel has an **Origin** in the panel **Properties**.

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 to Work with 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.

   Flix confirms if you want to send the selected sequence to Storyboard Pro.

4. Click **Export**.

Flix User Guide
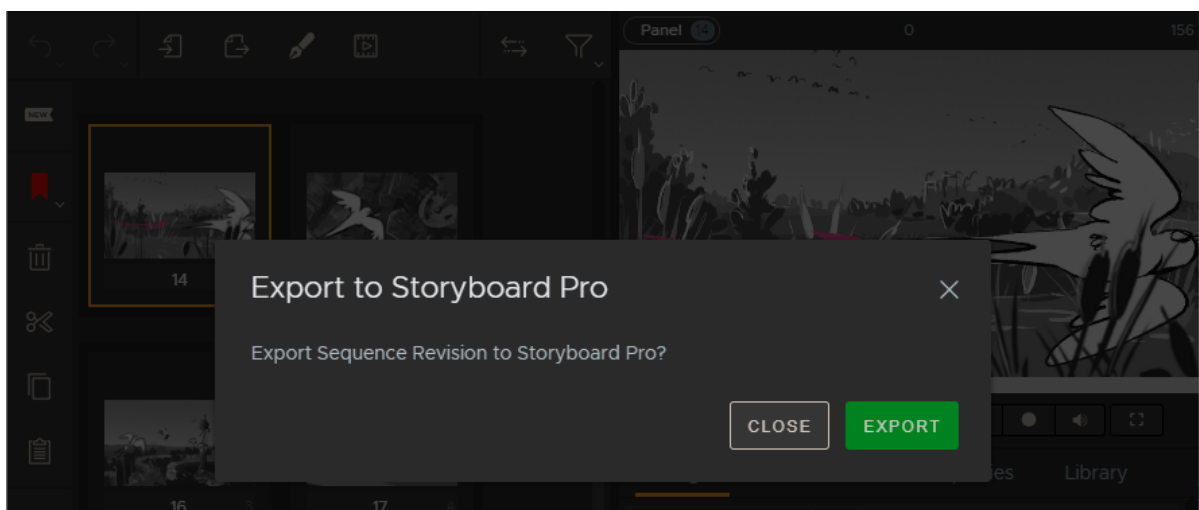
5. Flix creates a **.xml** file and saves it in the location nominated in your exports path.

   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.
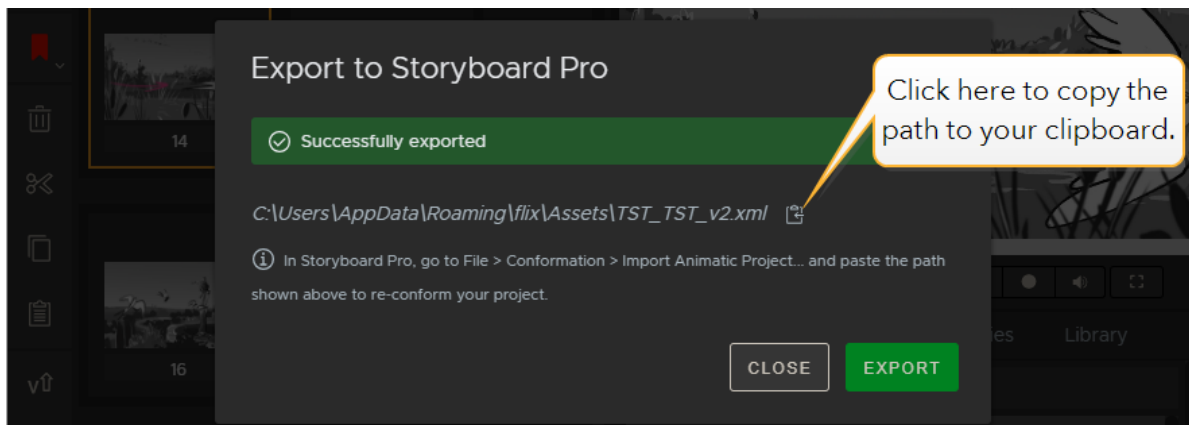
> **Note:** To change the default path for exports, go to **File** > **Preferences** > **Third Party Apps** > **Storyboard Pro** > **Export**.

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.
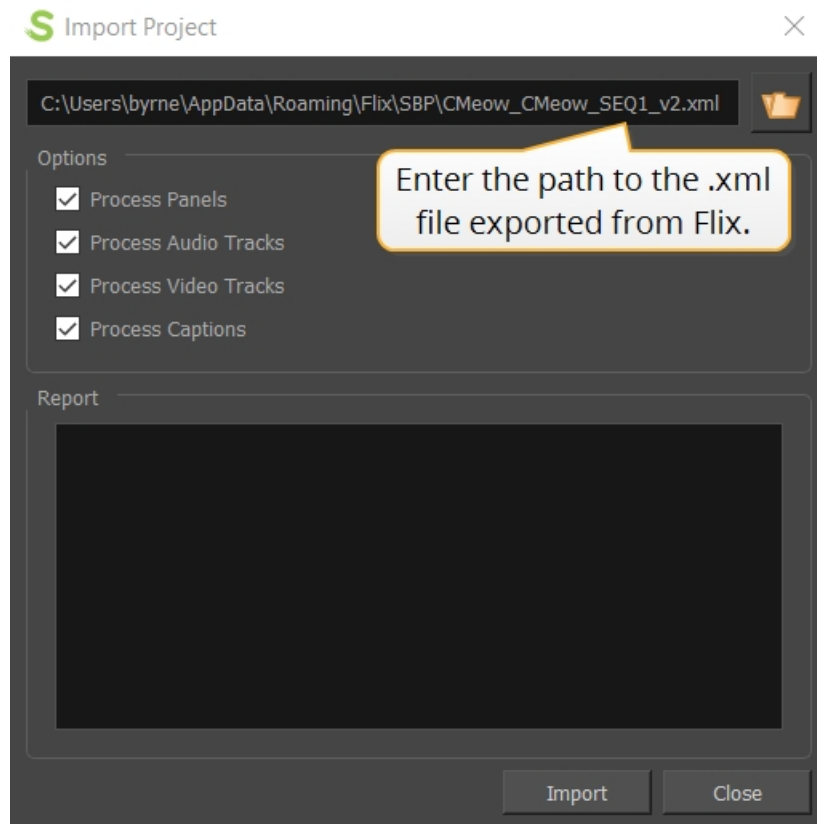
3. Enter the path to the **.xml** file exported from Flix, or click the 📁 icon and browse for the file.

> 💡 **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.
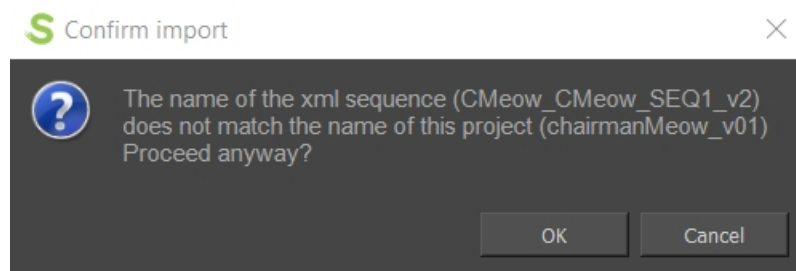
5. Click **OK** to proceed.

   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**.
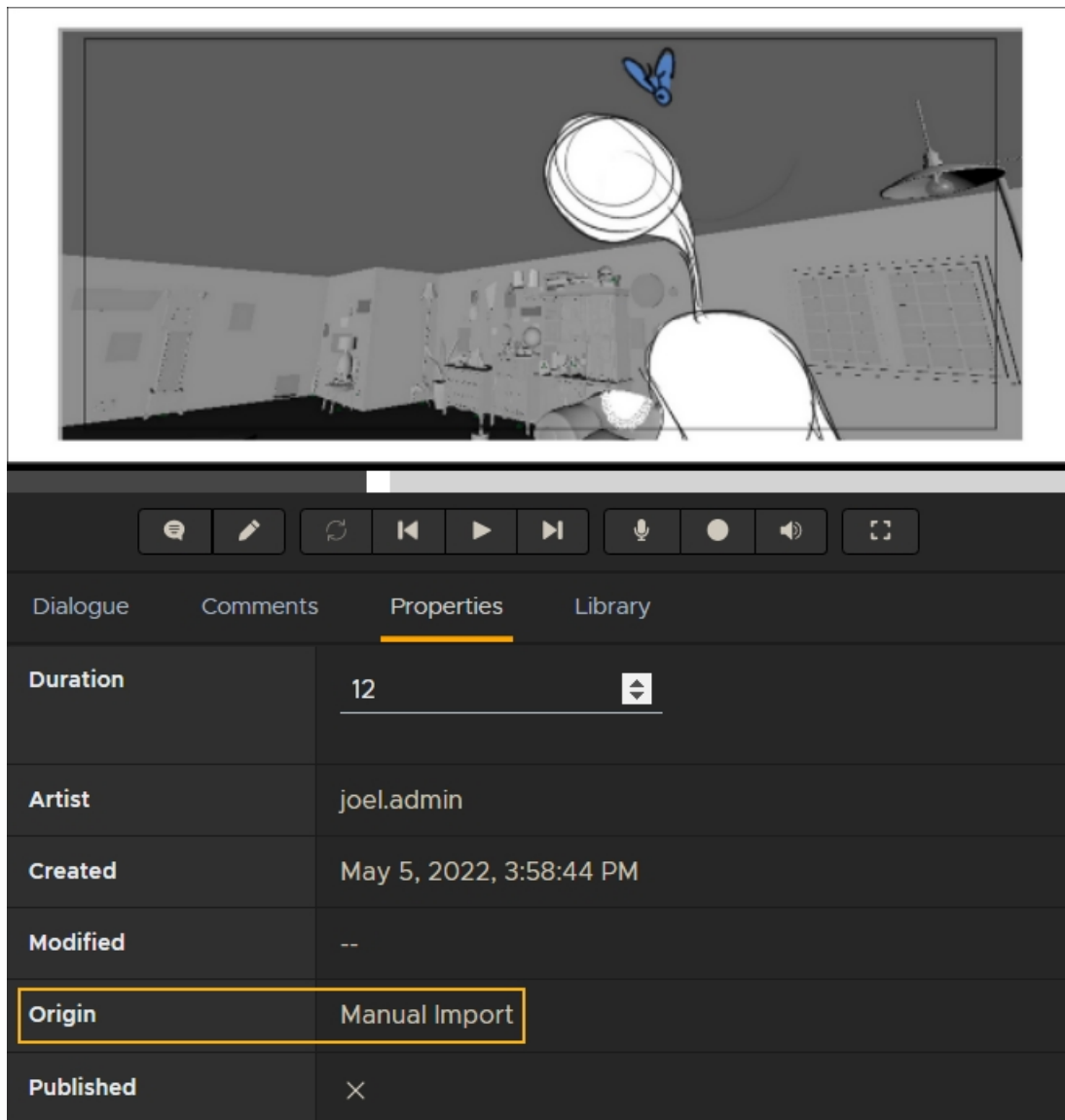
6.

> 📝 **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.

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. Make sure Premiere is your default Editorial Tool. For more information, see Third-Party Apps in the Preferences page.
2. In the **General** tab, go to the Editorial Tool dropdown menu and select Adobe Premiere.
3. Enter the **Publish Directory** path or click **Browse** and then select a folder. For example, T:\flix_publishes [show_tracking_code][sequence_tracking_code].

> 📝 **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.

4. In the **Adobe Premiere** tab, set your preferences for exporting **.xml** files to Premiere. See the table below for details.

Adobe Premiere	
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 add **Timeline** markers in Premiere, they are ignored by Flix.
Highlight New Panels	Sets whether new panels appear as highlighted clips in your Premiere sequence.

Setting up Flix with Avid Media Composer

1. Make sure Avid Media Composer is your default Editorial Tool. For more information, see Third-Party Apps in the Preferences page.

2. In the **Third-Party Apps** > **General** tab, go to the **Editorial Tool** dropdown menu and select **Avid Media Composer**.

3. Enter the **Editorial Clip Name** and **Editorial Sequence Name**, use the chips at the bottom of the screen to build your own, or use the defaults provided.

4. Enter the **Publish Directory** path or click **Browse** and then select a folder. For example, T:\flix_publishes [show_tracking_code][sequence_tracking_code].

> **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.

5. In the **Avid Media Composer** tab, set your preferences for exporting **.aaf** files to Media Composer. See the table below for details.

Avid Media Composer	
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.
Clip Duration	Specifies the default duration for clips sent to Media Composer in frames. (For example, 480 frames or 20 seconds for a 24fps show).
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.

	⚠️ **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.
Enable Mark In	Toggle ON to include a Mark In point on each clip.
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).
Enable Mark Out	Toggle ON to include a Mark Out point on each clip.
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.
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. • **PNG and 3D Warp** - this option preserves the camera move keyframes from Stroyboard Pro so that you can edit them in Avid and send them back to Flix without having to reproduce the effect.

	• **MOV** - the camera moves from Storyboard Pro are burned into the 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 Avid. If you disable this control, you can only send **.mov** files to Avid and the **Panels with Camera Move** control is disabled.
New Clip Color	Sets the color of new clips in your sequence.
Color Range	Sets either **Full** or **Legal** range color for publishes to Avid. 📝 **Note:** The default color range is set to **Full**.
Enable Audio	Controls whether audio is available in Avid Media Composer or not.

Preserve Sequence Timings from Flix

Flix integration with Storyboard Pro allows you to import camera moves and panel timings and pass them on to Avid Media Composer, 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.

- Enable **Preferences** > **Third-Party** > **Avid Media Composer** > **Use Flix Timing** to send every panel as a clip retaining its duration set within Flix.

- Disable **Use Flix Timing** to send every panel with the default duration.

> ⚠ **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.

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.
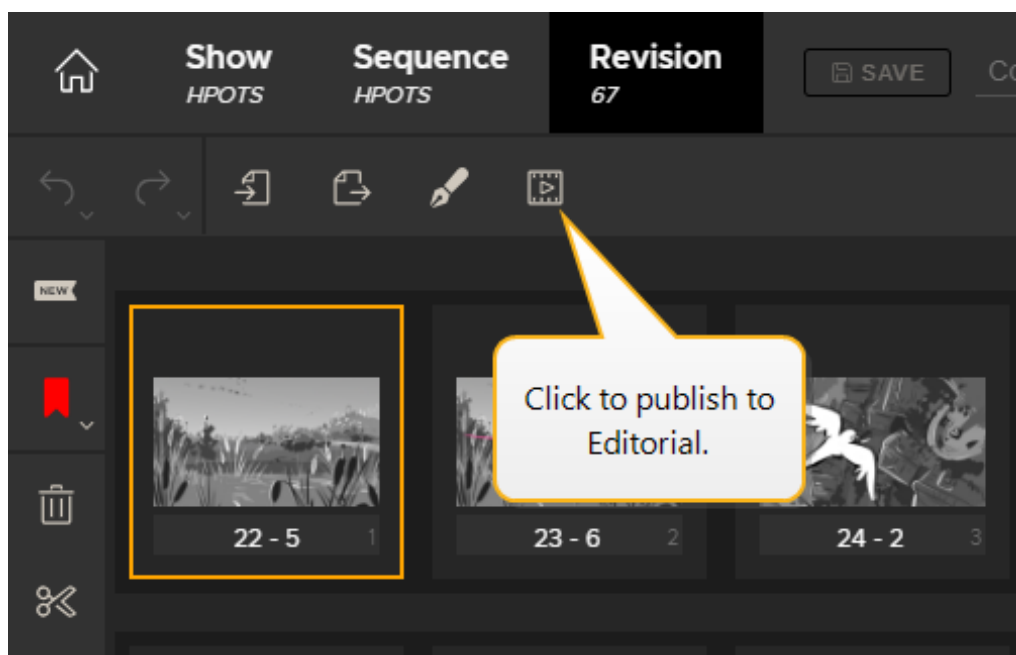
The **Publish to Editorial** dialog appears.

> **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.

2. Specify if you want to generate a Contact Sheet of the panels in your Revision by checking on **Generate Contact Sheet**. This is saved as a **.pdf** file.

3. 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.

4. Specify if you want to generate a text file of the dialogue in your Revision by checking on **Generate Dialogue File**. This is saved as a **.txt** file.

5. Enter a **Comment** if needed. This appears in the email notification.

6. 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 [icon] 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.
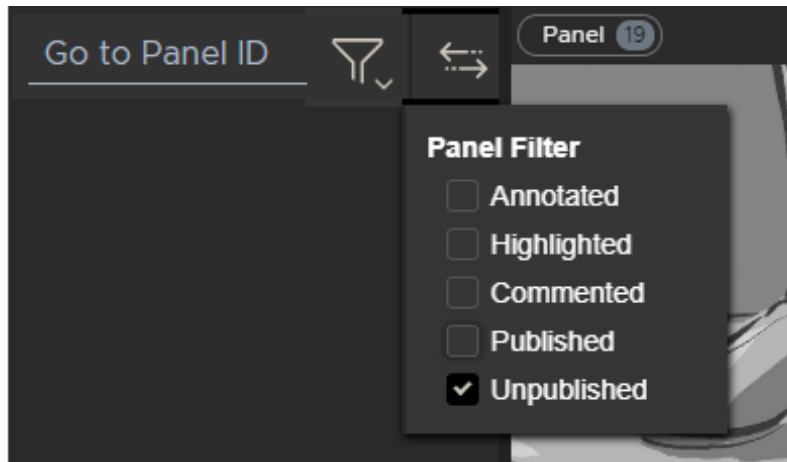
Importing Your Sequence to Adobe Premiere

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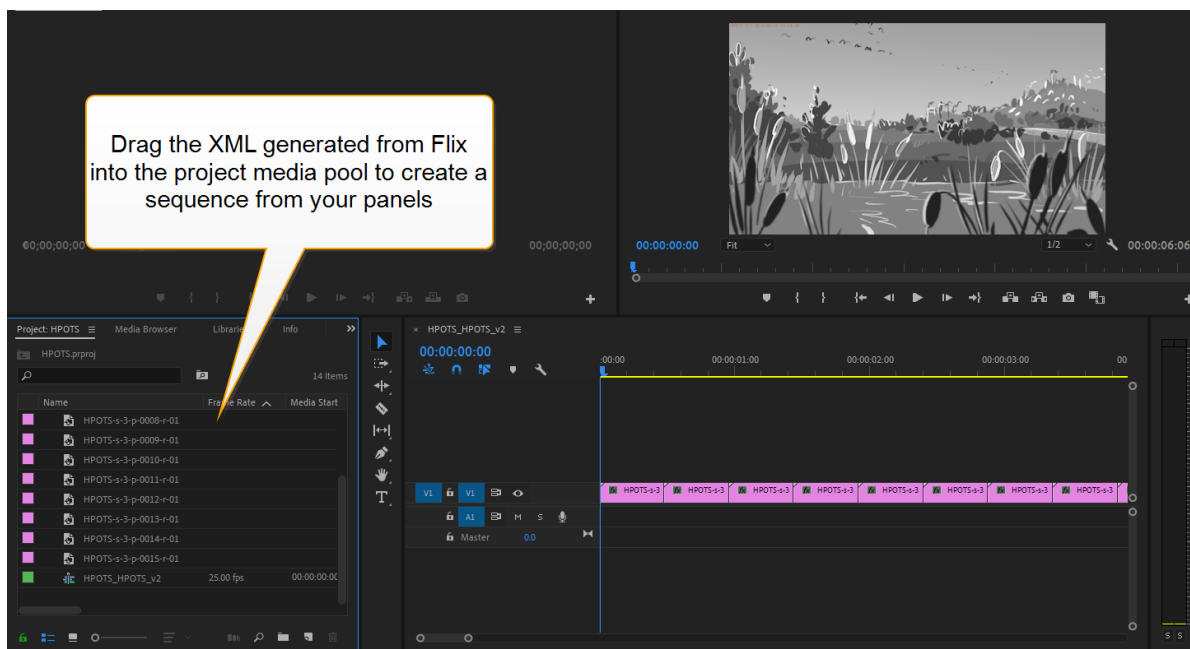
> 📝 **Note:** Recorded pitches embed shot duration metadata in the **.xml** file, so panels appear in the sequence with their recorded timings.

> 📝 **Note:** Animated panels originating from Storyboard Pro are sent to Premiere with the keyframe information intact, so they can be adjusted further in Premiere's effect editor.

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 After Effects 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.

> **Note:** Do not mix-down the composition panels in editorial or Flix can 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.
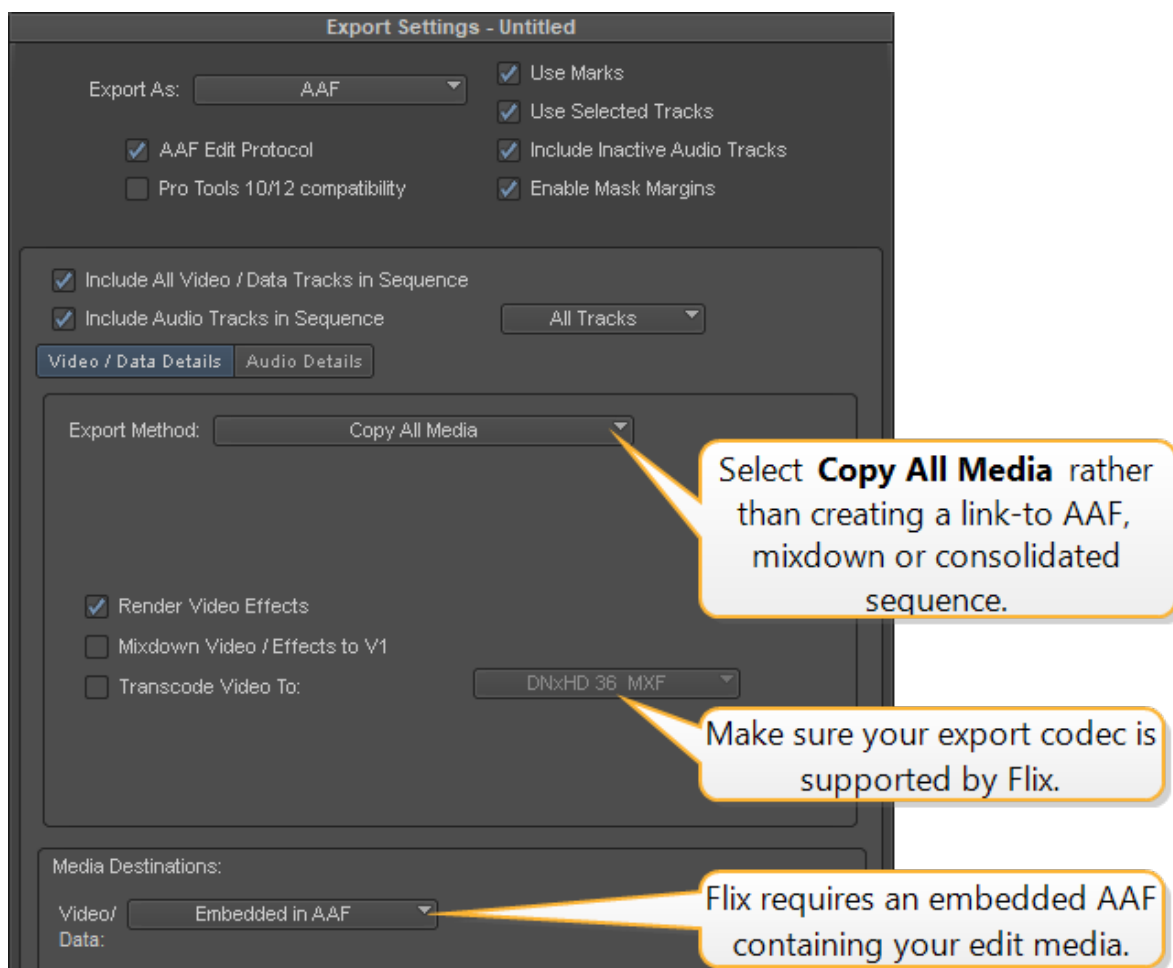
The **Audio Details** tab controls how audio is exported from Avid. Audio can severely impact the file size of AAFs exported from Avid to Flix.

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.
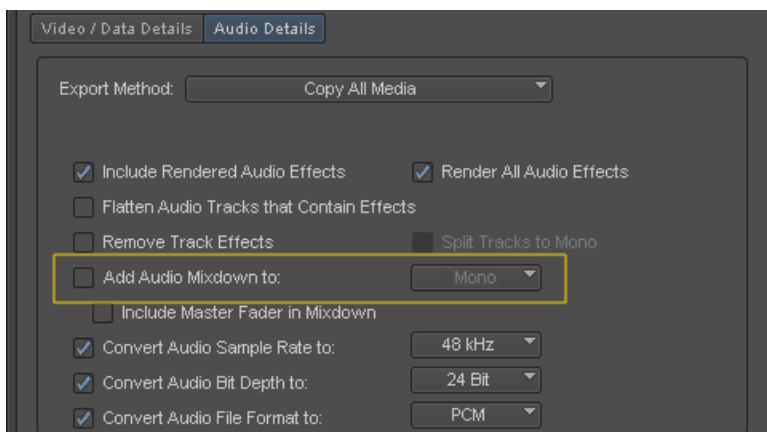
OR

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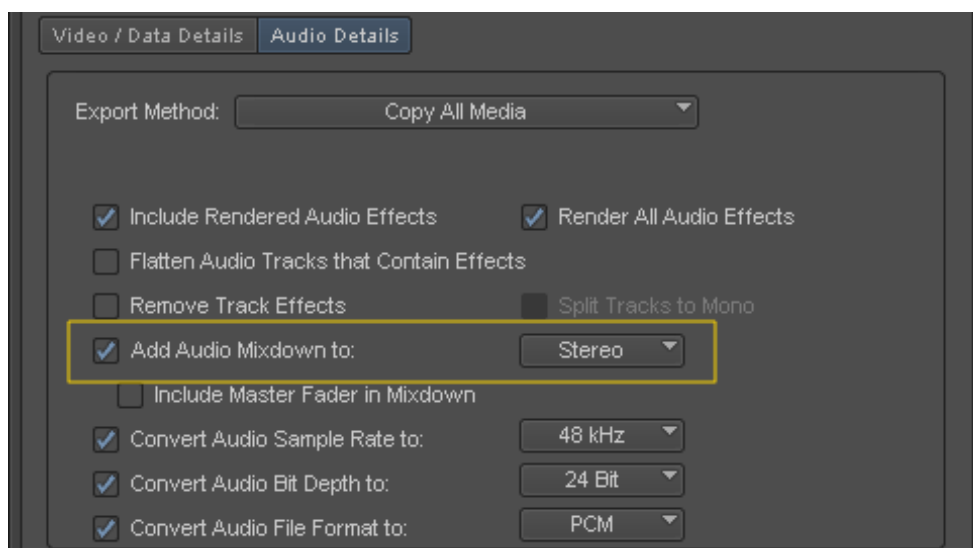
3. Click **Save** to finalize the video and audio export settings.
4. Choose a location to save your file and click **Save**.

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

Updating your Sequence in Flix

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

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

2. Click the **Editorial** button.
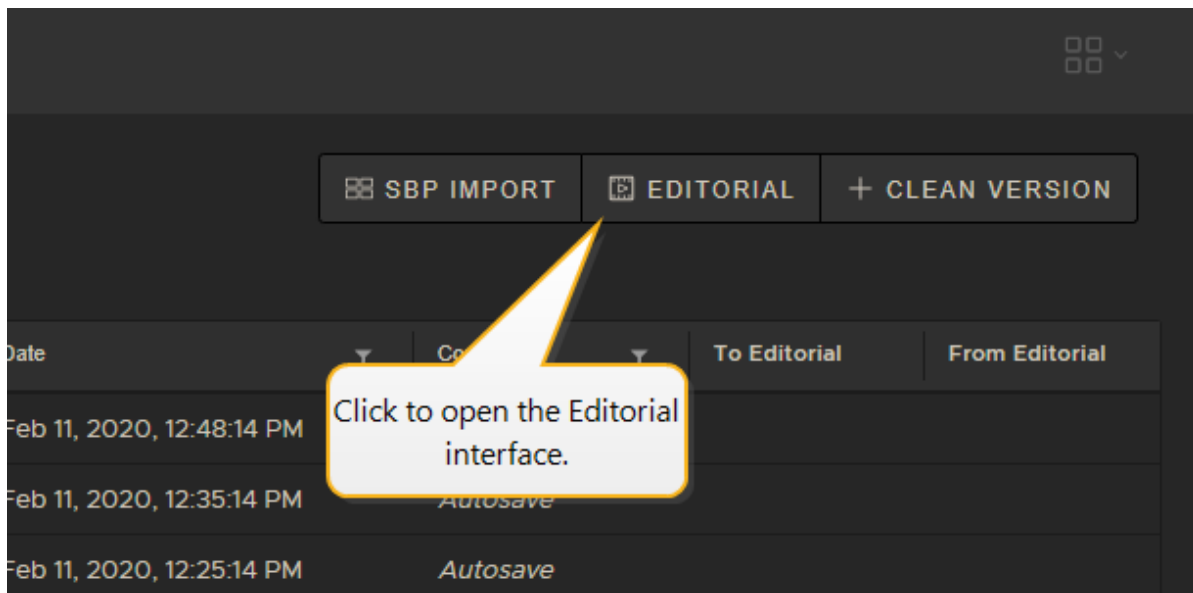
    This opens the Editorial Interface.
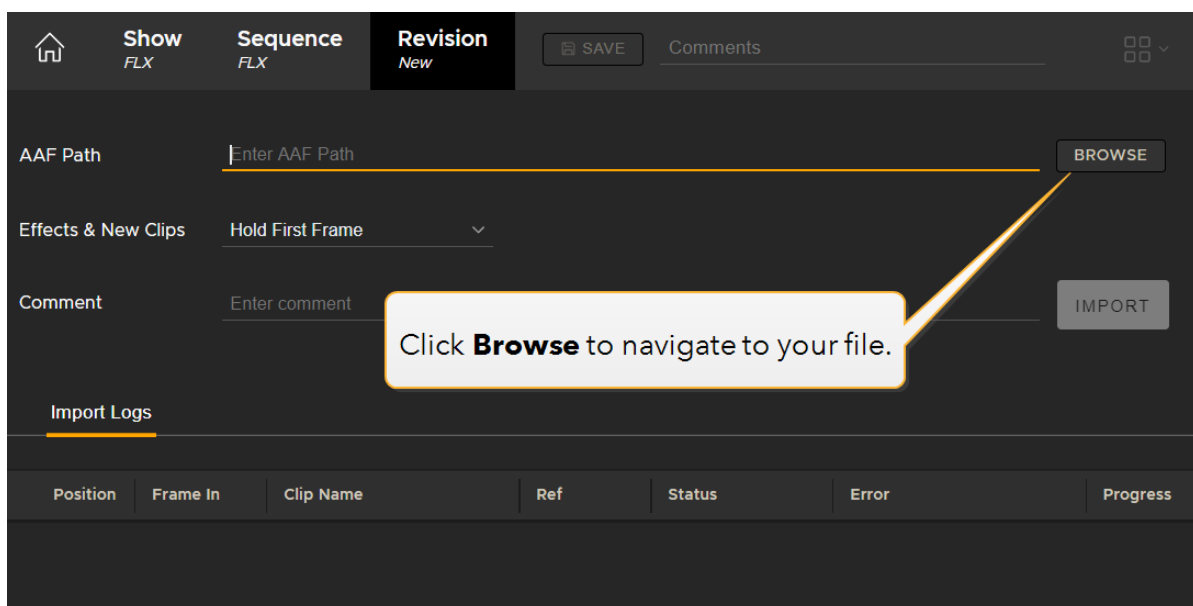
3. If you're using Adobe Premiere:

- **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. Select the **Effects** from the dropdown menu:

- **Ignore Effects** - (Fastest) - Flix reconforms the entire sequence without any effects.

- **Hold First Frame** - (Faster) - Flix renders only the first frame per panel and holds it for the duration of the clip. This renders faster, however the actual full effect in Flix cannot be seen, for instance, a camera move. If importing from Premiere, you can watch the reference QuickTime movie from Editorial to see the full effect.

- **Render All Frames** - Flix renders every single frame. This is slower to render but more accurate since the full animation can be seen.

5. Enter a **Comment** if required and click **Import**. The comment appears on the sequence revision in the Revisions list.

   The import details appear in the **Import Logs** so you can check what's changed on a panel by panel basis.

6. Go to the **Revisions** level to open the newly-imported sequence revision.
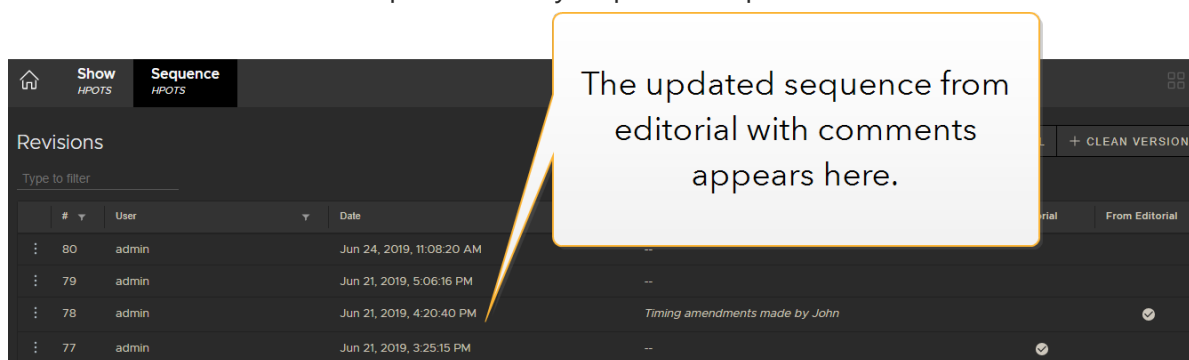
View Related Panels from Editorial in Flix

If you've created a composition from multiple panels 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.

A composition created from three panels.
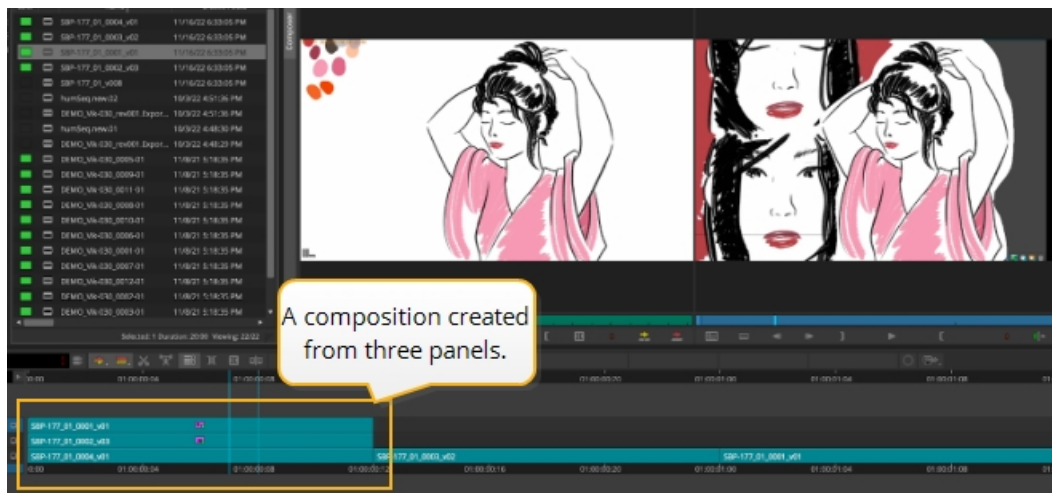
> ⚠️ **Warning:** Do not mix-down the composition panels in Editorial or Flix can not use its versioning system to create panel relations

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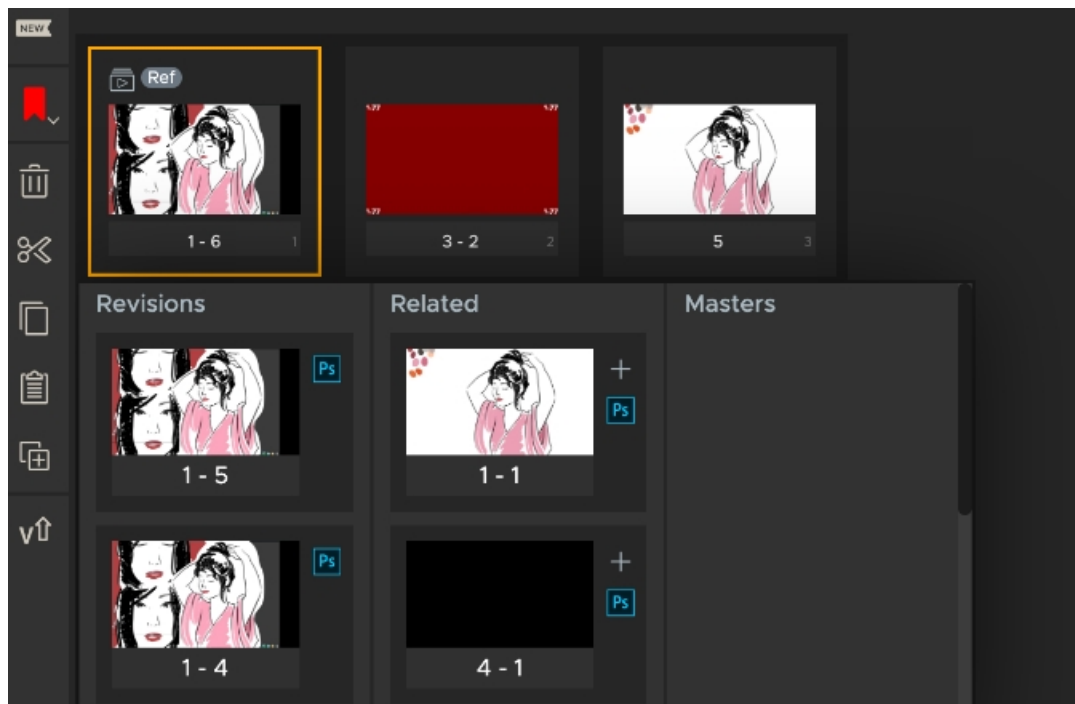
> 📝 **Note:** You may find that the **Related** panels column includes one or more versions of the same versioned-up reference panel ID. This is to ensure that the link between related panels and their artwork is maintained if the reference panel is duplicated.

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.

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

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

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 is 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 **distingishedName**, **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).

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.

Photoshop Actions

Flix links to Photoshop so you can edit your panels directly in Photoshop and send them back to Flix. Flix includes Photoshop actions, which you can access from Photoshop, to use this Flix and Photoshop workflow.

> **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.

To use Photoshop with Flix for the first time, before opening Photoshop, you first need to set up the required version of Photoshop and then install the plug-in:

1. Navigate to **File** > **Preferences > Third Party Apps**.
2. Set up the **Photoshop** preferences.

To open the Photoshop actions:

1. Open Photoshop.
2. Navigate to **Window** > **Extensions** > **Flix**.

   The Photoshop Actions open in a new tab called **Flix**.
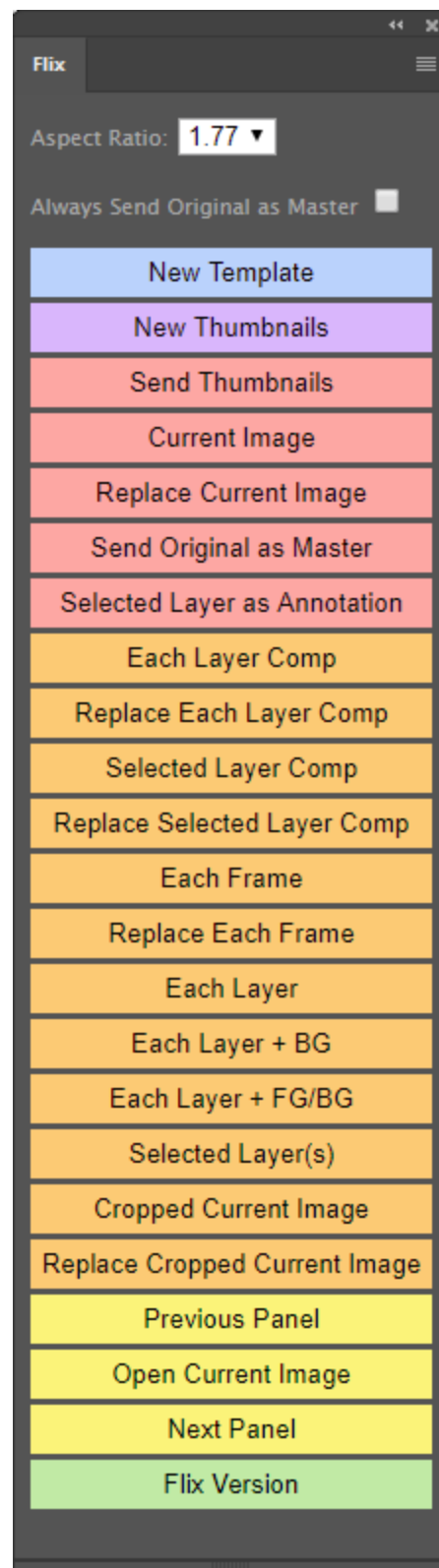
Select the aspect ratio at the top of the list to match your show's, then use any of the Photoshop Actions (see below) to edit and send panels to Flix.

> **Note:** Both Photoshop and Flix need to be open for these actions to function.

Action	Description

> **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.

Aspect ratio	Specifies the aspect ratio to set for the panels that are sent to Flix: • 1.0:1 • 1.77:1 • 1.85:1 • 2.2:1 • 2.35:1 • 2.39:1
Always Send Original as Master	When enabled, sends the current **.psd** file as a master image to Flix's Library.
New Template	Creates a new document with the aspect ratio specified in the **Aspect ratio** dropdown.
New Thumbnails	This creates a new document that contains nine thumbnails, each with the aspect ratio specified in the **Aspect ratio** dropdown.
Send Thumbnails	Sends thumbnails, created by the **New Thumbnails** action to Flix as individual panels.
Current Image	Sends the current image to Flix.
Replace Current Image	Replaces the currently-selected Flix panel with the image.
Send Original as Master	Exports the current **.psd** file as is, as a master image, which means that all information is saved such as hidden layers and empty groups. This is the only action that does not insert anything in the edit (sequence revision).
Selected Layer as Annotation	Sends the currently selected layer to Flix as an annotation.
Each Layer Comp	Imports or re-imports each layer comp as a separate panel with all the

Action	Description
	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.
Each Frame	Re-imports each frame in the **Frame Timeline** as a separate panel. This is only available if you are using the animation timeline workflow.
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.
Each Layer	Re-imports each visible layer in an image as a separate panel.
Each Layer + BG	Re-imports each visible layer in an image as a separate panel but keeps the background the same for each one.
Each Layer + FG/BG	Re-imports each visible layer in an image as a separate panel but keeps the background and foreground the same for each one.
Selected Layer(s)	Re-imports only the selected layers of the image as a new panel.
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.
Previous Panel	Selects the previous panel in the Flix current sequence from within Photoshop.
Open Current Image	Opens the currently-selected panel in Flix.
Next Panel	Selects the next panel in the Flix current sequence from within Photoshop.
Flix Version	Displays the current script version.

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.
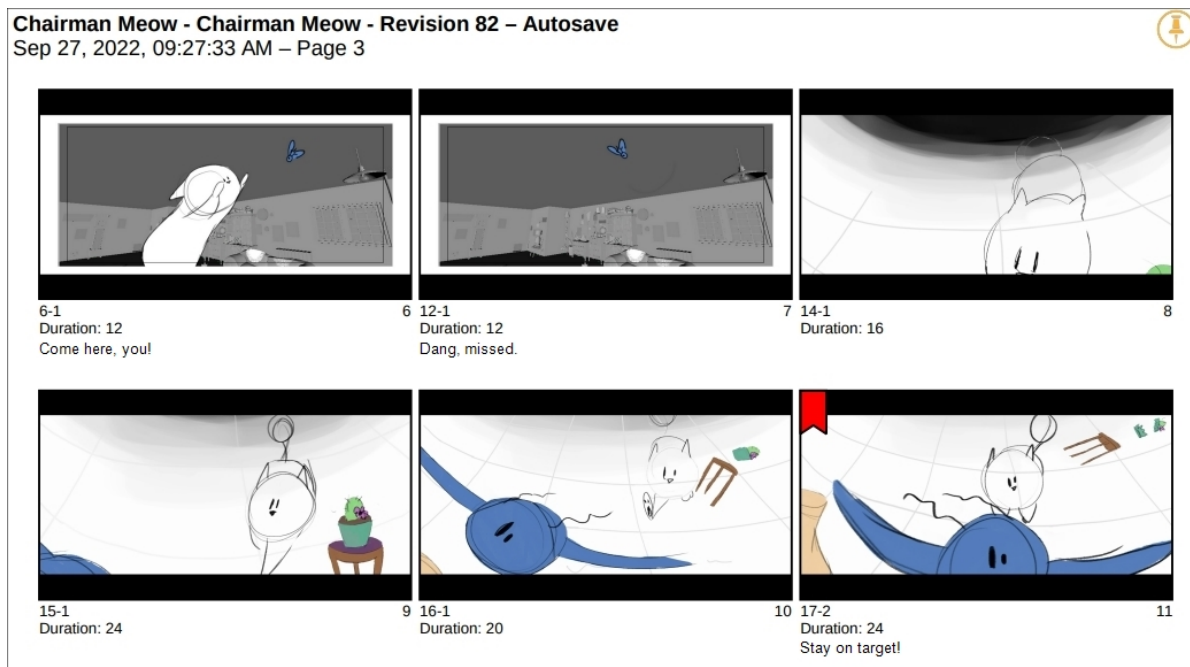
A 3x2 Storyboard (Landscape) Contact Sheet

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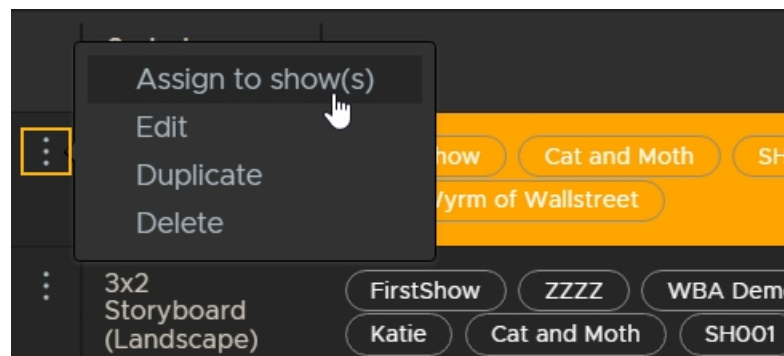
The **Assign to shows** dialog is displayed.
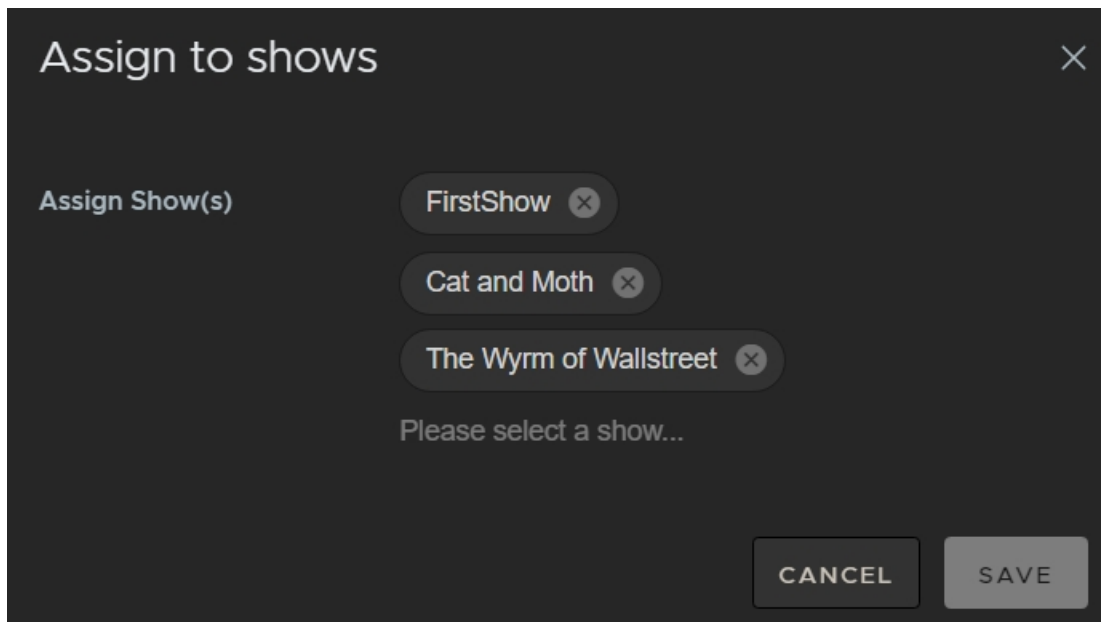
2. Click **Please select a show...** and then choose the show you want to assign.
3. Add as many shows as required and then click **Save**.

> **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.
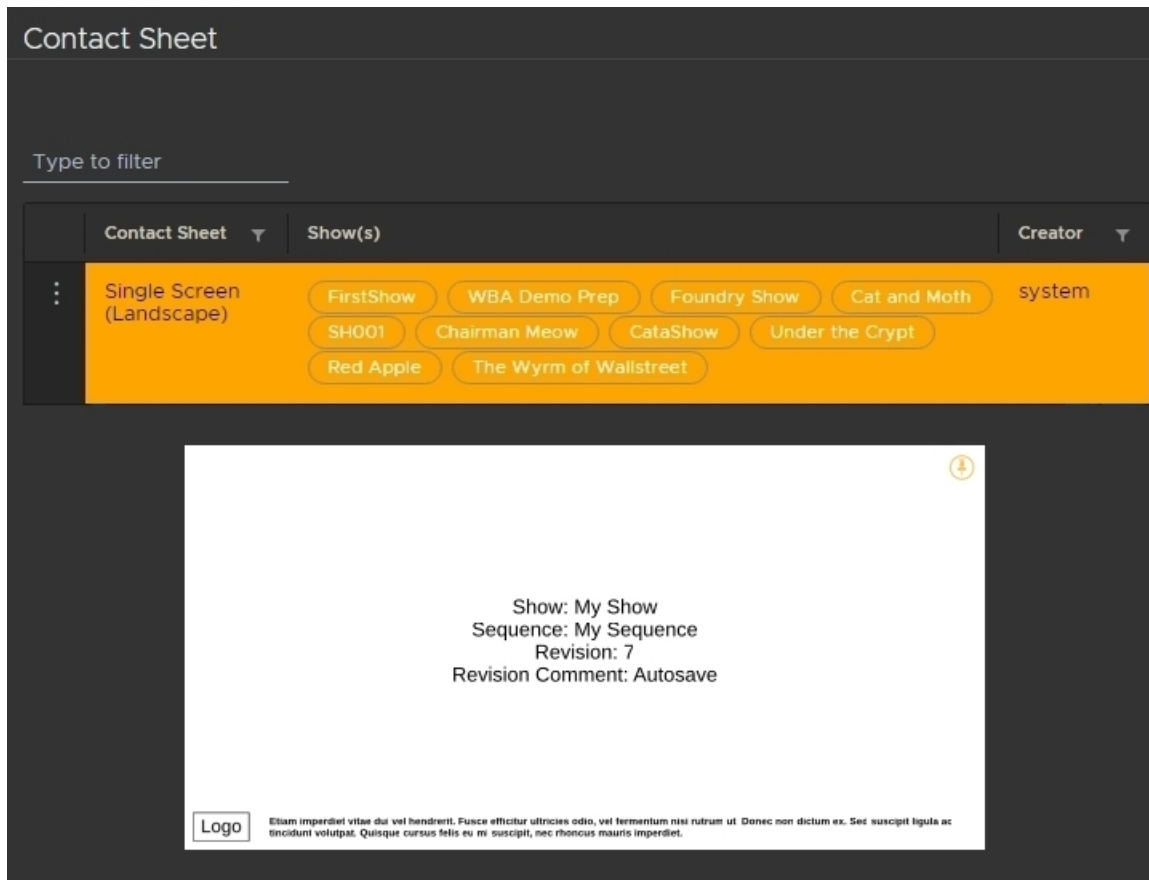
A Single Screen (Landscape) Contact Sheet and Preview

4. Click the menu ⋮ icon next to the template you want to edit and select **Edit**.
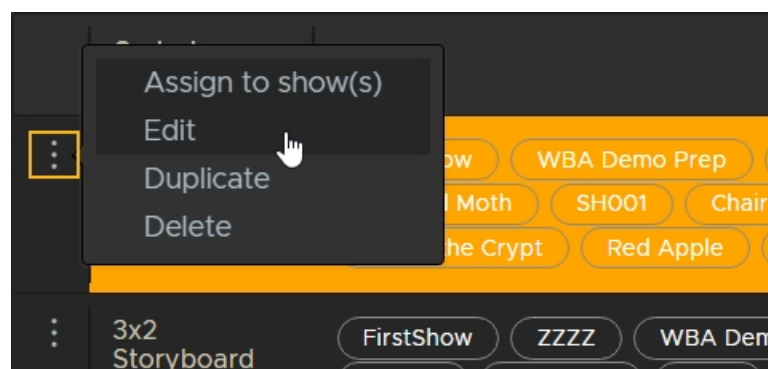
5. Make the required changes on the **General**, **Branding**, and **Cover** tabs.

> 📝 **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 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.

Importing Projects from Flix 5 to Flix 6

This page explains how to import projects from Flix 5 to Flix 6 using the Flix Migration Tool.

Requirements

The Migration Tool requires Python 3.7. Installing the dependencies using pip is recommended. You can follow the instructions found here: https://pip.pypa.io/en/stable/installing/

You can use the provided **requirements.txt** file for pip to install all the dependencies automatically, by running the following command:

python3 -m pip install -r requirements.txt

Flix Server Version Compatibility

With every release of Flix comes a version of the Migration Tool, which may be incompatible with earlier Flix versions.

The table below lists which release of Flix each Migration Tool version is compatible with.

Flix Server Version	Migration Tool Version
6.1.0	36
6.1.1	37
6.1.2	40
6.2.0	44
6.2.2	48
6.3.0	52
6.3.3	59

6.3.4	60
6.3.5	62
6.3.5-1	62
6.3.5-2	62
6.3.6	65
6.3.6-1	65
6.3.6-2	72
6.3.6-3	72
6.3.7	72

Usage

The **migrate5to6.py** file is the script that allows you to list or migrate your Flix 5 projects into Flix 6.

Required Arguments

--server SERVER - Flix 6 server url. For example 'http://flix001.mystudio.com:8080'

--user USER - Flix 6 client username. For example 'admin'.

--password PASSWORD - Flix 6 client password. For example 'admin'.

--flix-projects FLIX_PROJECTS - Path to your Flix 5 projects directory. For example '/mnt/flix/flixProjects'.

--list - List what the Migration Tool found from your Flix 5 project(s), without importing into Flix 6.

--migrate - Import what the Migration Tool found from your Flix 5 project(s) into Flix 6.

Optional Arguments

--help

--shows [SHOW NAMES [SHOW NAMES ...]] - Shows to list/migrate. By default, the Migration Tool will go through all the show names it finds in your Flix 5 projects directory. Make sure you use the show name, not the show tracking code.

--episodes [EPISODES [EPISODES ...]] - Episodes to list/migrate. By default, the Migration Tool will go through all the episodes in the current show.

--sequences [SEQUENCES [SEQUENCES ...]] - Sequences to list/migrate. By default, the Migration Tool will go through all the sequences in the current show or episode.

--revisions n - Only list/migrate the n most recent sequence revisions.

--all-revisions - Use this flag to list/migrate all the sequence revisions from the current sequence. By default, the Migration Tool will only import the most recent one.

--start-from-revision- Allows resuming a sequence migration starting from a specific revision. This only works with a single sequence. To migrate multiple sequences, use the **--revisions** flag.

--masters-only- Migrates only master panels from a given Flix 5 sequence. This option is used for cases where sequences have already been migrated, but master panels were excluded.

--port PORT - Download helper port.

--extra-checks - Flix migration tool only looks for certain filename patterns. Use this optional argument to include additional path patterns for Flix 5 assets.

Examples

• **python3 migrate5to6.py --server http://flix001.myStudio.com:8080 --user admin --password admin --flix-projects /mnt/flix/flixProjects --shows my_show --list**

   This command lists all the information from the show **my_show**.

• **python3 migrate5to6.py --server http://flix001.myStudio.com:8080 --user admin --password admin --flix-projects /mnt/flix/flixProjects --shows my_show --sequences my_sequence --migrate**

This command will migrate the last sequence revision from the sequence **my_sequence** from the show **my_show**.

- **python3 migrate5to6.py --server http://flix001.myStudio.com:8080 --user admin --password admin --flix-projects /mnt/flix/flixProjects --all-revisions --migrate**

    This command will migrate everything from your Flix 5 projects directory.

- **python3 migrate5to6.py --server http://flix001.myStudio.com:8080 --user admin --password admin --flix-projects /mnt/flix/flixProjects --shows my_show --sequences my_sequence --start-from-revision 35 --migrate**

    This command will migrate all sequence revisions, starting with revision 35, from the sequence **my_sequence** from the show **my_show**.

Reconform from Avid using Flix 5 assets

flix5_compatible_imports - For cases where Flix 5 sequences sent to Avid have not been sent back to Flix prior to a Flix 5 to Flix 6 migration. After having run a migration, it is possible for Flix 6 to recognize Flix 5 assets coming from Avid and relink them to the ones now available in Flix 6. To enable this behavior, you will need to set the **flix5_compatible_imports** option to **true** in your server's **config.yml** file.

> **Note:** When enabled, this may cause slowdowns while importing AAFs from Avid. We recommend disabling it once you have successfully migrated, reconformed all of your sequences in Flix 6 and there are no more Flix 5 assets left in your Avid project.

Flix Logs

There are three components to Flix that generate logs.

- **Flix Server** - Backend Processing
- **Flix Client** - User Interface
- **Flix Transfer Utility** - File Transfer

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 logs for all of your Flix Servers, click the **Download Logs** button.

• To access the log file for each server, click the **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.

To access Client Logs:

1. Go to **Preferences** > **General**.
2. Click the **Reveal Logs** button.

    A window opens to the location where Client Logs are stored.

Alternatively, they 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.

To access Transfer Logs:

1. Go to **Preferences** > **General**.
2. Click the **Reveal Logs** button.

   A window opens to the location where Transfer Logs are stored.

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**

Flix Resources

Flix Preferences

To access Flix's Preferences, navigate to **File** > **Preferences**.

> **Note:** These preferences can be enforced at the Studio and Show levels, see Settings & Preferences in **Flix for Production**.

General Settings

Show Splash Screen	Enables or disables Flix's splash screen at start-up.
Enable Autosave	Enables or disables autosaving of sequence revisions.
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.
Reset Preferences	Resets all preferences to their default values.
Logs	Opens the file location for Flix client logs.

Third-Party Apps

General	
Sketching	
Sketching Tool	Specifies the external application used to edit panels, Adobe Photoshop or Storyboard Pro.

Editorial	
Editorial Tool	Specifies the external application used for editorial roundtripping, Avid Media Composer or Adobe Premiere.
Editorial Clip Name	Specifies the naming convention of panels exported as clips for editorial. **Note:** This setting also determines the naming convention used for clip names in editorial publish email notifications.
Editorial Sequence Name	Specifies the naming convention of the sequence exported to editorial. **Note:** The chips located under the Editorial Clip Name and Editorial Sequence Name preferences show what information type you can include in the filename format. Click a chip to add it to your filename convention. See Setting Naming Conventions for more information.
Publish Directory	Sets the location where Flix stores published files for Editorial. The chips, located at the bottom of the preferences panel show what information type you can include in the filename format. See Setting Naming Conventions for more information.

	Note: Flix automatically creates missing directories if they don't already exist.
	Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **${HOME}** on macOS.

Adobe Photoshop	
Executable	Specifies the file path to the version of Photoshop 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.
Panel Open Behavior	Specifies how to open panels in Photoshop: • **Open as Separate PSD** - opens the selected panels in Photoshop as separate **.psd** files. • **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 Master Image	When enabled, opening a panel in Photoshop opens the master image, 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.
Install Plugin	Installs the Photoshop scripts to run the Photoshop actions. Restarting Photoshop is required after installing the Flix plugins. **Note:** Photoshop 2022 support requires some extra set up before you install the plug-in: • Launch Photoshop 2022 through the Creative Cloud application at least once before you start the Flix Client. • On Windows, you must run Flix with administrator permissions to install the plug-in. See Installing Plug-ins for Photoshop 2022 for more information.
Adobe Premiere	
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.

	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.
Highlight New Panels	Sets whether new panels appear as highlighted clips in your Premiere sequence.
Avid Media Composer	
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.
Clip Duration	Specifies the default duration for clips sent to Media Composer in frames. (For example, 480 frames or 20 seconds for a 24fps show).
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.
Enable Mark In	Toggle ON to include a Mark In point on each clip.
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).
Enable Mark Out	Toggle ON to include a Mark Out point on each clip.
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.
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. • **PNG and 3D Warp** - this option

	preserves the camera move keyframes from Stroyboard Pro so that you can edit them in Avid and send them back to Flix without having to reproduce the effect.
	• **MOV** - the camera moves from Storyboard Pro are burned into the 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 Avid. If you disable this control, you can only send **.mov** files to Avid and the **Panels with Camera Move** control is disabled.
New Clip Color	Sets the color of new clips in your sequence.
Color Range	Sets either **Full** or **Legal** range color for publishes to Avid. 📝 **Note:** The default color range is set to **Full**.
Enable Audio	Controls whether audio is available in Avid Media Composer or not.
Storyboard Pro	
Executable	Specifies the location of the Storyboard Pro executable file on your computer.

	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.

Export	
Export Path	Set the path for your exported Flix sequence.
Export Camera Moves	Toggle ON to update Camera Moves in Storyboard Pro.
Export Markers	Marker updates are not currently supported in Storyboard Pro.

Exporting

Exporting		
Filename Format (Dialogue, Audio, JSON, CSV, QuickTime, PDF)	Specifies the filename convention of your exported files for Dialogue, Audio, JSON, CSV, QuickTime, and PDF files.	**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.
Filename Format (Original Artwork, Image)	Specifies the filename convention of your exported files for Original Artwork and Image files.	
Default export path	Sets the default location where Flix stores exported panels.	

	Note: Flix automatically creates missing directories if they don't already exist.
	Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **${HOME}** on macOS.
QuickTime Export	
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

Audio	
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

Panel Browser	
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).
Markers	Toggle ON to add a button on the Panel Browser for adding markers in a sequence.

Marker Name Format	Specifies the naming convention of markers.
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.
Shot Number Format	
Minimum Length	Sets the minimum character length for your marker numbers. **Note:** Your marker name format must include the **[shot_number]** chip in order to display the shot number.
Increment	Sets the value of increments between markers. For example, if the shot number format is set to increments of 10, your first marker is labeled 0010, your second is 0020, 0030 etc.

Advanced

Advanced	
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.

Keyboard Shortcuts

Project

Open the Preferences	**Ctrl/Cmd**+**,** (comma)
Open the Management Console	**Ctrl/Cmd**+**.** (period)
Save	**Ctrl/Cmd**+**S**

Edit

Next Panel	Right arrow
Previous Panel	Left arrow
Create new panels	**Ctrl/Cmd**+**N**
Open selected panels in Photoshop	**Ctrl/Cmd**+**Enter**
Add a new line of dialogue in the dialogue box	**Enter**
Go to next panel when typing dialogue	**Tab**
Go to previous panel when typing dialogue	**Shift**+**Tab**

Undo	**Ctrl/Cmd**+**Z**
Redo	**Ctrl/Cmd**+**Shift**+**Z**
Cut Panels	**Ctrl/Cmd+X**
Copy Panels	**Ctrl/Cmd+C**
Paste Panels	**Ctrl/Cmd+V**
Delete	**Backspace**
Select All	**Ctrl/Cmd**+**A**

Player

| Add an Audience to the **Viewer** | **A** |
| Exit Full Screen Mode | **Esc** |

Window

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**

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.
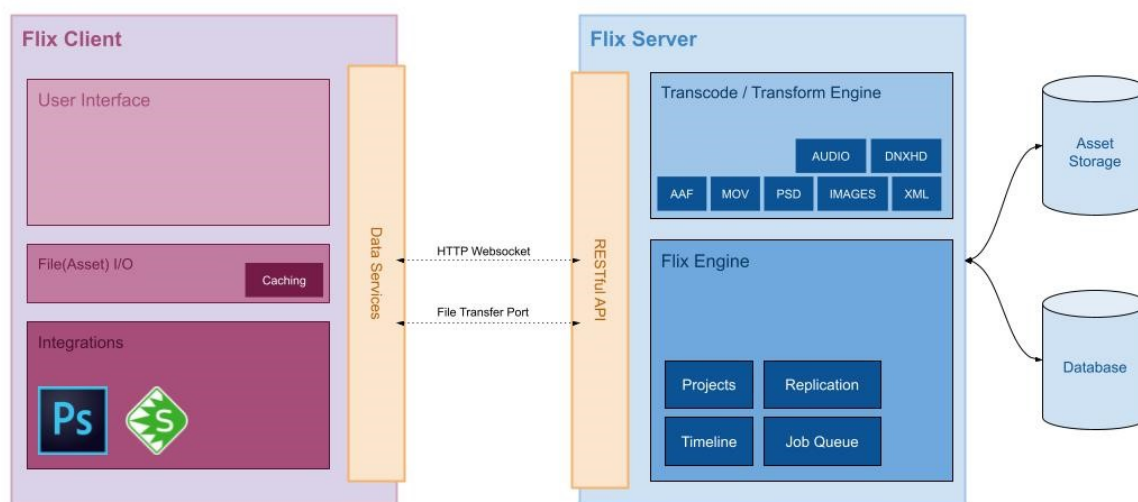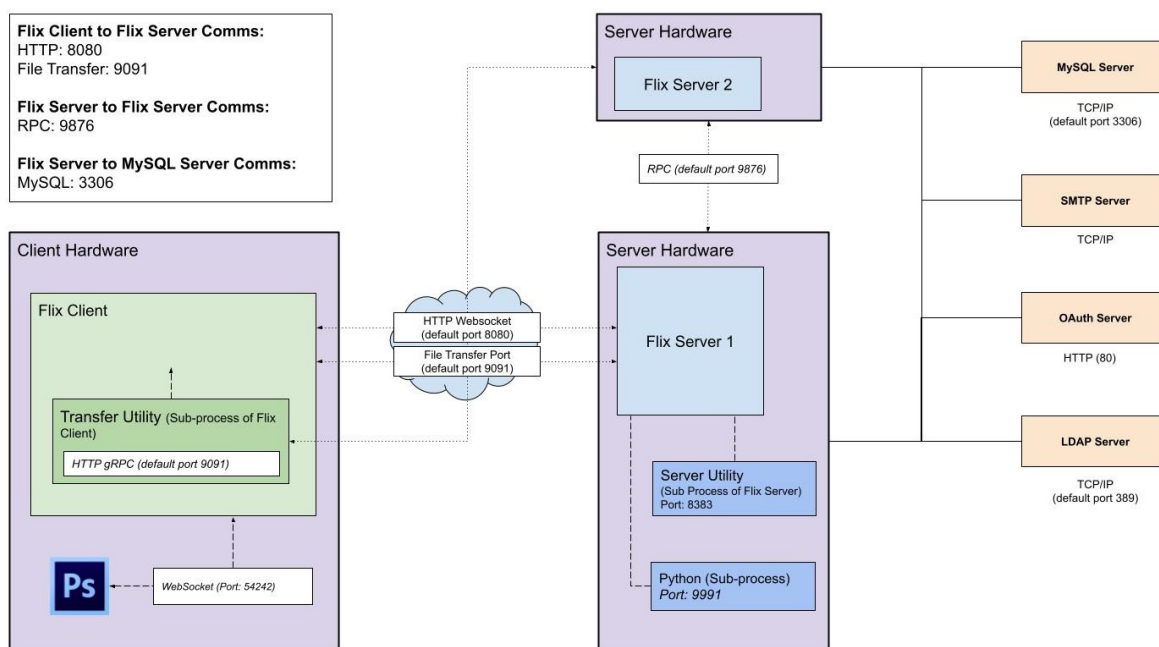
Server Requirements

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.
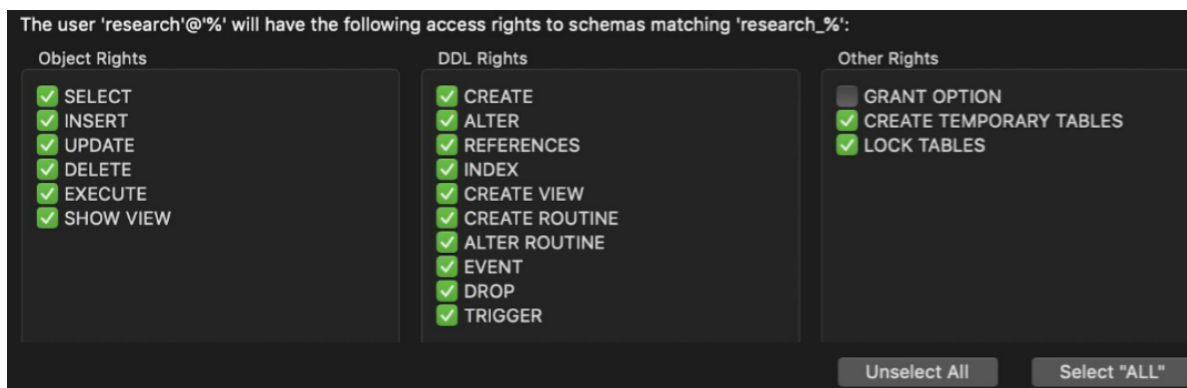
API and Websockets

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.
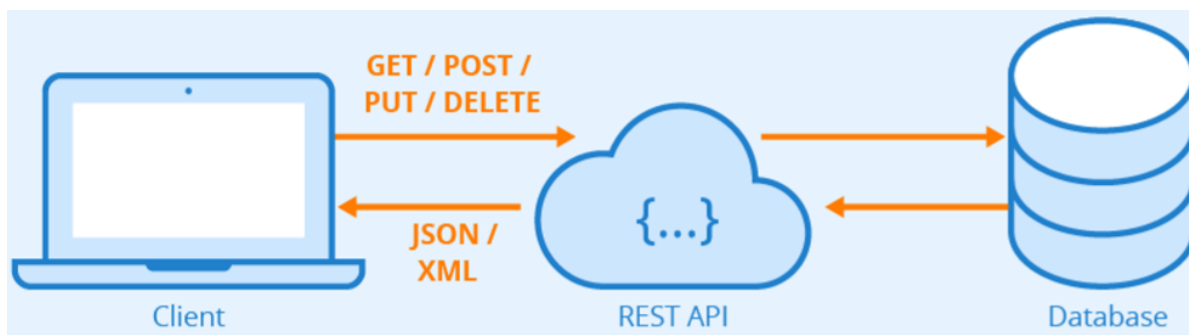
Flix also uses websockets for realtime communication between the client and server. We use websockets mostly for signalling purposes. For example, when the server has completed a long-running operation, it then notifies the client using a websocket message that it is complete, allowing the UI to be updated. 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	Significance
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	245.7	0.5	0	>30	Warning: chunk	

Import				min	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	
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 Developers API Reference

Flix's open, flexible API gives you freedom to integrate Flix with your pipeline and create custom tools. There are two versions to choose from depending on whether you're running Flix 6.4 and earlier or Flix 6.5 and later.

Flix 6.4 and Earlier

A reference guide to Flix's API for versions 6.4 and earlier.

Flix 6.5 and Later

A reference guide to Flix's API for versions 6.5 and later.

Supported File Formats

The following tables list file formats and codec support in Flix and compatible third-party applications.

Panel Formats

Format	Extension
Stills	
JPEG	.jpeg, .jpg
PNG	.png
PSD	.psd
OGG	.ogg
TIFF	.tiff

Flix and Storyboard Pro 20 Container Formats

For more information on format support in Storyboard Pro 20, see About Video Clip Format.

Container	Codec
Windows	
.mov	H.264
	Motion JPEG
	MPEG-4
.wmv	Windows Media Video (WMV)
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.

Container	Codec
Windows	
.mov	M-JPEG
	MPEG-1
	MPEG-2
	MPEG-3
	MPEG-4
	H.264
	DVC Pro 50
	Apple ProRes
	DNxHD
.wmv	Windows Media Video (WMV)
macOS	
.mov	M-JPEG
	MPEG-2
	H.264
	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.

Container	Codec
Windows	
.mov	M-JPEG
	MPEG-1
	MPEG-2
	MPEG-3
	MPEG-4
	H.264
	DVC Pro 50
	Apple ProRes
	DNxHD

> 📝 **Note:** From Media Composer version 2020.4 onwards, DNxHD resolution names have been simplified:

In Flix	In Avid Media Composer
DNxHD 36, 45	DNxHD LB
DNxHD 115, 120, and 145	DNxHD SQ

Container	Codec
macOS	
.mov	M-JPEG
	MPEG-2
	H.264
	Apple ProRes
	DNxHD

Container	Codec
	Note: From Media Composer version 2020.4 onwards, DNxHD resolution names have been simplified:

In Flix	In Avid Media Composer
DNxHD 36, 45	DNxHD LB
DNxHD 115, 120, and 145	DNxHD SQ

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 6.6.0

Application	Supported Versions
Adobe Photoshop	CC 2021, 2022

Note: Photoshop 2022 support requires some extra set up before you install the plug-in, see Installing Plug-ins for Photoshop 2022 for more information.

Application	Supported Versions
Adobe Premiere	CC 2021, 2022
Toon Boom Storyboard Pro	2020 (20.10.2, or later) and 2022

Avid Media Composer	2020, 2021
Operating System	**Supported Versions**
Flix Server	
Linux	CentOS 7
Flix Client	
Windows OS	Windows 10 Windows 11
macOS	11.x (Big Sur) 12.x (Monterey) 13.x (Ventura)

Flix 6.5.0

Application	Supported Versions
Adobe Photoshop	CC 2021, 2022

> 📝 **Note:** Photoshop 2022 support requires some extra set up before you install the plug-in, see Installing Plug-ins for Photoshop 2022 for more information.

Adobe Premiere	CC 2021, 2022
Toon Boom Storyboard Pro	Storyboard Pro 2020 (20.10.2, or later)
Avid Media Composer	2020, 2021
Operating System	**Supported Versions**
Flix Server	
Linux	CentOS 7
Flix Client	

Windows OS	Windows 10
	Windows 11
macOS	10.15.x (Catalina)
	11.x (Big Sur)
	12.x (Monterey)

Flix 6.4.0

> **Note:** From Flix 6.5.0 onwards, Photoshop CC 2020 and Avid Media Composer 2018 will no longer be supported.

Application	Supported Versions
Adobe Photoshop	CC 2020, 2021, 2022

> **Note:** Photoshop 2022 support requires some extra set up before you install the plug-in, see Installing Plug-ins for Photoshop 2022 for more information.

Application	Supported Versions
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	2018, 2019, 2020
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

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	2018, 2019, 2020
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

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	2018, 2019, 2020
Operating System	**Supported Versions**
Windows OS	10
macOS	10.14 (Mojave)

	10.15.x (Catalina)
Linux	CentOS 6
	CentOS 7

Flix 6.3.5

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6
Avid Media Composer	8.10, 2018
Operating System	**Supported Versions**
Windows OS	10
macOS	10.13 (High Sierra)
	10.14 (Mojave)
Linux	CentOS 6
	CentOS 7

Flix 6.3.4

Application	Supported Versions
Adobe Photoshop	CC 2018, 2019
Adobe Premiere	CC 2018, 2019
Toon Boom Storyboard Pro	6

Avid Media Composer	8.10, 2018
Operating System	**Supported Versions**
Windows OS	10
macOS	10.13 (High Sierra)
	10.14 (Mojave)
Linux	CentOS 6
	CentOS 7

Flix 6.3.3

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
Operating System	**Supported Versions**
Windows OS	10
macOS	10.12 (Sierra)
	10.13 (High Sierra)
	10.14 (Mojave)
Linux	CentOS 6
	CentOS 7

Flix 6.3.2

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
Operating System	**Supported Versions**
Windows	10
macOS	10.12 (Sierra)
	10.15.x (Catalina)
	11.x (Big Sur)
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.

Third-Party Package Versions

Flix Server

Name	Version	License
cloud.google.com/go	v0.100.2	Apache-2.0
cloud.google.com/go/bigquery	v1.8.0	Apache-2.0
cloud.google.com/go/compute	v1.6.1	Apache-2.0
cloud.google.com/go/datastore	v1.1.0	Apache-2.0
cloud.google.com/go/pubsub	v1.3.1	Apache-2.0
cloud.google.com/go/storage	v1.10.0	Apache-2.0
dmitri.shuralyov.com/gpu/mtl	v0.0.0-20190408044501-666a987793e9	BSD-3-Clause
fyne.io/fyne/v2	v2.2.1	BSD-3-Clause
fyne.io/systray	v1.10.0	Apache-2.0
gioui.org	v0.0.0-20220615093012-72669e19bc29	MIT
gioui.org/cpu	v0.0.0-20220412190645-f1e9e8c3b1f7	MIT
gioui.org/shader	v1.0.6	MIT
git.sr.ht/~sbinet/gg	v0.3.1	MIT
github.com/BurntSushi/toml	v0.3.1	MIT
github.com/BurntSushi/xgb	v0.0.0-20160522181843-27f122750802	BSD-3-Clause
github.com/ByteArena/poly2tri-go	v0.0.0-20170716161910-d102ad91854f	BSD-3-Clause

Name	Version	License
github.com/CloudyKit/fastprinter	v0.0.0-20200109182630-33d98a066a53	MIT
github.com/CloudyKit/jet/v3	v3.0.0	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/Masterminds/squirrel	v1.5.3	MIT
github.com/OneOfOne/xxhash	v1.2.2	Apache-2.0
github.com/Shopify/goreferrer	v0.0.0-20181106222321-ec9c9a553398	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/ajg/form	v1.5.1	BSD-3-Clause
github.com/ajstarks/svgo	v0.0.0-20211024235047-1546f124cd8b	CC-BY-4.0
github.com/antihax/optional	v1.0.0	MIT
github.com/arvidfm/jsonschema	v0.6.0-1	MIT
github.com/aymerick/raymond	v2.0.3-0.20180322193309-b565731e1464	MIT
github.com/benoitkugler/pstokenizer	v1.0.0	MIT
github.com/benoitkugler/textlayout	v0.1.3	MIT
github.com/benoitkugler/textlayout-testdata	v0.1.1	MIT

Name	Version	License
github.com/blang/semver	v3.5.1	MIT
github.com/boombuler/barcode	v1.0.1	MIT
github.com/census-instrumentation/opencensus-proto	v0.2.1	Apache-2.0
github.com/cespare/xxhash	v1.1.0	MIT
github.com/cespare/xxhash/v2	v2.1.1	MIT
github.com/cheekybits/is	v0.0.0-20150225183255-68e9c0620927	MIT
github.com/chzyer/logex	v1.1.10	MIT
github.com/chzyer/readline	v0.0.0-20180603132655-2972be24d48e	MIT
github.com/chzyer/test	v0.0.0-20180213035817-a1ea475d72b1	MIT
github.com/client9/misspell	v0.3.4	MIT
github.com/cncf/udpa/go	v0.0.0-20210930031921-04548b0d99d4	Apache-2.0
github.com/cncf/xds/go	v0.0.0-20211011173535-cb28da3451f1	Apache-2.0
github.com/codegangsta/inject	v0.0.0-20150114235600-33e0aa1cb7c0	MIT
github.com/corona10/goimagehash	v1.1.0	BSD-2-Clause
github.com/davecgh/go-spew	v1.1.1	ISC
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	v1.0.0	MIT

Name	Version	License
github.com/eknkc/amber	v0.0.0-20171010120322-cdade1c07385	MIT
github.com/envoyproxy/go-control-plane	v0.10.2-0.20220325020618-49ff273808a1	Apache-2.0
github.com/envoyproxy/protoc-gen-validate	v0.1.0	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/fogleman/gg	v1.3.0	MIT
github.com/fredbi/uri	v0.0.0-20181227131451-3dcfdacbaaf3	MIT
github.com/fsnotify/fsnotify	v1.5.4	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	v0.13.0	BSD-2-Clause
github.com/ghodss/yaml	v1.0.0	MIT
github.com/gin-contrib/sse	v0.1.0	MIT
github.com/gin-gonic/gin	v1.7.7	MIT
github.com/gioui/uax	v0.2.1-0.20220325163150-e3d987515a12	BSD-3-Clause
github.com/go-errors/errors	v1.0.1	MIT
github.com/go-fonts/dejavu	v0.1.0	BSD-3-Clause

Name	Version	License
github.com/go-fonts/latin-modern	v0.2.0	BSD-3-Clause
github.com/go-fonts/liberation	v0.2.0	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	v0.0.0-20190409004039-e6da0acd62b1	BSD-3-Clause
github.com/go-gl/glfw/v3.3/glfw	v0.0.0-20220622232848-a6c407ee30a0	BSD-3-Clause
github.com/go-latex/latex	v0.0.0-20210823091927-c0d11ff05a81	BSD-3-Clause
github.com/go-martini/martini	v0.0.0-20170121215854-22fa46961aab	MIT
github.com/go-pdf/fpdf	v0.6.0	MIT
github.com/go-playground/locales	v0.13.0	MIT
github.com/go-playground/universal-translator	v0.17.0	MIT
github.com/go-playground/validator/v10	v10.4.1	MIT
github.com/go-sql-driver/mysql	v1.6.0	MPL-2.0
github.com/go-text/typesetting	v0.0.0-20220411150340-35994bc27a7b	BSD-3-Clause
github.com/godbus/dbus/v5	v5.1.0	BSD-2-Clause
github.com/gofrs/uuid	v4.2.0	MIT
github.com/gojuno/minimock/v3	v3.0.10	MIT
github.com/goki/freetype	v0.0.0-20220119013949-7a161fd3728c	FTL

Name	Version	License
github.com/golang-jwt/jwt	v3.2.2	MIT
github.com/golang/freetype	v0.0.0-20170609003504-e2365dfdc4a0	FTL
github.com/golang/glog	v0.0.0-20160126235308-23def4e6c14b	Apache-2.0
github.com/golang/groupcache	v0.0.0-20210331224755-41bb18bfe9da	Apache-2.0
github.com/golang/mock	v1.6.0	Apache-2.0
github.com/golang/protobuf	v1.5.2	BSD-3-Clause
github.com/golang/snappy	v0.0.3	BSD-3-Clause
github.com/google/btree	v1.0.0	Apache-2.0
github.com/google/go-cmp	v0.5.8	BSD-3-Clause
github.com/google/go-querystring	v1.0.0	BSD-3-Clause
github.com/google/gofuzz	v1.2.0	Apache-2.0
github.com/google/martian	v2.1.0	Apache-2.0
github.com/google/martian/v3	v3.2.1	Apache-2.0
github.com/google/pprof	v0.0.0-20210720184732-4bb14d4b1be1	Apache-2.0
github.com/google/renameio	v0.1.0	Apache-2.0
github.com/google/uuid	v1.3.0	BSD-3-Clause
github.com/googleapis/enterprise-certificate-proxy	v0.1.0	Apache-2.0
github.com/googleapis/gax-go/v2	v2.4.0	BSD-3-Clause
github.com/gopherjs/gopherjs	v1.17.2	BSD-2-

Name	Version	License
		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
github.com/hashicorp/golang-lru	v0.5.1	MPL-2.0
github.com/hpcloud/tail	v1.0.0	MIT
github.com/iancoleman/orderedmap	v0.0.0-20190318233801-ac98e3ecb4b0	MIT
github.com/ianlancetaylor/demangle	v0.0.0-20200824232613-28f6c0f3b639	BSD-3-Clause
github.com/imkira/go-interpol	v1.1.0	MIT
github.com/inconshreveable/mousetrap	v1.0.0	Apache-2.0
github.com/iris-contrib/blackfriday	v2.0.0	BSD-2-Clause
github.com/iris-contrib/jade	v1.1.3	BSD-3-Clause
github.com/iris-contrib/pongo2	v0.0.1	MIT
github.com/iris-contrib/schema	v0.0.1	BSD-3-Clause
github.com/jmoiron/sqlx	v1.3.5	MIT
github.com/json-iterator/go	v1.1.9	MIT
github.com/jstemmer/go-junit-report	v0.9.1	MIT

Name	Version	License
github.com/jsummers/gobmp	v0.0.0-20151104160322-e2ba15ffa76e	MIT
github.com/jung-kurt/gofpdf	v1.0.0	MIT
github.com/kataras/golog	v0.0.10	BSD-3-Clause
github.com/kataras/iris/v12	v12.1.8	BSD-3-Clause
github.com/kataras/pio	v0.0.2	BSD-3-Clause
github.com/kataras/sitemap	v0.0.5	MIT
github.com/kisielk/gotool	v1.0.0	MIT
github.com/klauspost/compress	v1.9.7	BSD-3-Clause
github.com/klauspost/cpuid	v1.2.0	MIT
github.com/kr/pretty	v0.1.0	MIT
github.com/kr/pty	v1.1.1	MIT
github.com/kr/text	v0.1.0	MIT
github.com/labstack/echo/v4	v4.5.0	MIT
github.com/labstack/gommon	v0.3.0	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	v1.2.0	MIT
github.com/matryer/try	v0.0.0-20161228173917-9ac251b645a2	MIT
github.com/mattn/go-colorable	v0.1.11	MIT
github.com/mattn/go-isatty	v0.0.14	MIT

Name	Version	License
github.com/microcosm-cc/bluemonday	v1.0.2	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	v1.0.1	Apache-2.0
github.com/moul/http2curl	v1.0.0	MIT
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
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	v0.0.0-20220121172623-5db5eafcecbb	MIT
github.com/paulmach/orb	v0.7.1	MIT
github.com/paulmach/osm	v0.5.0	MIT
github.com/phpdave11/gofpdf	v1.4.2	MIT
github.com/phpdave11/gofpdi	v1.0.13	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/errors	v0.9.1	BSD-2-

Name	Version	License
		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/rogpeppe/fastuuid	v1.2.0	BSD-3-Clause
github.com/rogpeppe/go-internal	v1.3.0	BSD-3-Clause
github.com/ruudk/golang-pdf417	v0.0.0-20201230142125-a7e3863a1245	MIT
github.com/ryanuber/columnize	v2.1.0	MIT
github.com/schollz/closestmatch	v2.1.0	MIT
github.com/sergi/go-diff	v1.0.0	MIT
github.com/shurcooL/go	v0.0.0-20200502201357-93f07166e636	MIT
github.com/shurcooL/httpfs	v0.0.0-20190707220628-8d4bc4ba7749	MIT
github.com/shurcooL/sanitized_anchor_name	v1.0.0	MIT
github.com/shurcooL/vfsgen	v0.0.0-20200824052919-0d455de96546	MIT
github.com/sirupsen/logrus	v1.8.1	MIT
github.com/smartystreets/goconvey	v1.6.4	MIT
github.com/spaolacci/murmur3	v0.0.0-20180118202830-f09979ecbc72	BSD-3-Clause
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

Name	Version	License
github.com/srwiley/rasterx	v0.0.0-20220615024203-67b7089efd25	BSD-3-Clause
github.com/stretchr/objx	v0.1.0	MIT
github.com/stretchr/testify	v1.7.3	MIT
github.com/tdewolff/argp	v0.0.0-20221007181215-ebef9ed8a560	MIT
github.com/tdewolff/minify	v2.3.6	MIT
github.com/tdewolff/minify/v2	v2.12.4	MIT
github.com/tdewolff/parse	v2.3.4	MIT
github.com/tdewolff/parse/v2	v2.6.4	MIT
github.com/tdewolff/test	v1.0.7	MIT
github.com/tevino/abool	v1.2.0	MIT
github.com/ugorji/go/codec	v1.1.7	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	v1.6.0	MIT
github.com/valyala/fasttemplate	v1.2.1	MIT
github.com/wcharczuk/go-chart/v2	v2.1.0	MIT
github.com/xeipuuv/gojsonschema	v1.2.0	Apache-2.0
github.com/yalp/jsonpath	v0.0.0-20180802001716-5cc68e5049a0	BSD-3-Clause
github.com/yudai/gojsondiff	v1.0.0	MIT
github.com/yudai/golcs	v0.0.0-20170316035057-ecda9a501e82	MIT

Name	Version	License
github.com/yuin/goldmark	v1.4.12	MIT
go.opencensus.io	v0.23.0	Apache-2.0
go.opentelemetry.io/proto/otlp	v0.7.0	Apache-2.0
golang.org/x/crypto	v0.0.0-20220525230936-793ad666bf5e	BSD-3-Clause
golang.org/x/exp	v0.0.0-20220613132600-b0d781184e0d	BSD-3-Clause
golang.org/x/exp/shiny	v0.0.0-20220613132600-b0d781184e0d	BSD-3-Clause
golang.org/x/image	v0.0.0-20220617043117-41969df76e82	BSD-3-Clause
golang.org/x/lint	v0.0.0-20210508222113-6edffad5e616	BSD-3-Clause
golang.org/x/mobile	v0.0.0-20220518205345-8578da9835fd	BSD-3-Clause
golang.org/x/mod	v0.6.0-dev.0.20220106191415-9b9b3d81d5e3	BSD-3-Clause
golang.org/x/net	v0.0.0-20220617184016-355a448f1bc9	BSD-3-Clause
golang.org/x/oauth2	v0.0.0-20220608161450-d0670ef3b1eb	BSD-3-Clause
golang.org/x/sync	v0.0.0-20220601150217-0de741cfad7f	BSD-3-Clause
golang.org/x/sys	v0.2.0	BSD-3-Clause
golang.org/x/term	v0.0.0-20210927222741-03fcf44c2211	BSD-3-Clause
golang.org/x/text	v0.3.7	BSD-3-

Name	Version	License
		Clause
golang.org/x/time	v0.0.0-20201208040808-7e3f01d25324	BSD-3-Clause
golang.org/x/tools	v0.1.10	BSD-3-Clause
golang.org/x/xerrors	v0.0.0-20220609144429-65e65417b02f	BSD-3-Clause
gonum.org/v1/gonum	v0.11.0	BSD-3-Clause
gonum.org/v1/plot	v0.11.0	BSD-3-Clause
google.golang.org/api	v0.84.0	BSD-3-Clause
google.golang.org/appengine	v1.6.7	Apache-2.0
google.golang.org/genproto	v0.0.0-20220614165028-45ed7f3ff16e	Apache-2.0
google.golang.org/grpc	v1.48.0	Apache-2.0
google.golang.org/grpc/cmd/protoc-gen-go-grpc	v1.1.0	Apache-2.0
google.golang.org/protobuf	v1.28.0	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	v1.0.0-20180628173108-788fd7840127	BSD-2-Clause
gopkg.in/errgo.v2	v2.1.0	BSD-3-Clause
gopkg.in/fsnotify.v1	v1.4.7	BSD-3-

Name	Version	License
		Clause
gopkg.in/gomail.v2	v2.0.0-20160411212932-81ebce5c23df	MIT
gopkg.in/ini.v1	v1.51.1	Apache-2.0
gopkg.in/ldap.v2	v2.5.1	MIT
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/js/dom	v0.0.0-20210725211120-f030747120f2	MIT
honnef.co/go/tools	v0.0.1-2020.1.4	MIT
rsc.io/binaryregexp	v0.2.0	BSD-3-Clause
rsc.io/pdf	v0.1.1	BSD-3-Clause
rsc.io/quote/v3	v3.1.0	BSD-3-Clause
rsc.io/sampler	v1.3.0	BSD-3-Clause

Transfer Utility

Name	Version	License
cloud.google.com/go	v0.34.0	Apache-2.0

Name	Version	License
github.com/BurntSushi/toml	v0.3.1	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/v3	v3.0.0	Apache-2.0
github.com/DATA-DOG/go-sqlmock	v1.3.3	BSD-3-Clause
github.com/DATA-DOG/godog	v0.7.13	BSD-3-Clause
github.com/Shopify/goreferrer	v0.0.0-20181106222321-ec9c9a553398	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/ajg/form	v1.5.1	BSD-3-Clause
github.com/alecthomas/template	v0.0.0-20160405071501-a0175ee3bccc	BSD-3-Clause
github.com/alecthomas/units	v0.0.0-20151022065526-2efee857e7cf	MIT
github.com/antihax/optional	v1.0.0	MIT
github.com/aymerick/raymond	v2.0.3-0.20180322193309-b565731e1464	MIT
github.com/benoitkugler/textlayout	v0.1.3	MIT
github.com/beorn7/perks	v1.0.0	MIT
github.com/census-instrumentation/opencensus-proto	v0.2.1	Apache-2.0

Name	Version	License
github.com/cespare/xxhash/v2	v2.1.1	MIT
github.com/client9/misspell	v0.3.4	MIT
github.com/cncf/udpa/go	v0.0.0-20210930031921-04548b0d99d4	Apache-2.0
github.com/cncf/xds/go	v0.0.0-20211011173535-cb28da3451f1	Apache-2.0
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	v0.10.2-0.20220325020618-49ff273808a1	Apache-2.0
github.com/envoyproxy/protoc-gen-validate	v0.1.0	Apache-2.0
github.com/fatih/structs	v1.1.0	MIT
github.com/fsnotify/fsnotify	v1.4.9	BSD-3-Clause
github.com/getsentry/sentry-go	v0.13.0	BSD-2-Clause
github.com/ghodss/yaml	v1.0.0	MIT
github.com/gin-contrib/sse	v0.1.0	MIT
github.com/gin-gonic/gin	v1.7.7	MIT
github.com/go-errors/errors	v1.0.1	MIT

Name	Version	License
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
github.com/go-martini/martini	v0.0.0-20170121215854-22fa46961aab	MIT
github.com/go-playground/locales	v0.13.0	MIT
github.com/go-playground/universal-translator	v0.17.0	MIT
github.com/go-playground/validator/v10	v10.4.1	MIT
github.com/go-sql-driver/mysql	v1.6.0	MPL-2.0
github.com/go-stack/stack	v1.8.0	MIT
github.com/gogo/protobuf	v1.1.1	BSD-3-Clause
github.com/gojuno/minimock/v3	v3.0.10	MIT
github.com/golang-jwt/jwt	v3.2.2	MIT
github.com/golang/freetype	v0.0.0-20170609003504-e2365dfdc4a0	FTL
github.com/golang/glog	v0.0.0-20160126235308-23def4e6c14b	Apache-2.0
github.com/golang/mock	v1.1.1	Apache-2.0
github.com/golang/protobuf	v1.5.2	BSD-3-Clause
github.com/google/go-cmp	v0.5.8	BSD-3-Clause
github.com/google/go-querystring	v1.0.0	BSD-3-Clause
github.com/google/gofuzz	v1.2.0	Apache-2.0
github.com/google/uuid	v1.3.0	BSD-3-

Name	Version	License
		Clause
github.com/gopherjs/gopherjs	v1.17.2	BSD-2-Clause
github.com/gorilla/websocket	v1.4.0	BSD-2-Clause
github.com/grpc-ecosystem/grpc-gateway	v1.16.0	BSD-3-Clause
github.com/hexdigest/gowrap	v1.1.8	MIT
github.com/hpcloud/tail	v1.0.0	MIT
github.com/imkira/go-interpol	v1.1.0	MIT
github.com/inconshreveable/mousetrap	v1.0.0	Apache-2.0
github.com/iris-contrib/blackfriday	v2.0.0	BSD-2-Clause
github.com/iris-contrib/jade	v1.1.3	BSD-3-Clause
github.com/iris-contrib/pongo2	v0.0.1	MIT
github.com/iris-contrib/schema	v0.0.1	BSD-3-Clause
github.com/jmoiron/sqlx	v1.3.5	MIT
github.com/json-iterator/go	v1.1.9	MIT
github.com/julienschmidt/httprouter	v1.2.0	BSD-3-Clause
github.com/kataras/golog	v0.0.10	BSD-3-Clause
github.com/kataras/iris/v12	v12.1.8	BSD-3-Clause

Name	Version	License
github.com/kataras/pio	v0.0.2	BSD-3-Clause
github.com/kataras/sitemap	v0.0.5	MIT
github.com/klauspost/compress	v1.9.7	BSD-3-Clause
github.com/konsorten/go-windows-terminal-sequences	v1.0.1	MIT
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	v4.5.0	MIT
github.com/labstack/gommon	v0.3.0	MIT
github.com/leodido/go-urn	v1.2.0	MIT
github.com/mattn/go-colorable	v0.1.11	MIT
github.com/mattn/go-isatty	v0.0.14	MIT
github.com/matttproud/golang_protobuf_extensions	v1.0.1	Apache-2.0
github.com/microcosm-cc/bluemonday	v1.0.2	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	v1.0.1	Apache-2.0
github.com/moul/http2curl	v1.0.0	MIT
github.com/mwitkow/go-conntrack	v0.0.0-20161129095857-cc309e4a2223	Apache-2.0

Name	Version	License
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
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	v0.0.0-20220121172623-5db5eafcecbb	MIT
github.com/opentracing/opentracing-go	v1.0.2	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/rogpeppe/fastuuid	v1.2.0	BSD-3-Clause
github.com/ryanuber/columnize	v2.1.0	MIT
github.com/schollz/closestmatch	v2.1.0	MIT

Name	Version	License
github.com/sergi/go-diff	v1.0.0	MIT
github.com/shurcooL/go	v0.0.0-20200502201357-93f07166e636	MIT
github.com/shurcooL/httpfs	v0.0.0-20190707220628-8d4bc4ba7749	MIT
github.com/shurcooL/sanitized_anchor_name	v1.0.0	MIT
github.com/shurcooL/vfsgen	v0.0.0-20200824052919-0d455de96546	MIT
github.com/sirupsen/logrus	v1.8.1	MIT
github.com/smartystreets/goconvey	v1.6.4	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	v1.7.0	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	v1.1.7	MIT
github.com/urfave/negroni	v1.0.0	MIT
github.com/valyala/bytebufferpool	v1.0.0	MIT
github.com/valyala/fasthttp	v1.6.0	MIT
github.com/valyala/fasttemplate	v1.2.1	MIT
github.com/wcharczuk/go-chart/v2	v2.1.0	MIT
github.com/xeipuuv/gojsonschema	v1.2.0	Apache-2.0

Name	Version	License
github.com/yalp/jsonpath	v0.0.0-20180802001716-5cc68e5049a0	BSD-3-Clause
github.com/yudai/gojsondiff	v1.0.0	MIT
github.com/yudai/golcs	v0.0.0-20170316035057-ecda9a501e82	MIT
github.com/yuin/goldmark	v1.3.5	MIT
go.opentelemetry.io/proto/otlp	v0.7.0	Apache-2.0
golang.org/x/crypto	v0.0.0-20210921155107-089bfa567519	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	v0.6.0-dev.0.20220106191415-9b9b3d81d5e3	BSD-3-Clause
golang.org/x/net	v0.0.0-20220617184016-355a448f1bc9	BSD-3-Clause
golang.org/x/oauth2	v0.0.0-20200107190931-bf48bf16ab8d	BSD-3-Clause
golang.org/x/sync	v0.0.0-20210220032951-036812b2e83c	BSD-3-Clause
golang.org/x/sys	v0.2.0	BSD-3-Clause
golang.org/x/term	v0.0.0-20210927222741-03fcf44c2211	BSD-3-Clause
golang.org/x/text	v0.3.7	BSD-3-Clause

Name	Version	License
golang.org/x/time	v0.0.0-20201208040808-7e3f01d25324	BSD-3-Clause
golang.org/x/tools	v0.1.10	BSD-3-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	v1.4.0	Apache-2.0
google.golang.org/genproto	v0.0.0-20220614165028-45ed7f3ff16e	Apache-2.0
google.golang.org/grpc	v1.48.0	Apache-2.0
google.golang.org/protobuf	v1.28.0	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	v1.51.1	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	v3.0.0-20210107192922-496545a6307b	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
@ampproject/remapping	2.2.0	Apache-2.0
@angular-builders/custom-webpack	13.1.0	MIT
@angular-devkit/architect	0.1303.8	MIT
@angular-devkit/architect	0.1303.9	MIT
@angular-devkit/build-angular	13.3.8	MIT
@angular-devkit/build-webpack	0.1303.8	MIT
@angular-devkit/core	12.2.18	MIT

Package	Version	License
@angular-devkit/core	13.3.8	MIT
@angular-devkit/core	13.3.9	MIT
@angular-devkit/schematics	12.2.18	MIT
@angular-devkit/schematics	13.3.8	MIT
@angular-eslint/builder	13.5.0	MIT
@angular-eslint/bundled-angular-compiler	12.6.1	MIT
@angular-eslint/bundled-angular-compiler	13.5.0	MIT
@angular-eslint/eslint-plugin-template	13.5.0	MIT
@angular-eslint/eslint-plugin	13.5.0	MIT
@angular-eslint/schematics	13.5.0	MIT
@angular-eslint/template-parser	12.6.1	MIT
@angular-eslint/utils	13.5.0	MIT
@angular-slider/ngx-slider	2.0.3	MIT
@angular/animations	13.3.11	MIT
@angular/cdk	13.3.9	MIT
@angular/cli	13.3.8	MIT
@angular/common	13.3.11	MIT
@angular/compiler-cli	13.3.11	MIT
@angular/compiler	13.3.11	MIT
@angular/compiler	9.0.0	MIT
@angular/core	13.3.11	MIT
@angular/core	9.0.0	MIT
@angular/forms	13.3.11	MIT

Package	Version	License
@angular/language-service	13.3.11	MIT
@angular/material	13.3.9	MIT
@angular/platform-browser-dynamic	13.3.11	MIT
@angular/platform-browser	13.3.11	MIT
@angular/router	13.3.11	MIT
@assemblyscript/loader	0.10.1	Apache-2.0
@babel/code-frame	7.18.6	MIT
@babel/compat-data	7.18.8	MIT
@babel/core	7.16.12	MIT
@babel/core	7.18.9	MIT
@babel/generator	7.16.8	MIT
@babel/generator	7.18.9	MIT
@babel/helper-annotate-as-pure	7.16.7	MIT
@babel/helper-annotate-as-pure	7.18.6	MIT
@babel/helper-builder-binary-assignment-operator-visitor	7.18.9	MIT
@babel/helper-compilation-targets	7.18.9	MIT
@babel/helper-create-class-features-plugin	7.18.9	MIT
@babel/helper-create-regexp-features-plugin	7.18.6	MIT
@babel/helper-define-polyfill-provider	0.3.1	MIT
@babel/helper-environment-visitor	7.18.9	MIT
@babel/helper-explode-assignable-expression	7.18.6	MIT
@babel/helper-function-name	7.18.9	MIT

Package	Version	License
@babel/helper-hoist-variables	7.18.6	MIT
@babel/helper-member-expression-to-functions	7.18.9	MIT
@babel/helper-module-imports	7.18.6	MIT
@babel/helper-module-transforms	7.18.9	MIT
@babel/helper-optimise-call-expression	7.18.6	MIT
@babel/helper-plugin-utils	7.18.9	MIT
@babel/helper-remap-async-to-generator	7.18.9	MIT
@babel/helper-replace-supers	7.18.9	MIT
@babel/helper-simple-access	7.18.6	MIT
@babel/helper-skip-transparent-expression-wrappers	7.18.9	MIT
@babel/helper-split-export-declaration	7.18.6	MIT
@babel/helper-validator-identifier	7.18.6	MIT
@babel/helper-validator-option	7.18.6	MIT
@babel/helper-wrap-function	7.18.9	MIT
@babel/helpers	7.18.9	MIT
@babel/highlight	7.18.6	MIT
@babel/parser	7.18.9	MIT
@babel/plugin-bugfix-safari-id-destructuring-collision-in-function-expression	7.18.6	MIT
@babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining	7.18.9	MIT
@babel/plugin-proposal-async-generator-functions	7.16.8	MIT
@babel/plugin-proposal-class-properties	7.18.6	MIT

Package	Version	License
@babel/plugin-proposal-class-static-block	7.18.6	MIT
@babel/plugin-proposal-dynamic-import	7.18.6	MIT
@babel/plugin-proposal-export-namespace-from	7.18.9	MIT
@babel/plugin-proposal-json-strings	7.18.6	MIT
@babel/plugin-proposal-logical-assignment-operators	7.18.9	MIT
@babel/plugin-proposal-nullish-coalescing-operator	7.18.6	MIT
@babel/plugin-proposal-numeric-separator	7.18.6	MIT
@babel/plugin-proposal-object-rest-spread	7.18.9	MIT
@babel/plugin-proposal-optional-catch-binding	7.18.6	MIT
@babel/plugin-proposal-optional-chaining	7.18.9	MIT
@babel/plugin-proposal-private-methods	7.18.6	MIT
@babel/plugin-proposal-private-property-in-object	7.18.6	MIT
@babel/plugin-proposal-unicode-property-regex	7.18.6	MIT
@babel/plugin-syntax-async-generators	7.8.4	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-json-strings	7.8.3	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

Package	Version	License
@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-transform-arrow-functions	7.18.6	MIT
@babel/plugin-transform-async-to-generator	7.16.8	MIT
@babel/plugin-transform-block-scoped-functions	7.18.6	MIT
@babel/plugin-transform-block-scoping	7.18.9	MIT
@babel/plugin-transform-classes	7.18.9	MIT
@babel/plugin-transform-computed-properties	7.18.9	MIT
@babel/plugin-transform-destructuring	7.18.9	MIT
@babel/plugin-transform-dotall-regex	7.18.6	MIT
@babel/plugin-transform-duplicate-keys	7.18.9	MIT
@babel/plugin-transform-exponentiation-operator	7.18.6	MIT
@babel/plugin-transform-for-of	7.18.8	MIT
@babel/plugin-transform-function-name	7.18.9	MIT
@babel/plugin-transform-literals	7.18.9	MIT
@babel/plugin-transform-member-expression-literals	7.18.6	MIT
@babel/plugin-transform-modules-amd	7.18.6	MIT
@babel/plugin-transform-modules-commonjs	7.18.6	MIT
@babel/plugin-transform-modules-systemjs	7.18.9	MIT
@babel/plugin-transform-modules-umd	7.18.6	MIT
@babel/plugin-transform-named-capturing-groups-regex	7.18.6	MIT

Package	Version	License
@babel/plugin-transform-new-target	7.18.6	MIT
@babel/plugin-transform-object-super	7.18.6	MIT
@babel/plugin-transform-parameters	7.18.8	MIT
@babel/plugin-transform-property-literals	7.18.6	MIT
@babel/plugin-transform-regenerator	7.18.6	MIT
@babel/plugin-transform-reserved-words	7.18.6	MIT
@babel/plugin-transform-runtime	7.16.10	MIT
@babel/plugin-transform-shorthand-properties	7.18.6	MIT
@babel/plugin-transform-spread	7.18.9	MIT
@babel/plugin-transform-sticky-regex	7.18.6	MIT
@babel/plugin-transform-template-literals	7.18.9	MIT
@babel/plugin-transform-typeof-symbol	7.18.9	MIT
@babel/plugin-transform-unicode-escapes	7.18.6	MIT
@babel/plugin-transform-unicode-regex	7.18.6	MIT
@babel/preset-env	7.16.11	MIT
@babel/preset-modules	0.1.5	MIT
@babel/runtime-corejs3	7.18.9	MIT
@babel/runtime	7.16.7	MIT
@babel/template	7.16.7	MIT
@babel/template	7.18.6	MIT
@babel/traverse	7.18.9	MIT
@babel/types	7.18.9	MIT
@cds/city	1.1.0	OFL-1.1

Package	Version	License
@cds/core	6.0.4	MIT
@clr/angular	13.4.2	MIT
@clr/icons	13.0.2	MIT
@clr/ui	13.4.2	MIT
@colors/colors	1.5.0	MIT
@cspotcode/source-map-support	0.8.1	MIT
@csstools/postcss-progressive-custom-properties	1.3.0	CC0-1.0
@csstools/selector-specificity	2.0.2	CC0-1.0
@develar/schema-utils	2.6.5	MIT
@discoveryjs/json-ext	0.5.6	MIT
@electron/get	1.14.1	MIT
@electron/universal	1.2.1	MIT
@eslint/eslintrc	1.4.0	MIT
@fortawesome/angular-fontawesome	0.10.2	MIT
@fortawesome/fontawesome-common-types	6.1.0	MIT
@fortawesome/fontawesome-free	6.1.0	(CC-BY-4.0 AND OFL-1.1 AND MIT)
@fortawesome/fontawesome-svg-core	6.1.0	MIT
@fortawesome/free-regular-svg-icons	6.1.0	(CC-BY-4.0 AND MIT)
@fortawesome/free-solid-svg-icons	6.1.0	(CC-BY-4.0 AND MIT)
@gar/promisify	1.1.3	MIT
@grpc/grpc-js	1.6.7	Apache-2.0
@grpc/proto-loader	0.6.13	Apache-2.0

Package	Version	License
@hapi/hoek	9.3.0	BSD-3-Clause
@hapi/topo	5.1.0	BSD-3-Clause
@humanwhocodes/config-array	0.11.8	Apache-2.0
@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
@istanbuljs/load-nyc-config	1.1.0	ISC
@istanbuljs/schema	0.1.3	MIT
@jridgewell/gen-mapping	0.1.1	MIT
@jridgewell/gen-mapping	0.3.2	MIT
@jridgewell/resolve-uri	3.1.0	MIT
@jridgewell/set-array	1.1.2	MIT
@jridgewell/sourcemap-codec	1.4.14	MIT
@jridgewell/trace-mapping	0.3.14	MIT
@jridgewell/trace-mapping	0.3.9	MIT
@lit/reactive-element	1.3.4	BSD-3-Clause
@malept/cross-spawn-promise	1.1.1	Apache-2.0
@malept/flatpak-bundler	0.4.0	MIT
@mapbox/node-pre-gyp	1.0.9	BSD-3-Clause
@ngtools/webpack	13.3.8	MIT
@ngx-translate/core	13.0.0	MIT
@ngx-translate/http-loader	6.0.0	MIT
@nodelib/fs.scandir	2.1.5	MIT

Package	Version	License
@nodelib/fs.stat	2.0.5	MIT
@nodelib/fs.walk	1.2.8	MIT
@npmcli/fs	1.1.1	ISC
@npmcli/fs	2.1.1	ISC
@npmcli/git	2.1.0	ISC
@npmcli/installed-package-contents	1.0.7	ISC
@npmcli/move-file	1.1.2	MIT
@npmcli/move-file	2.0.0	MIT
@npmcli/node-gyp	1.0.3	ISC
@npmcli/promise-spawn	1.3.2	ISC
@npmcli/run-script	2.0.0	ISC
@nrwl/cli	14.5.2	MIT
@nrwl/devkit	13.1.3	MIT
@nrwl/tao	13.1.3	MIT
@nrwl/tao	14.5.2	MIT
@parcel/watcher	2.0.4	MIT
@phenomnomnominal/tsquery	4.2.0	MIT
@playwright/test	1.23.1	Apache-2.0
@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
@protobufjs/fetch	1.1.0	BSD-3-Clause

Package	Version	License
@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
@rollup/plugin-json	4.1.0	MIT
@rollup/plugin-node-resolve	13.3.0	MIT
@rollup/pluginutils	3.1.0	MIT
@schematics/angular	12.2.18	MIT
@schematics/angular	13.3.8	MIT
@sideway/address	4.1.4	BSD-3-Clause
@sideway/formula	3.0.0	BSD-3-Clause
@sideway/pinpoint	2.0.0	BSD-3-Clause
@sindresorhus/is	0.14.0	MIT
@szmarczak/http-timer	1.1.2	MIT
@tootallnate/once	1.1.2	MIT
@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	1.0.3	MIT
@tweenjs/tween.js	17.4.0	MIT
@types/body-parser	1.19.2	MIT

Package	Version	License
@types/bonjour	3.5.10	MIT
@types/component-emitter	1.2.11	MIT
@types/connect-history-api-fallback	1.3.5	MIT
@types/connect	3.4.35	MIT
@types/cookie	0.4.1	MIT
@types/cors	2.8.12	MIT
@types/debug	4.1.7	MIT
@types/eslint-scope	3.7.4	MIT
@types/eslint	8.4.5	MIT
@types/estree	0.0.39	MIT
@types/estree	0.0.51	MIT
@types/express-serve-static-core	4.17.29	MIT
@types/express	4.17.13	MIT
@types/fs-extra	9.0.13	MIT
@types/glob	7.2.0	MIT
@types/http-proxy	1.17.9	MIT
@types/jasmine	3.8.1	MIT
@types/jasminewd2	2.0.10	MIT
@types/json-schema	7.0.11	MIT
@types/json5	0.0.29	MIT
@types/long	4.0.2	MIT
@types/mime	1.3.2	MIT
@types/minimatch	3.0.5	MIT

Package	Version	License
@types/minimist	1.2.2	MIT
@types/ms	0.7.31	MIT
@types/node	16.11.45	MIT
@types/node	16.11.7	MIT
@types/normalize-package-data	2.4.1	MIT
@types/parse-json	4.0.0	MIT
@types/qs	6.9.7	MIT
@types/quill	1.3.10	MIT
@types/range-parser	1.2.4	MIT
@types/resolve	1.17.1	MIT
@types/retry	0.12.0	MIT
@types/semver	7.3.13	MIT
@types/serve-index	1.9.1	MIT
@types/serve-static	1.13.10	MIT
@types/sockjs	0.3.33	MIT
@types/trusted-types	2.0.2	MIT
@types/tween.js	17.2.0	MIT
@types/ws	8.5.3	MIT
@types/yargs-parser	21.0.0	MIT
@types/yargs	17.0.10	MIT
@typescript-eslint/eslint-plugin	5.47.0	MIT
@typescript-eslint/experimental-utils	5.27.1	MIT
@typescript-eslint/parser	5.47.0	BSD-2-Clause

Package	Version	License
@typescript-eslint/scope-manager	5.27.1	MIT
@typescript-eslint/scope-manager	5.47.0	MIT
@typescript-eslint/type-utils	5.47.0	MIT
@typescript-eslint/types	5.27.1	MIT
@typescript-eslint/types	5.47.0	MIT
@typescript-eslint/typescript-estree	5.27.1	BSD-2-Clause
@typescript-eslint/typescript-estree	5.47.0	BSD-2-Clause
@typescript-eslint/utils	5.27.1	MIT
@typescript-eslint/utils	5.47.0	MIT
@typescript-eslint/visitor-keys	5.27.1	MIT
@typescript-eslint/visitor-keys	5.47.0	MIT
@webassemblyjs/ast	1.11.1	MIT
@webassemblyjs/floating-point-hex-parser	1.11.1	MIT
@webassemblyjs/helper-api-error	1.11.1	MIT
@webassemblyjs/helper-buffer	1.11.1	MIT
@webassemblyjs/helper-numbers	1.11.1	MIT
@webassemblyjs/helper-wasm-bytecode	1.11.1	MIT
@webassemblyjs/helper-wasm-section	1.11.1	MIT
@webassemblyjs/ieee754	1.11.1	MIT
@webassemblyjs/leb128	1.11.1	Apache-2.0
@webassemblyjs/utf8	1.11.1	MIT
@webassemblyjs/wasm-edit	1.11.1	MIT
@webassemblyjs/wasm-gen	1.11.1	MIT

Package	Version	License
@webassemblyjs/wasm-opt	1.11.1	MIT
@webassemblyjs/wasm-parser	1.11.1	MIT
@webassemblyjs/wast-printer	1.11.1	MIT
@webcomponents/custom-elements	1.5.0	BSD-3-Clause
@xtuc/ieee754	1.2.0	BSD-3-Clause
@xtuc/long	4.2.2	Apache-2.0
@yarnpkg/lockfile	1.1.0	BSD-2-Clause
Flix	6.6.0	UNLICENSED
JSONStream	1.3.5	(MIT OR Apache-2.0)
abab	2.0.6	BSD-3-Clause
abbrev	1.1.1	ISC
accepts	1.3.8	MIT
acorn-import-assertions	1.8.0	MIT
acorn-jsx	5.3.2	MIT
acorn-walk	8.2.0	MIT
acorn	8.8.0	MIT
add-stream	1.0.0	MIT
adjust-sourcemap-loader	4.0.0	MIT
adm-zip	0.4.16	MIT
agent-base	6.0.2	MIT
agentkeepalive	4.2.1	MIT
aggregate-error	3.1.0	MIT
ajv-formats	2.1.0	MIT

Package	Version	License
ajv-formats	2.1.1	MIT
ajv-keywords	3.5.2	MIT
ajv-keywords	5.1.0	MIT
ajv	6.12.6	MIT
ajv	8.6.2	MIT
ajv	8.9.0	MIT
angular-split	13.2.0	Apache-2.0
ansi-align	3.0.1	ISC
ansi-colors	4.1.1	MIT
ansi-escapes	4.3.2	MIT
ansi-html-community	0.0.8	Apache-2.0
ansi-regex	2.1.1	MIT
ansi-regex	5.0.1	MIT
ansi-regex	6.0.1	MIT
ansi-styles	2.2.1	MIT
ansi-styles	3.2.1	MIT
ansi-styles	4.3.0	MIT
anymatch	3.1.2	ISC
app-builder-bin	4.0.0	MIT
app-builder-lib	23.1.0	MIT
app-root-path	3.0.0	MIT
aproba	2.0.0	ISC
are-we-there-yet	2.0.0	ISC

Package	Version	License
are-we-there-yet	3.0.0	ISC
arg	4.1.3	MIT
argparse	1.0.10	MIT
argparse	2.0.1	Python-2.0
aria-query	3.0.0	Apache-2.0
aria-query	4.2.2	Apache-2.0
array-flatten	1.1.1	MIT
array-flatten	2.1.2	MIT
array-ify	1.0.0	MIT
array-union	1.0.2	MIT
array-union	2.1.0	MIT
array-union	3.0.1	MIT
array-uniq	1.0.3	MIT
arrify	1.0.1	MIT
asar	3.1.0	MIT
asn1.js	5.4.1	MIT
asn1	0.2.6	MIT
assert-plus	1.0.0	MIT
assert	2.0.0	MIT
ast-types-flow	0.0.7	ISC
async-exit-hook	2.0.1	MIT
async	2.6.4	MIT
async	3.0.1	MIT

Package	Version	License
async	3.2.4	MIT
asynckit	0.4.0	MIT
at-least-node	1.0.0	ISC
atob	2.1.2	(MIT OR Apache-2.0)
autoprefixer	10.4.7	MIT
available-typed-arrays	1.0.5	MIT
aws-sign2	0.7.0	Apache-2.0
aws4	1.11.0	MIT
axios	0.25.0	MIT
axobject-query	2.0.2	Apache-2.0
axobject-query	2.2.0	Apache-2.0
babel-loader	8.2.5	MIT
babel-plugin-dynamic-import-node	2.3.3	MIT
babel-plugin-istanbul	6.1.1	BSD-3-Clause
babel-plugin-polyfill-corejs2	0.3.1	MIT
babel-plugin-polyfill-corejs3	0.5.2	MIT
babel-plugin-polyfill-regenerator	0.3.1	MIT
balanced-match	1.0.2	MIT
base64-js	1.5.1	MIT
base64id	2.0.0	MIT
batch	0.6.1	MIT
bcrypt-pbkdf	1.0.2	BSD-3-Clause
bent	7.3.12	Apache-2.0

Package	Version	License
big.js	5.2.2	MIT
binary-extensions	2.2.0	MIT
bl	4.1.0	MIT
bluebird-lst	1.0.9	MIT
bluebird	3.7.2	MIT
bn.js	4.12.0	MIT
bn.js	5.2.1	MIT
body-parser	1.20.0	MIT
bonjour	3.5.0	MIT
boolbase	1.0.0	ISC
boolean	3.2.0	MIT
boxen	4.2.0	MIT
boxen	5.1.2	MIT
brace-expansion	1.1.11	MIT
brace-expansion	2.0.1	MIT
braces	3.0.2	MIT
brorand	1.1.0	MIT
browserify-aes	1.2.0	MIT
browserify-cipher	1.0.1	MIT
browserify-des	1.0.2	MIT
browserify-rsa	4.1.0	MIT
browserify-sign	4.2.1	ISC
browserify-zlib	0.2.0	MIT

Package	Version	License
browserslist	4.21.2	MIT
buffer-alloc-unsafe	1.1.0	MIT
buffer-alloc	1.2.0	MIT
buffer-crc32	0.2.13	MIT
buffer-equal	1.0.0	MIT
buffer-fill	1.0.0	MIT
buffer-from	1.1.2	MIT
buffer-indexof	1.1.1	MIT
buffer-xor	1.0.3	MIT
buffer	5.7.1	MIT
buffer	6.0.3	MIT
builder-util-runtime	9.0.2	MIT
builder-util	23.0.9	MIT
builtin-modules	1.1.1	MIT
builtin-modules	3.3.0	MIT
builtin-status-codes	3.0.0	MIT
builtins	1.0.3	MIT
bytes	3.0.0	MIT
bytes	3.1.2	MIT
bytesish	0.4.4	(Apache-2.0 AND MIT)
cacache	15.3.0	ISC
cacache	16.1.1	ISC
cacheable-request	6.1.0	MIT

Package	Version	License
call-bind	1.0.2	MIT
callsites	3.1.0	MIT
camelcase-keys	6.2.2	MIT
camelcase	5.3.1	MIT
camelcase	6.3.0	MIT
caniuse-lite	1.0.30001368	CC-BY-4.0
caseless	0.12.0	Apache-2.0
chalk	1.1.3	MIT
chalk	2.4.2	MIT
chalk	3.0.0	MIT
chalk	4.1.0	MIT
chalk	4.1.2	MIT
chardet	0.7.0	MIT
chokidar	3.5.3	MIT
chownr	2.0.0	ISC
chrome-trace-event	1.0.3	MIT
chromium-pickle-js	0.2.0	MIT
ci-info	2.0.0	MIT
ci-info	3.3.2	MIT
cipher-base	1.0.4	MIT
circular-dependency-plugin	5.2.2	ISC
clean-stack	2.2.0	MIT
cli-boxes	2.2.1	MIT

Package	Version	License
cli-cursor	3.1.0	MIT
cli-spinners	2.6.1	MIT
cli-width	3.0.0	ISC
cliui	6.0.0	ISC
cliui	7.0.4	ISC
clone-deep	4.0.1	MIT
clone-response	1.0.3	MIT
clone	1.0.4	MIT
clone	2.1.2	MIT
codelyzer	6.0.2	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
colorette	2.0.19	MIT
colors	1.0.3	MIT
colors	1.4.0	MIT
combine-source-map	0.8.0	MIT
combined-stream	1.0.8	MIT
commander	1.0.4	MIT*
commander	2.20.3	MIT
commander	2.9.0	MIT

Package	Version	License
commander	5.1.0	MIT
commander	6.2.1	MIT
commander	8.3.0	MIT
commondir	1.0.1	MIT
compare-func	2.0.0	MIT
compare-version	0.1.2	MIT
component-emitter	1.3.0	MIT
compressible	2.0.18	MIT
compression	1.7.4	MIT
concat-map	0.0.1	MIT
concat-stream	1.6.2	MIT
config-chain	1.1.13	MIT
configstore	5.0.1	BSD-2-Clause
connect-history-api-fallback	1.6.0	MIT
connect	3.7.0	MIT
console-browserify	1.2.0	MIT
console-control-strings	1.1.0	ISC
constants-browserify	1.0.0	MIT
content-disposition	0.5.4	MIT
content-type	1.0.4	MIT
conventional-changelog-angular	5.0.13	ISC
conventional-changelog-atom	2.0.8	ISC
conventional-changelog-cli	2.2.2	MIT

Package	Version	License
conventional-changelog-codemirror	2.0.8	ISC
conventional-changelog-conventionalcommits	4.6.3	ISC
conventional-changelog-core	4.2.4	MIT
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
convert-source-map	1.1.3	MIT
convert-source-map	1.2.0	MIT
convert-source-map	1.8.0	MIT
cookie-signature	1.0.6	MIT
cookie	0.4.2	MIT
cookie	0.5.0	MIT
copy-anything	2.0.6	MIT
copy-webpack-plugin	10.2.1	MIT
core-js-compat	3.23.5	MIT
core-js-pure	3.23.5	MIT

Package	Version	License
core-js	3.20.3	MIT
core-util-is	1.0.2	MIT
core-util-is	1.0.3	MIT
cors	2.8.5	MIT
cosmiconfig	7.0.1	MIT
create-ecdh	4.0.4	MIT
create-hash	1.2.0	MIT
create-hmac	1.1.7	MIT
create-require	1.1.1	MIT
critters	0.0.16	Apache-2.0
cross-spawn	6.0.5	MIT
cross-spawn	7.0.3	MIT
crypto-browserify	3.12.0	MIT
crypto-js	4.1.1	MIT
crypto-random-string	2.0.0	MIT
css-blank-pseudo	3.0.3	CC0-1.0
css-has-pseudo	3.0.4	CC0-1.0
css-loader	6.5.1	MIT
css-prefers-color-scheme	6.0.3	CC0-1.0
css-select	4.3.0	BSD-2-Clause
css-selector-tokenizer	0.7.3	MIT
css-what	6.1.0	BSD-2-Clause
css	3.0.0	MIT

Package	Version	License
cssauron	1.4.0	MIT
cssdb	5.1.0	CC0-1.0
cssesc	3.0.0	MIT
cuint	0.2.2	MIT
custom-event	1.0.1	MIT
damerau-levenshtein	1.0.8	BSD-2-Clause
dargs	7.0.0	MIT
dashdash	1.14.1	MIT
date-format	4.0.13	MIT
dateformat	3.0.3	MIT
debug	2.6.9	MIT
debug	3.2.7	MIT
debug	4.3.3	MIT
debug	4.3.4	MIT
decamelize-keys	1.1.0	MIT
decamelize	1.2.0	MIT
decamelize	4.0.0	MIT
decode-uri-component	0.2.0	MIT
decompress-response	3.3.0	MIT
deep-diff	1.0.2	MIT
deep-equal	1.1.1	MIT
deep-extend	0.6.0	MIT
deep-is	0.1.4	MIT

Package	Version	License
deepmerge	4.2.2	MIT
default-gateway	6.0.3	BSD-2-Clause
defaults	1.0.3	MIT
defer-to-connect	1.1.3	MIT
define-lazy-prop	2.0.0	MIT
define-properties	1.1.4	MIT
del	2.2.2	MIT
del	6.1.1	MIT
delayed-stream	1.0.0	MIT
delegates	1.0.0	MIT
depd	1.1.2	MIT
depd	2.0.0	MIT
dependency-graph	0.11.0	MIT
des.js	1.0.1	MIT
destroy	1.2.0	MIT
detect-libc	2.0.1	Apache-2.0
detect-node	2.1.0	MIT
detect-passive-events	1.0.5	MIT
di	0.0.1	MIT
diff	4.0.2	BSD-3-Clause
diffie-hellman	5.0.3	MIT
dir-compare	2.4.0	MIT
dir-glob	3.0.1	MIT

Package	Version	License
dmg-builder	23.1.0	MIT
dns-equal	1.0.0	MIT
dns-packet	1.3.4	MIT
dns-txt	2.0.2	MIT
doctrine	3.0.0	Apache-2.0
dom-serialize	2.2.1	MIT
dom-serializer	1.4.1	MIT
domain-browser	4.22.0	MIT
domelementtype	2.3.0	BSD-2-Clause
domhandler	4.3.1	BSD-2-Clause
domutils	2.8.0	BSD-2-Clause
dot-prop	5.3.0	MIT
dotenv-expand	5.1.0	BSD-2-Clause
dotenv	10.0.0	BSD-2-Clause
dotenv	9.0.2	BSD-2-Clause
duplexer3	0.1.5	BSD-3-Clause
ecc-jsbn	0.1.2	MIT
ee-first	1.1.1	MIT
ejs	3.1.8	Apache-2.0
electron-builder	23.1.0	MIT
electron-log	4.4.8	MIT
electron-notarize	1.2.1	MIT
electron-osx-sign	0.6.0	BSD-2-Clause

Package	Version	License
electron-publish	23.0.9	MIT
electron-reload	1.5.0	MIT
electron-to-chromium	1.4.198	ISC
electron-updater	5.0.5	MIT
electron	19.0.8	MIT
elliptic	6.5.4	MIT
emoji-regex	8.0.0	MIT
emojis-list	3.0.0	MIT
encodeurl	1.0.2	MIT
encoding	0.1.13	MIT
end-of-stream	1.4.4	MIT
engine.io-parser	5.0.4	MIT
engine.io	6.2.0	MIT
enhanced-resolve	5.10.0	MIT
enquirer	2.3.6	MIT
ent	2.2.0	MIT
entities	2.2.0	BSD-2-Clause
env-paths	2.2.1	MIT
err-code	2.0.3	MIT
errno	0.1.8	MIT
error-ex	1.3.2	MIT
es-abstract	1.20.1	MIT
es-module-lexer	0.9.3	MIT

Package	Version	License
es-to-primitive	1.2.1	MIT
es6-error	4.1.1	MIT
es6-object-assign	1.1.0	MIT
esbuild-linux-64	0.14.22	MIT
esbuild-wasm	0.14.22	MIT
esbuild	0.14.22	MIT
escalade	3.1.1	MIT
escape-goat	2.1.1	MIT
escape-html	1.0.3	MIT
escape-string-regexp	1.0.5	MIT
escape-string-regexp	4.0.0	MIT
eslint-config-prettier	8.5.0	MIT
eslint-scope	5.1.1	BSD-2-Clause
eslint-scope	7.1.1	BSD-2-Clause
eslint-utils	3.0.0	MIT
eslint-visitor-keys	2.1.0	Apache-2.0
eslint-visitor-keys	3.3.0	Apache-2.0
eslint	8.30.0	MIT
espree	9.4.1	BSD-2-Clause
esprima	4.0.1	BSD-2-Clause
esquery	1.4.0	BSD-3-Clause
esrecurse	4.3.0	BSD-2-Clause
estraverse	4.3.0	BSD-2-Clause

Package	Version	License
estraverse	5.3.0	BSD-2-Clause
estree-walker	1.0.1	MIT
esutils	2.0.3	BSD-2-Clause
etag	1.8.1	MIT
eventemitter-asyncresource	1.0.0	MIT
eventemitter3	2.0.3	MIT
eventemitter3	4.0.7	MIT
events	3.3.0	MIT
evp_bytestokey	1.0.3	MIT
execa	5.1.1	MIT
express	4.18.1	MIT
extend	3.0.2	MIT
external-editor	3.1.0	MIT
extract-zip	1.7.0	BSD-2-Clause
extsprintf	1.3.0	MIT
fast-deep-equal	3.1.3	MIT
fast-diff	1.1.2	Apache-2.0
fast-glob	3.2.11	MIT
fast-glob	3.2.7	MIT
fast-json-stable-stringify	2.1.0	MIT
fast-levenshtein	2.0.6	MIT
fastparse	1.1.2	MIT
fastq	1.13.0	ISC

Package	Version	License
faye-websocket	0.11.4	Apache-2.0
fd-slicer	1.1.0	MIT
figures	3.2.0	MIT
file-entry-cache	6.0.1	MIT
filelist	1.0.4	Apache-2.0
fill-range	7.0.1	MIT
filter-obj	2.0.2	MIT
finalhandler	1.1.2	MIT
finalhandler	1.2.0	MIT
find-cache-dir	3.3.2	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
flat	5.0.2	BSD-3-Clause
flatted	3.2.6	ISC
follow-redirects	1.15.1	MIT
for-each	0.3.3	MIT
forever-agent	0.6.1	Apache-2.0
form-data	2.3.3	MIT
form-data	4.0.0	MIT
forwarded	0.2.0	MIT
fraction.js	4.2.0	MIT

Package	Version	License
fresh	0.5.2	MIT
fs-constants	1.0.0	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-monkey	1.0.3	Unlicense
fs.realpath	1.0.0	ISC
function-bind	1.1.1	MIT
function.prototype.name	1.1.5	MIT
functions-have-names	1.2.3	MIT
gauge	3.0.2	ISC
gauge	4.0.4	ISC
gensync	1.0.0-beta.2	MIT
get-caller-file	2.0.5	ISC
get-intrinsic	1.1.2	MIT
get-package-type	0.1.0	MIT
get-pkg-repo	4.2.1	MIT
get-stream	4.1.0	MIT
get-stream	5.2.0	MIT
get-stream	6.0.1	MIT
get-symbol-description	1.0.0	MIT
getpass	0.1.7	MIT

Package	Version	License
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-to-regexp	0.4.1	BSD-2-Clause
glob	7.1.4	ISC
glob	7.2.0	ISC
glob	8.0.3	ISC
global-agent	3.0.0	BSD-3-Clause
global-dirs	3.0.0	MIT
global-tunnel-ng	2.7.1	BSD-3-Clause
globals	11.12.0	MIT
globals	13.19.0	MIT
globalthis	1.0.3	MIT
globby	11.1.0	MIT
globby	12.2.0	MIT
globby	5.0.0	MIT
google-protobuf	3.15.8	BSD-3-Clause
got	9.6.0	MIT
graceful-fs	4.2.10	ISC
graceful-readlink	1.0.1	MIT

Package	Version	License
grapheme-splitter	1.0.4	MIT
grpc-tools	1.11.2	Apache*
grpc_tools_node_protoc_ts	5.3.2	MIT
handle-thing	2.0.1	MIT
handlebars	4.7.7	MIT
har-schema	2.0.0	ISC
har-validator	5.1.5	MIT
hard-rejection	2.1.0	MIT
has-ansi	2.0.0	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-symbols	1.0.3	MIT
has-tostringtag	1.0.0	MIT
has-unicode	2.0.1	ISC
has-yarn	2.1.0	MIT
has	1.0.3	MIT
hash-base	3.1.0	MIT
hash.js	1.1.7	MIT
hdr-histogram-js	2.0.3	BSD*
hdr-histogram-percentiles-obj	3.0.0	MIT
hmac-drbg	1.0.1	MIT

Package	Version	License
hosted-git-info	2.8.9	ISC
hosted-git-info	4.1.0	ISC
hpack.js	2.1.6	MIT
html-entities	2.3.3	MIT
html-escaper	2.0.2	MIT
http-cache-semantics	4.1.0	BSD-2-Clause
http-deceiver	1.2.7	MIT
http-errors	1.6.3	MIT
http-errors	2.0.0	MIT
http-parser-js	0.5.8	MIT
http-proxy-agent	4.0.1	MIT
http-proxy-agent	5.0.0	MIT
http-proxy-middleware	2.0.6	MIT
http-proxy	1.18.1	MIT
http-signature	1.2.0	MIT
https-browserify	1.0.0	MIT
https-proxy-agent	5.0.0	MIT
human-signals	2.1.0	Apache-2.0
humanize-ms	1.2.1	MIT
iconv-lite	0.4.24	MIT
iconv-lite	0.6.3	MIT
icss-utils	5.1.0	ISC
ieee754	1.2.1	BSD-3-Clause

Package	Version	License
ignore-walk	4.0.1	ISC
ignore	5.2.0	MIT
image-size	0.5.5	MIT
immutable	4.1.0	MIT
import-fresh	3.3.0	MIT
import-lazy	2.1.0	MIT
imurmurhash	0.1.4	MIT
indent-string	4.0.0	MIT
infer-owner	1.0.4	ISC
inflight	1.0.6	ISC
inherits	2.0.3	ISC
inherits	2.0.4	ISC
ini	1.3.8	ISC
ini	2.0.0	ISC
injection-js	2.4.0	MIT
inline-source-map	0.6.2	MIT
inquirer	8.2.0	MIT
internal-slot	1.0.3	MIT
ip	1.1.8	MIT
ip	2.0.0	MIT
ipaddr.js	1.9.1	MIT
ipaddr.js	2.0.1	MIT
is-arguments	1.1.1	MIT

Package	Version	License
is-arrayish	0.2.1	MIT
is-bigint	1.0.4	MIT
is-binary-path	2.1.0	MIT
is-boolean-object	1.1.2	MIT
is-builtin-module	3.1.0	MIT
is-callable	1.2.4	MIT
is-ci	2.0.0	MIT
is-ci	3.0.1	MIT
is-core-module	2.9.0	MIT
is-date-object	1.0.5	MIT
is-docker	2.2.1	MIT
is-extglob	2.1.1	MIT
is-fullwidth-code-point	3.0.0	MIT
is-generator-function	1.0.10	MIT
is-glob	4.0.3	MIT
is-installed-globally	0.4.0	MIT
is-interactive	1.0.0	MIT
is-lambda	1.0.1	MIT
is-module	1.0.0	MIT
is-nan	1.3.2	MIT
is-negative-zero	2.0.2	MIT
is-npm	5.0.0	MIT
is-number-object	1.0.7	MIT

Package	Version	License
is-number	7.0.0	MIT
is-obj	2.0.0	MIT
is-path-cwd	1.0.0	MIT
is-path-cwd	2.2.0	MIT
is-path-in-cwd	1.0.1	MIT
is-path-inside	1.0.1	MIT
is-path-inside	3.0.3	MIT
is-plain-obj	1.1.0	MIT
is-plain-obj	3.0.0	MIT
is-plain-object	2.0.4	MIT
is-regex	1.1.4	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-text-path	1.0.1	MIT
is-typed-array	1.1.9	MIT
is-typedarray	1.0.0	MIT
is-unicode-supported	0.1.0	MIT
is-weakref	1.0.2	MIT
is-what	3.14.1	MIT
is-wsl	2.2.0	MIT
is-yarn-global	0.3.0	MIT

Package	Version	License
isarray	1.0.0	MIT
isbinaryfile	3.0.3	MIT
isbinaryfile	4.0.10	MIT
isexe	2.0.0	ISC
isobject	3.0.1	MIT
isstream	0.1.2	MIT
istanbul-lib-coverage	2.0.5	BSD-3-Clause
istanbul-lib-coverage	3.2.0	BSD-3-Clause
istanbul-lib-instrument	5.2.0	BSD-3-Clause
istanbul-lib-report	3.0.0	BSD-3-Clause
istanbul-lib-source-maps	3.0.6	BSD-3-Clause
istanbul-lib-source-maps	4.0.1	BSD-3-Clause
istanbul-reports	3.1.5	BSD-3-Clause
jake	10.8.5	Apache-2.0
jasmine-core	3.8.0	MIT
jasmine-spec-reporter	7.0.0	Apache-2.0
jest-worker	27.5.1	MIT
joi	17.6.0	BSD-3-Clause
jquery	3.6.0	MIT
js-sdsl	4.2.0	MIT
js-string-escape	1.0.1	MIT
js-tokens	4.0.0	MIT
js-yaml	3.14.1	MIT

Package	Version	License
js-yaml	4.1.0	MIT
jsbn	0.1.1	MIT
jsesc	0.5.0	MIT
jsesc	2.5.2	MIT
json-buffer	3.0.0	MIT
json-parse-better-errors	1.0.2	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-schema	0.4.0	(AFL-2.1 OR BSD-3-Clause)
json-stable-stringify-without-jsonify	1.0.1	MIT
json-stringify-safe	5.0.1	ISC
json2csv	5.0.7	MIT
json5	1.0.1	MIT
json5	2.2.1	MIT
jsonc-parser	3.0.0	MIT
jsonfile	4.0.0	MIT
jsonfile	6.1.0	MIT
jsonparse	1.3.1	MIT
jsprim	1.4.2	MIT
karma-chrome-launcher	3.1.1	MIT
karma-coverage-istanbul-reporter	3.0.3	MIT
karma-coverage	2.2.0	MIT

Package	Version	License
karma-electron	7.0.0	Unlicense
karma-jasmine-html-reporter	1.7.0	MIT
karma-jasmine	4.0.1	MIT
karma-source-map-support	1.4.0	MIT
karma-sourcemap-loader	0.3.8	MIT
karma-spec-reporter	0.0.34	MIT
karma	6.4.0	MIT
keyboardevent-from-electron-accelerator	2.0.0	MIT
keyboardevents-areequal	0.2.2	MIT
keypress	0.1.0	MIT
keyv	3.1.0	MIT
kind-of	6.0.3	MIT
kleur	3.0.3	MIT
klona	2.0.5	MIT
latest-version	5.1.0	MIT
lazy-val	1.0.5	MIT
less-loader	10.2.0	MIT
less	4.1.2	Apache-2.0
levn	0.4.1	MIT
license-webpack-plugin	4.0.2	ISC
line-by-line	0.1.6	MIT
lines-and-columns	1.2.4	MIT
lit-element	3.2.2	BSD-3-Clause

Package	Version	License
lit-html	2.2.7	BSD-3-Clause
lit	2.2.8	BSD-3-Clause
load-json-file	4.0.0	MIT
loader-runner	4.3.0	MIT
loader-utils	2.0.2	MIT
loader-utils	3.2.0	MIT
locate-path	2.0.0	MIT
locate-path	5.0.0	MIT
locate-path	6.0.0	MIT
lodash.camelcase	4.3.0	MIT
lodash.debounce	4.0.8	MIT
lodash.escaperegexp	4.1.2	MIT
lodash.get	4.4.2	MIT
lodash.isequal	4.5.0	MIT
lodash.ismatch	4.4.0	MIT
lodash.memoize	3.0.4	MIT
lodash.merge	4.6.2	MIT
lodash	4.17.21	MIT
log-symbols	4.1.0	MIT
log4js	6.6.0	Apache-2.0
long	4.0.0	Apache-2.0
lowercase-keys	1.0.1	MIT
lowercase-keys	2.0.0	MIT

Package	Version	License
lru-cache	6.0.0	ISC
lru-cache	7.13.1	ISC
magic-string	0.25.7	MIT
magic-string	0.26.2	MIT
make-dir	2.1.0	MIT
make-dir	3.1.0	MIT
make-error	1.3.6	ISC
make-fetch-happen	10.2.0	ISC
make-fetch-happen	9.1.0	ISC
map-obj	1.0.1	MIT
map-obj	4.3.0	MIT
matcher	3.0.0	MIT
material-icons	1.11.4	Apache-2.0
md5.js	1.3.5	MIT
media-typer	0.3.0	MIT
memfs	3.4.7	Unlicense
memorystream	0.3.1	MIT
meow	8.1.2	MIT
merge-descriptors	1.0.1	MIT
merge-stream	2.0.0	MIT
merge2	1.4.1	MIT
methods	1.1.2	MIT
micromatch	4.0.5	MIT

Package	Version	License
miller-rabin	4.0.1	MIT
mime-db	1.52.0	MIT
mime-types	2.1.35	MIT
mime	1.6.0	MIT
mime	2.5.2	MIT
mime	2.6.0	MIT
mimic-fn	2.1.0	MIT
mimic-response	1.0.1	MIT
min-indent	1.0.1	MIT
mini-css-extract-plugin	2.5.3	MIT
minimalistic-assert	1.0.1	ISC
minimalistic-crypto-utils	1.0.1	MIT
minimatch	3.0.4	ISC
minimatch	3.0.5	ISC
minimatch	3.1.2	ISC
minimatch	5.1.0	ISC
minimist-options	4.1.0	MIT
minimist	1.2.6	MIT
minipass-collect	1.0.2	ISC
minipass-fetch	1.4.1	MIT
minipass-fetch	2.1.0	MIT
minipass-flush	1.0.5	ISC
minipass-json-stream	1.0.1	MIT

Package	Version	License
minipass-pipeline	1.2.4	ISC
minipass-sized	1.0.3	ISC
minipass	3.3.4	ISC
minizlib	2.1.2	MIT
minstache	1.2.0	MIT*
mkdirp	0.5.6	MIT
mkdirp	1.0.4	MIT
modern-normalize	1.1.0	MIT
modify-values	1.0.1	MIT
moment	2.29.4	MIT
ms	2.0.0	MIT
ms	2.1.2	MIT
ms	2.1.3	MIT
multicast-dns-service-types	1.1.0	MIT
multicast-dns	6.2.3	MIT
mute-stream	0.0.8	ISC
nan	2.16.0	MIT
nanoid	3.3.4	MIT
natural-compare-lite	1.4.0	MIT
natural-compare	1.4.0	MIT
needle	2.9.1	MIT
negotiator	0.6.3	MIT
neo-async	2.6.2	MIT

Package	Version	License
ng-packagr	13.3.1	MIT
ng-samurai	3.0.0	MIT
ng2-mock-component	0.2.0	MIT
ngx-quill	16.2.1	MIT
ngx-virtual-scroller	4.0.3	MIT
nice-napi	1.0.2	MIT
nice-try	1.0.5	MIT
node-addon-api	3.2.1	MIT
node-fetch	2.6.7	MIT
node-forge	1.3.1	(BSD-3-Clause OR GPL-2.0)
node-gyp-build	4.5.0	MIT
node-gyp	8.4.1	MIT
node-polyfill-webpack-plugin	1.1.4	MIT
node-releases	2.0.6	MIT
nopt	5.0.0	ISC
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-range	0.1.2	MIT
normalize-url	4.5.1	MIT
npm-bundled	1.1.2	ISC
npm-conf	1.1.3	MIT
npm-install-checks	4.0.0	BSD-2-Clause

Package	Version	License
npm-normalize-package-bin	1.0.1	ISC
npm-package-arg	8.1.5	ISC
npm-packlist	3.0.0	ISC
npm-pick-manifest	6.1.1	ISC
npm-registry-fetch	12.0.2	ISC
npm-run-all	4.1.5	MIT
npm-run-path	4.0.1	MIT
npmlog	5.0.1	ISC
npmlog	6.0.2	ISC
nth-check	2.1.1	BSD-2-Clause
nx	13.1.3	MIT
nx	14.5.2	MIT
oauth-sign	0.9.0	Apache-2.0
object-assign	4.1.1	MIT
object-inspect	1.12.2	MIT
object-is	1.1.5	MIT
object-keys	1.1.1	MIT
object.assign	4.1.2	MIT
obuf	1.1.2	MIT
on-finished	2.3.0	MIT
on-finished	2.4.1	MIT
on-headers	1.0.2	MIT
once	1.4.0	ISC

Package	Version	License
onetime	5.1.2	MIT
open	8.4.0	MIT
optionator	0.9.1	MIT
ora	5.4.1	MIT
os-browserify	0.3.0	MIT
os-tmpdir	1.0.2	MIT
p-cancelable	1.1.0	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-retry	4.6.2	MIT
p-try	1.0.0	MIT
p-try	2.2.0	MIT
package-json	6.5.0	MIT
pacote	12.0.3	ISC
pako	1.0.11	(MIT AND Zlib)
parchment	1.1.4	BSD-3-Clause
parent-module	1.0.1	MIT
parse-asn1	5.1.6	ISC

Package	Version	License
parse-json	4.0.0	MIT
parse-json	5.2.0	MIT
parse-node-version	1.0.1	MIT
parse5-html-rewriting-stream	6.0.1	MIT
parse5-htmlparser2-tree-adapter	6.0.1	MIT
parse5-sax-parser	6.0.1	MIT
parse5	5.1.1	MIT
parse5	6.0.1	MIT
parseurl	1.3.3	MIT
path-browserify	1.0.1	MIT
path-exists	3.0.0	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	2.0.1	MIT
path-key	3.1.1	MIT
path-parse	1.0.7	MIT
path-to-regexp	0.1.7	MIT
path-type	3.0.0	MIT
path-type	4.0.0	MIT
pbkdf2	3.1.2	MIT
pend	1.2.0	MIT
performance-now	2.1.0	MIT

Package	Version	License
picocolors	1.0.0	ISC
picomatch	2.3.1	MIT
pidtree	0.3.1	MIT
pify	2.3.0	MIT
pify	3.0.0	MIT
pify	4.0.1	MIT
pinkie-promise	2.0.1	MIT
pinkie	2.0.4	MIT
piscina	3.2.0	MIT
pkg-dir	4.2.0	MIT
playwright-core	1.23.1	Apache-2.0
playwright	1.23.1	Apache-2.0
plist	3.0.6	MIT
portfinder	1.0.28	MIT
postcss-attribute-case-insensitive	5.0.2	MIT
postcss-color-functional-notation	4.2.4	CC0-1.0
postcss-color-hex-alpha	8.0.4	MIT
postcss-color-rebeccapurple	7.1.1	CC0-1.0
postcss-custom-media	8.0.2	MIT
postcss-custom-properties	12.1.8	MIT
postcss-custom-selectors	6.0.3	MIT
postcss-dir-pseudo-class	6.0.5	CC0-1.0
postcss-double-position-gradients	3.1.2	CC0-1.0

Package	Version	License
postcss-env-function	4.0.6	CC0-1.0
postcss-focus-visible	6.0.4	CC0-1.0
postcss-focus-within	5.0.4	CC0-1.0
postcss-font-variant	5.0.0	MIT
postcss-gap-properties	3.0.5	CC0-1.0
postcss-image-set-function	4.0.7	CC0-1.0
postcss-import	14.0.2	MIT
postcss-initial	4.0.1	MIT
postcss-lab-function	4.2.1	CC0-1.0
postcss-loader	6.2.1	MIT
postcss-logical	5.0.4	CC0-1.0
postcss-media-minmax	5.0.0	MIT
postcss-modules-extract-imports	3.0.0	ISC
postcss-modules-local-by-default	4.0.0	MIT
postcss-modules-scope	3.0.0	ISC
postcss-modules-values	4.0.0	ISC
postcss-nesting	10.1.10	CC0-1.0
postcss-overflow-shorthand	3.0.4	CC0-1.0
postcss-page-break	3.0.4	MIT
postcss-place	7.0.5	CC0-1.0
postcss-preset-env	7.2.3	CC0-1.0
postcss-pseudo-class-any-link	7.1.6	CC0-1.0
postcss-replace-overflow-wrap	4.0.0	MIT

Package	Version	License
postcss-selector-not	5.0.0	MIT
postcss-selector-parser	6.0.10	MIT
postcss-url	10.1.3	MIT
postcss-value-parser	4.2.0	MIT
postcss	8.4.5	MIT
prelude-ls	1.2.1	MIT
prepend-http	2.0.0	MIT
pretty-bytes	5.6.0	MIT
process-nextick-args	2.0.1	MIT
process	0.11.10	MIT
progress	2.0.3	MIT
promise-inflight	1.0.1	ISC
promise-retry	2.0.1	MIT
prompts	2.4.2	MIT
proto-list	1.2.4	ISC
protobufjs	6.11.3	BSD-3-Clause
proxy-addr	2.0.7	MIT
prr	1.0.1	MIT
psl	1.9.0	MIT
public-encrypt	4.0.3	MIT
pump	3.0.0	MIT
punycode	1.3.2	MIT
punycode	2.1.1	MIT

Package	Version	License
pupa	2.1.1	MIT
q	1.5.1	MIT
qjobs	1.2.0	MIT
qs	6.10.3	BSD-3-Clause
qs	6.5.3	BSD-3-Clause
querystring-es3	0.2.1	MIT
querystring	0.2.0	MIT
queue-microtask	1.2.3	MIT
quick-lru	4.0.1	MIT
quill-delta	3.6.3	MIT
quill	1.3.7	BSD-3-Clause
ramda	0.28.0	MIT
randombytes	2.1.0	MIT
randomfill	1.0.4	MIT
range-parser	1.2.1	MIT
raw-body	2.5.1	MIT
rc	1.2.8	(BSD-2-Clause OR MIT OR Apache-2.0)
read-cache	1.0.0	MIT
read-config-file	6.2.0	MIT
read-package-json-fast	2.0.3	ISC
read-pkg-up	3.0.0	MIT
read-pkg-up	7.0.1	MIT

Package	Version	License
read-pkg	3.0.0	MIT
read-pkg	5.2.0	MIT
readable-stream	2.3.7	MIT
readable-stream	3.6.0	MIT
readdirp	3.6.0	MIT
redent	3.0.0	MIT
reflect-metadata	0.1.13	Apache-2.0
regenerate-unicode-properties	10.0.1	MIT
regenerate	1.4.2	MIT
regenerator-runtime	0.13.9	MIT
regenerator-transform	0.15.0	MIT
regex-parser	2.2.11	MIT
regexp.prototype.flags	1.4.3	MIT
regexpp	3.2.0	MIT
regexpu-core	5.1.0	MIT
registry-auth-token	4.2.2	MIT
registry-url	5.1.0	MIT
regjsgen	0.6.0	MIT
regjsparser	0.8.4	BSD-2-Clause
request	2.88.2	Apache-2.0
require-directory	2.1.1	MIT
require-from-string	2.0.2	MIT
require-main-filename	2.0.0	ISC

Package	Version	License
requires-port	1.0.0	MIT
resolve-from	4.0.0	MIT
resolve-from	5.0.0	MIT
resolve-url-loader	5.0.0	MIT
resolve	1.22.0	MIT
responselike	1.0.2	MIT
restore-cursor	3.1.0	MIT
retry	0.12.0	MIT
retry	0.13.1	MIT
reusify	1.0.4	MIT
rfdc	1.3.0	MIT
rimraf	2.7.1	ISC
rimraf	3.0.2	ISC
ripemd160	2.0.2	MIT
roarr	2.15.4	BSD-3-Clause
rollup-plugin-sourcemaps	0.6.3	MIT
rollup	2.77.0	MIT
run-async	2.4.1	MIT
run-parallel	1.2.0	MIT
rxjs-compat	6.6.7	Apache-2.0
rxjs-for-await	0.0.2	MIT
rxjs-report-usage	1.0.6	MIT
rxjs-tslint-rules	4.34.8	MIT

Package	Version	License
rxjs-tslint	0.1.8	MIT
rxjs	6.6.7	Apache-2.0
rxjs	7.5.6	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
sass-loader	13.0.2	MIT
sass	1.49.9	MIT
sax	1.2.4	ISC
schema-utils	2.7.1	MIT
schema-utils	3.1.1	MIT
schema-utils	4.0.0	MIT
select-hose	2.0.0	MIT
selfsigned	2.0.1	MIT
semver-compare	1.0.0	MIT
semver-diff	3.1.1	MIT
semver-dsl	1.0.1	MIT
semver	5.7.1	ISC
semver	6.3.0	ISC
semver	7.0.0	ISC
semver	7.3.4	ISC
semver	7.3.5	ISC

Package	Version	License
semver	7.3.7	ISC
semver	7.3.8	ISC
send	0.18.0	MIT
serialize-error	7.0.1	MIT
serialize-javascript	6.0.0	BSD-3-Clause
serve-index	1.9.1	MIT
serve-static	1.15.0	MIT
set-blocking	2.0.0	ISC
setimmediate	1.0.5	MIT
setprototypeof	1.1.0	ISC
setprototypeof	1.2.0	ISC
sha.js	2.4.11	(MIT AND BSD-3-Clause)
shallow-clone	3.0.1	MIT
shebang-command	1.2.0	MIT
shebang-command	2.0.0	MIT
shebang-regex	1.0.0	MIT
shebang-regex	3.0.0	MIT
shell-quote	1.7.3	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

Package	Version	License
smart-buffer	4.2.0	MIT
socket.io-adapter	2.4.0	MIT
socket.io-parser	4.0.5	MIT
socket.io	4.5.1	MIT
sockjs	0.3.24	MIT
socks-proxy-agent	6.2.1	MIT
socks-proxy-agent	7.0.0	MIT
socks	2.7.0	MIT
source-map-js	1.0.2	BSD-3-Clause
source-map-loader	3.0.1	MIT
source-map-resolve	0.6.0	MIT
source-map-support	0.5.21	MIT
source-map	0.5.7	BSD-3-Clause
source-map	0.6.1	BSD-3-Clause
source-map	0.7.3	BSD-3-Clause
sourcemap-codec	1.4.8	MIT
spdx-correct	3.1.1	Apache-2.0
spdx-exceptions	2.3.0	CC-BY-3.0
spdx-expression-parse	3.0.1	MIT
spdx-license-ids	3.0.11	CC0-1.0
spdy-transport	3.0.0	MIT
spdy	4.0.2	MIT
split2	3.2.2	ISC

Package	Version	License
split	1.0.1	MIT
sprintf-js	1.0.3	BSD-3-Clause
sprintf-js	1.1.2	BSD-3-Clause
sshpk	1.17.0	MIT
ssri	8.0.1	ISC
ssri	9.0.1	ISC
stat-mode	1.0.0	MIT
statuses	1.5.0	MIT
statuses	2.0.1	MIT
stream-browserify	3.0.0	MIT
stream-http	3.2.0	MIT
streamroller	3.1.1	MIT
string-width	4.2.3	MIT
string.prototype.padend	3.1.3	MIT
string.prototype.trimend	1.0.5	MIT
string.prototype.trimstart	1.0.5	MIT
string_decoder	1.1.1	MIT
string_decoder	1.3.0	MIT
strip-ansi	3.0.1	MIT
strip-ansi	6.0.1	MIT
strip-ansi	7.0.1	MIT
strip-bom	3.0.0	MIT
strip-final-newline	2.0.0	MIT

Package	Version	License
strip-indent	3.0.0	MIT
strip-json-comments	2.0.1	MIT
strip-json-comments	3.1.1	MIT
stylus-loader	6.2.0	MIT
stylus	0.56.0	MIT
sumchecker	3.0.1	Apache-2.0
supports-color	2.0.0	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-observable	4.0.0	MIT
tapable	2.2.1	MIT
tar-stream	2.2.0	MIT
tar	6.1.11	ISC
temp-dir	2.0.0	MIT
temp-file	3.4.0	MIT
tempfile	3.0.0	MIT
term-size	2.2.1	MIT
terser-webpack-plugin	5.3.3	MIT
terser	5.11.0	BSD-2-Clause
test-exclude	6.0.0	ISC
text-extensions	1.9.0	MIT

Package	Version	License
text-table	0.2.0	MIT
through2	2.0.5	MIT
through2	4.0.2	MIT
through	2.3.8	MIT
thunky	1.1.0	MIT
timers-browserify	2.0.12	MIT
tmp-promise	3.0.3	MIT
tmp	0.0.33	MIT
tmp	0.2.1	MIT
to-fast-properties	2.0.0	MIT
to-readable-stream	1.0.0	MIT
to-regex-range	5.0.1	MIT
toidentifier	1.0.1	MIT
tough-cookie	2.5.0	BSD-3-Clause
tr46	0.0.3	MIT
tree-kill	1.2.2	MIT
trim-newlines	3.0.1	MIT
truncate-utf8-bytes	1.0.2	WTFPL
ts-node	10.8.2	MIT
tsconfig-paths	3.14.1	MIT
tslib	1.14.1	0BSD
tslib	2.3.1	0BSD
tslib	2.4.0	0BSD

Package	Version	License
tslint	5.20.1	Apache-2.0
tslint	6.1.3	Apache-2.0
tsutils-etc	1.4.1	MIT
tsutils	2.29.0	MIT
tsutils	3.21.0	MIT
tty-browserify	0.0.1	MIT
tunnel-agent	0.6.0	Apache-2.0
tunnel	0.0.6	MIT
tweetnacl	0.14.5	Unlicense
type-check	0.4.0	MIT
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	0.21.3	(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-is	1.6.18	MIT
typed-assert	1.0.9	MIT
typed-emitter	2.1.0	MIT
typedarray-to-buffer	3.1.5	MIT
typedarray	0.0.6	MIT
typescript	4.6.4	Apache-2.0
ua-parser-js	0.7.31	MIT

Package	Version	License
uglify-js	3.16.2	BSD-2-Clause
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	2.0.0	MIT
unicode-property-aliases-ecmascript	2.0.0	MIT
unique-filename	1.1.1	ISC
unique-slug	2.0.2	ISC
unique-string	2.0.0	MIT
universalify	0.1.2	MIT
universalify	2.0.0	MIT
unpipe	1.0.0	MIT
update-browserslist-db	1.0.5	MIT
update-notifier	5.1.0	BSD-2-Clause
uri-js	4.4.1	BSD-2-Clause
url-parse-lax	3.0.0	MIT
url	0.11.0	MIT
utf8-byte-length	1.0.4	WTFPL
util-deprecate	1.0.2	MIT
util	0.12.4	MIT
utils-merge	1.0.1	MIT
uuid	3.4.0	MIT
uuid	8.3.2	MIT

Package	Version	License
v8-compile-cache-lib	3.0.1	MIT
v8-compile-cache	2.3.0	MIT
validate-npm-package-license	3.0.4	Apache-2.0
validate-npm-package-name	3.0.0	ISC
vary	1.1.2	MIT
verror	1.10.0	MIT
vm-browserify	1.1.2	MIT
void-elements	2.0.1	MIT
wait-on	6.0.1	MIT
watchpack	2.4.0	MIT
wav-encoder	1.3.0	MIT
wbuf	1.7.3	MIT
wcwidth	1.0.1	MIT
webdriver-manager	12.1.8	MIT
webidl-conversions	3.0.1	BSD-2-Clause
webpack-dev-middleware	5.3.0	MIT
webpack-dev-server	4.7.3	MIT
webpack-merge	5.8.0	MIT
webpack-sources	3.2.3	MIT
webpack-subresource-integrity	5.1.0	MIT
webpack	5.70.0	MIT
webpack	5.73.0	MIT
websocket-driver	0.7.4	Apache-2.0

Package	Version	License
websocket-extensions	0.1.4	Apache-2.0
whatwg-url	5.0.0	MIT
which-boxed-primitive	1.0.2	MIT
which-module	2.0.0	ISC
which-typed-array	1.1.8	MIT
which	1.3.1	ISC
which	2.0.2	ISC
wide-align	1.1.5	ISC
widest-line	3.1.0	MIT
wildcard	2.0.0	MIT
word-wrap	1.2.3	MIT
wordwrap	1.0.0	MIT
wrap-ansi	6.2.0	MIT
wrap-ansi	7.0.0	MIT
wrappy	1.0.2	ISC
write-file-atomic	3.0.3	ISC
ws	8.2.3	MIT
ws	8.8.0	MIT
xdg-basedir	4.0.0	MIT
xml2js	0.4.23	MIT
xmlbuilder	11.0.1	MIT
xmlbuilder	15.1.1	MIT
xtend	4.0.2	MIT

Package	Version	License
xxhashjs	0.2.2	MIT
y18n	4.0.3	ISC
y18n	5.0.8	ISC
yallist	4.0.0	ISC
yaml	1.10.2	ISC
yargs-parser	18.1.3	ISC
yargs-parser	20.0.0	ISC
yargs-parser	20.2.9	ISC
yargs-parser	21.0.1	ISC
yargs	15.4.1	MIT
yargs	16.2.0	MIT
yargs	17.5.1	MIT
yauzl	2.10.0	MIT
yn	3.1.1	MIT
yocto-queue	0.1.0	MIT
zone.js	0.10.3	MIT
zone.js	0.11.6	MIT
@types/node	18.0.6	MIT
@types/semver	7.3.10	MIT
axios	0.27.2	MIT
electron-is-accelerator	0.1.2	MIT
electron-json-storage	4.5.0	MIT
electron-localshortcut	3.2.1	MIT

Package	Version	License
glob	7.2.3	ISC
google-protobuf	3.20.1	BSD-3-Clause
lockfile	1.0.4	ISC
write-file-atomic	2.4.3	ISC

Photoshop

Package	Version	License
@angular/animations	4.4.7	MIT
@angular/cli	1.2.0	MIT
@angular/common	4.4.7	MIT
@angular/compiler-cli	4.4.7	MIT
@angular/compiler	4.4.7	MIT
@angular/core	4.4.7	MIT
@angular/forms	4.4.7	MIT
@angular/http	4.4.7	MIT
@angular/language-service	4.4.7	MIT
@angular/platform-browser-dynamic	4.4.7	MIT
@angular/platform-browser	4.4.7	MIT
@angular/router	4.4.7	MIT
@angular/tsc-wrapped	4.4.7	MIT
@ngtools/json-schema	1.1.0	MIT

Package	Version	License
@ngtools/webpack	1.5.0	MIT
@types/jasmine	2.5.54	MIT
@types/jasminewd2	2.0.8	MIT
@types/node	6.0.118	MIT
@types/q	0.0.32	MIT
@types/selenium-webdriver	2.53.45	MIT
abbrev	1.1.1	ISC
accepts	1.3.3	MIT
accepts	1.3.7	MIT
acorn-dynamic-import	2.0.2	MIT
acorn	4.0.13	MIT
acorn	5.7.4	MIT
adm-zip	0.4.16	MIT
adm-zip	0.4.4	MIT
after	0.8.2	MIT
agent-base	2.1.1	MIT
ajv-keywords	1.5.1	MIT
ajv	4.11.8	MIT
ajv	5.5.2	MIT
ajv	6.12.6	MIT
align-text	0.1.4	MIT
alphanum-sort	1.0.2	MIT
amdefine	1.0.1	BSD-3-Clause OR MIT

Package	Version	License
ansi-escapes	3.2.0	MIT
ansi-html	0.0.7	Apache-2.0
ansi-regex	2.1.1	MIT
ansi-regex	3.0.0	MIT
ansi-regex	4.1.0	MIT
ansi-styles	2.2.1	MIT
ansi-styles	3.2.1	MIT
anymatch	1.3.2	ISC
anymatch	2.0.0	ISC
anymatch	3.1.1	ISC
app-root-path	2.2.1	MIT
append-transform	0.4.0	MIT
aproba	1.2.0	ISC
are-we-there-yet	1.1.5	ISC
argparse	1.0.10	MIT
arr-diff	2.0.0	MIT
arr-diff	4.0.0	MIT
arr-flatten	1.1.0	MIT
arr-union	3.1.0	MIT
array-find-index	1.0.2	MIT
array-flatten	1.1.1	MIT
array-slice	0.2.3	MIT
array-union	1.0.2	MIT

Package	Version	License
array-uniq	1.0.3	MIT
array-unique	0.2.1	MIT
array-unique	0.3.2	MIT
arraybuffer.slice	0.0.6	MIT*
arraybuffer.slice	0.0.7	MIT
arrify	1.0.1	MIT
asap	2.0.6	MIT
asn1.js	5.4.1	MIT
asn1	0.2.4	MIT
assert-plus	0.2.0	MIT
assert-plus	1.0.0	MIT
assert	1.5.0	MIT
assign-symbols	1.0.0	MIT
async-each	1.0.3	MIT
async-foreach	0.1.3	MIT*
async-limiter	1.0.1	MIT
async	0.2.10	MIT
async	2.6.3	MIT
asynckit	0.4.0	MIT
atob	2.1.2	(MIT OR Apache-2.0)
autoprefixer	6.7.7	MIT
aws-sign2	0.6.0	Apache-2.0
aws-sign2	0.7.0	Apache-2.0

Package	Version	License
aws4	1.11.0	MIT
babel-code-frame	6.26.0	MIT
babel-generator	6.26.1	MIT
babel-messages	6.23.0	MIT
babel-runtime	6.26.0	MIT
babel-template	6.26.0	MIT
babel-traverse	6.26.0	MIT
babel-types	6.26.0	MIT
babylon	6.18.0	MIT
backo2	1.0.2	MIT
balanced-match	0.4.2	MIT
balanced-match	1.0.0	MIT
base64-arraybuffer	0.1.4	MIT
base64-arraybuffer	0.1.5	MIT
base64-js	1.5.1	MIT
base64id	1.0.0	MIT
base64id	2.0.0	MIT
base	0.11.2	MIT
batch	0.6.1	MIT
bcrypt-pbkdf	1.0.2	BSD-3-Clause
better-assert	1.0.2	MIT*
big.js	3.2.0	MIT
big.js	5.2.2	MIT

Package	Version	License
binary-extensions	1.13.1	MIT
binary-extensions	2.2.0	MIT
bindings	1.5.0	MIT
blob	0.0.4	MIT*
blob	0.0.5	MIT
block-stream	0.0.9	ISC
blocking-proxy	0.0.5	MIT
bluebird	3.7.2	MIT
bn.js	4.12.0	MIT
bn.js	5.2.0	MIT
body-parser	1.19.0	MIT
boolbase	1.0.0	ISC
boom	2.10.1	BSD-3-Clause
brace-expansion	1.1.11	MIT
braces	0.1.5	MIT
braces	1.8.5	MIT
braces	2.3.2	MIT
braces	3.0.2	MIT
brorand	1.1.0	MIT
browserify-aes	1.2.0	MIT
browserify-cipher	1.0.1	MIT
browserify-des	1.0.2	MIT
browserify-rsa	4.1.0	MIT

Package	Version	License
browserify-sign	4.2.1	ISC
browserify-zlib	0.2.0	MIT
browserslist	1.7.7	MIT
buffer-alloc-unsafe	1.1.0	MIT
buffer-alloc	1.2.0	MIT
buffer-fill	1.0.0	MIT
buffer-xor	1.0.3	MIT
buffer	4.9.2	MIT
builtin-status-codes	3.0.0	MIT
bytes	3.0.0	MIT
bytes	3.1.0	MIT
cache-base	1.0.1	MIT
callsite	1.0.0	MIT*
camel-case	3.0.0	MIT
camelcase-keys	2.1.0	MIT
camelcase	1.2.1	MIT
camelcase	2.1.1	MIT
camelcase	3.0.0	MIT
camelcase	5.3.1	MIT
caniuse-api	1.6.1	MIT
caniuse-db	1.0.30001204	CC-BY-4.0
caseless	0.12.0	Apache-2.0
center-align	0.1.3	MIT

Package	Version	License
chalk	1.1.3	MIT
chalk	2.4.2	MIT
chardet	0.4.2	MIT
chokidar	1.7.0	MIT
chokidar	2.1.8	MIT
chokidar	3.5.1	MIT
cipher-base	1.0.4	MIT
clap	1.2.3	MIT
class-utils	0.3.6	MIT
clean-css	4.2.3	MIT
cli-cursor	2.1.0	MIT
cli-width	2.2.1	ISC
cliui	2.1.0	ISC
cliui	3.2.0	ISC
cliui	5.0.0	ISC
clone-deep	2.0.2	MIT
clone	1.0.4	MIT
clone	2.1.2	MIT
co	4.6.0	MIT
coa	1.0.4	MIT
code-point-at	1.1.0	MIT
codelyzer	3.0.1	MIT
collection-visit	1.0.0	MIT

Package	Version	License
color-convert	1.9.3	MIT
color-name	1.1.3	MIT
color-string	0.3.0	MIT
color	0.11.4	MIT
colormin	1.1.2	MIT
colors	1.1.2	MIT
combine-lists	1.0.1	MIT
combined-stream	1.0.8	MIT
commander	2.17.1	MIT
commander	2.19.0	MIT
common-tags	1.8.0	MIT
component-bind	1.0.0	MIT*
component-emitter	1.1.2	MIT*
component-emitter	1.2.1	MIT
component-emitter	1.3.0	MIT
component-inherit	0.0.3	MIT*
compressible	2.0.18	MIT
compression	1.7.4	MIT
concat-map	0.0.1	MIT
connect-history-api-fallback	1.6.0	MIT
connect	3.7.0	MIT
console-browserify	1.2.0	MIT
console-control-strings	1.1.0	ISC

Package	Version	License
constants-browserify	1.0.0	MIT
content-disposition	0.5.3	MIT
content-type	1.0.4	MIT
convert-source-map	1.7.0	MIT
cookie-signature	1.0.6	MIT
cookie	0.3.1	MIT
cookie	0.4.0	MIT
cookie	0.4.1	MIT
copy-descriptor	0.1.1	MIT
core-js	2.6.12	MIT
core-object	3.1.5	MIT
core-util-is	1.0.2	MIT
cosmiconfig	2.2.2	MIT
create-ecdh	4.0.4	MIT
create-hash	1.2.0	MIT
create-hmac	1.1.7	MIT
cross-spawn	3.0.1	MIT
cryptiles	2.0.5	BSD-3-Clause
crypto-browserify	3.12.0	MIT
csinterface-ts	1.0.3	MIT
css-color-names	0.0.4	MIT
css-loader	0.28.11	MIT
css-parse	2.0.0	MIT

Package	Version	License
css-select	2.1.0	BSD-2-Clause
css-selector-tokenizer	0.7.3	MIT
css-what	3.4.2	BSD-2-Clause
css	2.2.4	MIT
cssauron	1.4.0	MIT
cssesc	3.0.0	MIT
cssnano	3.10.0	MIT
csso	2.3.2	MIT
currently-unhandled	0.4.1	MIT
custom-event	1.0.1	MIT
dashdash	1.14.1	MIT
debug	2.2.0	MIT
debug	2.3.3	MIT
debug	2.6.9	MIT
debug	3.1.0	MIT
debug	3.2.7	MIT
debug	4.1.1	MIT
decamelize	1.2.0	MIT
decode-uri-component	0.2.0	MIT
default-require-extensions	1.0.0	MIT
define-property	0.2.5	MIT
define-property	1.0.0	MIT
define-property	2.0.2	MIT

Package	Version	License
defined	1.0.0	MIT
del	2.2.2	MIT
delayed-stream	1.0.0	MIT
delegates	1.0.0	MIT
denodeify	1.2.1	MIT
depd	1.1.2	MIT
des.js	1.0.1	MIT
destroy	1.0.4	MIT
detect-indent	4.0.0	MIT
detect-node	2.0.5	MIT
di	0.0.1	MIT
diff	3.5.0	BSD-3-Clause
diffie-hellman	5.0.3	MIT
directory-encoder	0.7.2	MIT
dom-converter	0.2.0	MIT
dom-serialize	2.2.1	MIT
dom-serializer	0.2.2	MIT
domain-browser	1.2.0	MIT
domelementtype	1.3.1	BSD-2-Clause
domelementtype	2.1.0	BSD-2-Clause
domhandler	2.4.2	BSD-2-Clause
domutils	1.7.0	BSD-2-Clause
ecc-jsbn	0.1.2	MIT

Package	Version	License
ee-first	1.1.1	MIT
electron-to-chromium	1.3.698	ISC
elliptic	6.5.4	MIT
ember-cli-normalize-entity-name	1.0.0	ISC
ember-cli-string-utils	1.1.0	ISC
emoji-regex	7.0.3	MIT
emojis-list	2.1.0	MIT
emojis-list	3.0.0	MIT
encodeurl	1.0.2	MIT
engine.io-client	1.8.3	MIT
engine.io-client	3.5.1	MIT
engine.io-parser	1.3.2	MIT
engine.io-parser	2.2.1	MIT
engine.io	1.8.3	MIT
engine.io	3.5.0	MIT
enhanced-resolve	3.4.1	MIT
ensure-posix-path	1.1.1	ISC
ent	2.2.0	MIT
entities	1.1.2	BSD-2-Clause
entities	2.2.0	BSD-2-Clause
errno	0.1.8	MIT
error-ex	1.3.2	MIT
escape-html	1.0.3	MIT

Package	Version	License
escape-string-regexp	1.0.5	MIT
esprima	2.7.3	BSD-2-Clause
esutils	2.0.3	BSD-2-Clause
etag	1.8.1	MIT
eventemitter3	4.0.7	MIT
events	3.3.0	MIT
eventsource	0.1.6	MIT
evp_bytestokey	1.0.3	MIT
exit	0.1.2	MIT
expand-braces	0.1.2	MIT
expand-brackets	0.1.5	MIT
expand-brackets	2.1.4	MIT
expand-range	0.1.1	MIT
expand-range	1.8.2	MIT
exports-loader	0.6.4	MIT
express	4.17.1	MIT
extend-shallow	2.0.1	MIT
extend-shallow	3.0.2	MIT
extend	3.0.2	MIT
external-editor	2.2.0	MIT
extglob	0.3.2	MIT
extglob	2.0.4	MIT
extract-text-webpack-plugin	2.1.2	MIT

Package	Version	License
extsprintf	1.3.0	MIT
fast-deep-equal	1.1.0	MIT
fast-deep-equal	3.1.3	MIT
fast-json-stable-stringify	2.1.0	MIT
fastparse	1.1.2	MIT
faye-websocket	0.10.0	MIT
faye-websocket	0.11.3	Apache-2.0
figures	2.0.0	MIT
file-loader	0.10.1	MIT
file-uri-to-path	1.0.0	MIT
filename-regex	2.0.1	MIT
fileset	2.0.3	MIT
fill-range	2.2.4	MIT
fill-range	4.0.0	MIT
fill-range	7.0.1	MIT
finalhandler	1.1.2	MIT
find-up	1.1.2	MIT
find-up	3.0.0	MIT
flatten	1.0.3	MIT
flix-cep	6.6.0	UNLICENSED
follow-redirects	1.13.3	MIT
for-in	0.1.8	MIT
for-in	1.0.2	MIT

Package	Version	License
for-own	0.1.5	MIT
for-own	1.0.0	MIT
forever-agent	0.6.1	Apache-2.0
form-data	2.1.4	MIT
form-data	2.3.3	MIT
forwarded	0.1.2	MIT
fragment-cache	0.2.1	MIT
fresh	0.5.2	MIT
fs-access	1.0.1	MIT
fs-extra	0.23.1	MIT
fs-extra	3.0.1	MIT
fs.realpath	1.0.0	ISC
fs	0.0.1-security	ISC
fsevents	1.2.13	MIT
fsevents	2.3.2	MIT
fstream	1.0.12	ISC
function-bind	1.1.1	MIT
gauge	2.7.4	ISC
gaze	1.1.3	MIT
get-caller-file	1.0.3	ISC
get-caller-file	2.0.5	ISC
get-stdin	4.0.1	MIT
get-value	2.0.6	MIT

Package	Version	License
getpass	0.1.7	MIT
glob-base	0.3.0	MIT
glob-parent	2.0.0	ISC
glob-parent	3.1.0	ISC
glob-parent	5.1.2	ISC
glob	7.1.6	ISC
globals	9.18.0	MIT
globby	5.0.0	MIT
globule	1.3.2	MIT
graceful-fs	4.2.6	ISC
handle-thing	1.2.5	MIT
handlebars	1.3.0	MIT
handlebars	4.7.7	MIT
har-schema	1.0.5	ISC
har-schema	2.0.0	ISC
har-validator	4.2.1	ISC
har-validator	5.1.5	MIT
has-ansi	2.0.0	MIT
has-binary2	1.0.3	MIT
has-binary	0.1.7	MIT
has-cors	1.1.0	MIT
has-flag	1.0.0	MIT
has-flag	3.0.0	MIT

Package	Version	License
has-unicode	2.0.1	ISC
has-value	0.3.1	MIT
has-value	1.0.0	MIT
has-values	0.1.4	MIT
has-values	1.0.0	MIT
has	1.0.3	MIT
hash-base	3.1.0	MIT
hash.js	1.1.7	MIT
hawk	3.1.3	BSD-3-Clause
he	1.2.0	MIT
heimdalljs-logger	0.1.10	MIT
heimdalljs	0.2.6	MIT
hmac-drbg	1.0.1	MIT
hoek	2.16.3	BSD-3-Clause
hosted-git-info	2.8.8	ISC
hpack.js	2.1.6	MIT
html-comment-regex	1.1.2	MIT
html-entities	1.4.0	MIT
html-minifier	3.5.21	MIT
html-webpack-plugin	2.30.1	MIT
htmlparser2	3.10.1	MIT
http-deceiver	1.2.7	MIT
http-errors	1.6.3	MIT

Package	Version	License
http-errors	1.7.2	MIT
http-parser-js	0.5.3	MIT
http-proxy-middleware	0.17.4	MIT
http-proxy	1.18.1	MIT
http-signature	1.1.1	MIT
http-signature	1.2.0	MIT
https-browserify	1.0.0	MIT
https-proxy-agent	1.0.0	MIT
iconv-lite	0.4.24	MIT
icss-replace-symbols	1.1.0	ISC
icss-utils	2.1.0	ISC
ieee754	1.2.1	BSD-3-Clause
image-size	0.5.5	MIT
img-stats	0.5.2	MIT
in-publish	2.0.1	ISC
indent-string	2.1.0	MIT
indexes-of	1.0.1	MIT
indexof	0.0.1	MIT*
inflection	1.12.0	MIT
inflight	1.0.6	ISC
inherits	2.0.1	ISC
inherits	2.0.3	ISC
inherits	2.0.4	ISC

Package	Version	License
ini	1.3.8	ISC
inquirer	3.3.0	MIT
interpret	1.4.0	MIT
invariant	2.2.4	MIT
invert-kv	1.0.0	MIT
ipaddr.js	1.9.1	MIT
is-absolute-url	2.1.0	MIT
is-accessor-descriptor	0.1.6	MIT
is-accessor-descriptor	1.0.0	MIT
is-arrayish	0.2.1	MIT
is-binary-path	1.0.1	MIT
is-binary-path	2.1.0	MIT
is-buffer	1.1.6	MIT
is-core-module	2.2.0	MIT
is-data-descriptor	0.1.4	MIT
is-data-descriptor	1.0.0	MIT
is-descriptor	0.1.6	MIT
is-descriptor	1.0.2	MIT
is-directory	0.3.1	MIT
is-dotfile	1.0.3	MIT
is-equal-shallow	0.1.3	MIT
is-extendable	0.1.1	MIT
is-extendable	1.0.1	MIT

Package	Version	License
is-extglob	1.0.0	MIT
is-extglob	2.1.1	MIT
is-finite	1.1.0	MIT
is-fullwidth-code-point	1.0.0	MIT
is-fullwidth-code-point	2.0.0	MIT
is-glob	2.0.1	MIT
is-glob	3.1.0	MIT
is-glob	4.0.1	MIT
is-number	0.1.1	MIT
is-number	2.1.0	MIT
is-number	3.0.0	MIT
is-number	4.0.0	MIT
is-number	7.0.0	MIT
is-path-cwd	1.0.0	MIT
is-path-in-cwd	1.0.1	MIT
is-path-inside	1.0.1	MIT
is-plain-obj	1.1.0	MIT
is-plain-object	2.0.4	MIT
is-posix-bracket	0.1.1	MIT
is-primitive	2.0.0	MIT
is-svg	2.1.0	MIT
is-typedarray	1.0.0	MIT
is-utf8	0.2.1	MIT

Package	Version	License
is-windows	1.0.2	MIT
isarray	0.0.1	MIT
isarray	1.0.0	MIT
isarray	2.0.1	MIT
isbinaryfile	3.0.3	MIT
isexe	2.0.0	ISC
isobject	2.1.0	MIT
isobject	3.0.1	MIT
isstream	0.1.2	MIT
istanbul-api	1.3.7	BSD-3-Clause
istanbul-instrumenter-loader	2.0.0	MIT
istanbul-lib-coverage	1.2.1	BSD-3-Clause
istanbul-lib-hook	1.2.2	BSD-3-Clause
istanbul-lib-instrument	1.10.2	BSD-3-Clause
istanbul-lib-report	1.1.5	BSD-3-Clause
istanbul-lib-source-maps	1.2.6	BSD-3-Clause
istanbul-reports	1.5.1	BSD-3-Clause
jasmine-core	2.6.4	MIT
jasmine-core	2.99.1	MIT
jasmine-spec-reporter	4.1.1	Apache-2.0
jasmine	2.99.0	MIT
jasminewd2	2.2.0	MIT
js-base64	2.6.4	BSD-3-Clause

Package	Version	License
js-tokens	3.0.2	MIT
js-yaml	3.7.0	MIT
jsbn	0.1.1	MIT
jsesc	1.3.0	MIT
json-loader	0.5.7	MIT
json-schema-traverse	0.3.1	MIT
json-schema-traverse	0.4.1	MIT
json-schema	0.2.3	['AFLv2.1', 'BSD']
json-stable-stringify	1.0.1	MIT
json-stringify-safe	5.0.1	ISC
json3	3.3.2	MIT
json3	3.3.3	MIT
json5	0.5.1	MIT
json5	1.0.1	MIT
jsonfile	2.4.0	MIT
jsonfile	3.0.1	MIT
jsonify	0.0.0	Public Domain
jsprim	1.4.1	MIT
karma-chrome-launcher	2.1.1	MIT
karma-cli	1.0.1	MIT
karma-coverage-istanbul-reporter	1.4.3	MIT
karma-jasmine-html-reporter	0.2.2	MIT
karma-jasmine	1.1.2	MIT

Package	Version	License
karma	1.7.1	MIT
kind-of	3.2.2	MIT
kind-of	4.0.0	MIT
kind-of	5.1.0	MIT
kind-of	6.0.3	MIT
lazy-cache	1.0.4	MIT
lcid	1.0.0	MIT
less-loader	4.1.0	MIT
less	2.7.3	Apache-2.0
license-webpack-plugin	0.4.3	ISC
load-json-file	1.1.0	MIT
loader-runner	2.4.0	MIT
loader-utils	0.2.17	MIT
loader-utils	1.4.0	MIT
locate-path	3.0.0	MIT
lodash.camelcase	4.3.0	MIT
lodash.clonedeep	4.5.0	MIT
lodash.memoize	4.1.2	MIT
lodash.tail	4.1.1	MIT
lodash.uniq	4.5.0	MIT
lodash	3.10.1	MIT
lodash	4.17.21	MIT
log4js	0.6.38	Apache-2.0

Package	Version	License
longest	1.0.1	MIT
loose-envify	1.4.0	MIT
loud-rejection	1.6.0	MIT
lower-case	1.1.4	MIT
lru-cache	4.1.5	ISC
magic-string	0.19.1	MIT
make-error	1.3.6	ISC
map-cache	0.2.2	MIT
map-obj	1.0.1	MIT
map-visit	1.0.0	MIT
matcher-collection	1.1.2	ISC
math-expression-evaluator	1.3.7	MIT
math-random	1.0.4	MIT
md5.js	1.3.5	MIT
media-typer	0.3.0	MIT
memory-fs	0.4.1	MIT
meow	3.7.0	MIT
merge-descriptors	1.0.1	MIT
methods	1.1.2	MIT
micromatch	2.3.11	MIT
micromatch	3.1.10	MIT
miller-rabin	4.0.1	MIT
mime-db	1.46.0	MIT

Package	Version	License
mime-types	2.1.29	MIT
mime	1.3.6	MIT
mime	1.6.0	MIT
mimic-fn	1.2.0	MIT
minimalistic-assert	1.0.1	ISC
minimalistic-crypto-utils	1.0.1	MIT
minimatch	3.0.4	ISC
minimist	0.0.10	MIT
minimist	1.2.5	MIT
mixin-deep	1.3.2	MIT
mixin-object	2.0.1	MIT
mkdirp	0.5.5	MIT
mkdirp	1.0.4	MIT
ms	0.7.1	MIT*
ms	0.7.2	MIT
ms	2.0.0	MIT
ms	2.1.1	MIT
ms	2.1.3	MIT
mute-stream	0.0.7	ISC
nan	2.14.2	MIT
nanomatch	1.2.13	MIT
negotiator	0.6.1	MIT
negotiator	0.6.2	MIT

Package	Version	License
neo-async	2.6.2	MIT
no-case	2.3.2	MIT
node-gyp	3.8.0	MIT
node-libs-browser	2.2.1	MIT
node-modules-path	1.0.2	ISC
node-sass	4.14.1	MIT
nopt	3.0.6	ISC
nopt	4.0.3	ISC
normalize-package-data	2.5.0	BSD-2-Clause
normalize-path	2.1.1	MIT
normalize-path	3.0.0	MIT
normalize-range	0.1.2	MIT
normalize-url	1.9.1	MIT
npmlog	4.1.2	ISC
nth-check	1.0.2	BSD-2-Clause
null-check	1.0.0	MIT
num2fraction	1.2.2	MIT
number-is-nan	1.0.1	MIT
oauth-sign	0.8.2	Apache-2.0
oauth-sign	0.9.0	Apache-2.0
object-assign	4.1.0	MIT
object-assign	4.1.1	MIT
object-component	0.0.3	MIT*

Package	Version	License
object-copy	0.1.0	MIT
object-visit	1.0.1	MIT
object.omit	2.0.1	MIT
object.pick	1.3.0	MIT
obuf	1.1.2	MIT
on-finished	2.3.0	MIT
on-headers	1.0.2	MIT
once	1.4.0	ISC
onetime	2.0.1	MIT
opn	4.0.2	MIT
optimist	0.3.7	MIT*
optimist	0.6.1	MIT*
options	0.0.6	MIT*
original	1.0.2	MIT
os-browserify	0.3.0	MIT
os-homedir	1.0.2	MIT
os-locale	1.4.0	MIT
os-tmpdir	1.0.2	MIT
osenv	0.1.5	ISC
p-limit	2.3.0	MIT
p-locate	3.0.0	MIT
p-try	2.2.0	MIT
pako	1.0.11	(MIT AND Zlib)

Package	Version	License
param-case	2.1.1	MIT
parse-asn1	5.1.6	ISC
parse-glob	3.0.4	MIT
parse-json	2.2.0	MIT
parsejson	0.0.3	MIT
parseqs	0.0.5	MIT
parseqs	0.0.6	MIT
parseuri	0.0.5	MIT
parseuri	0.0.6	MIT
parseurl	1.3.3	MIT
pascalcase	0.1.1	MIT
path-browserify	0.0.1	MIT
path-dirname	1.0.2	MIT
path-exists	2.1.0	MIT
path-exists	3.0.0	MIT
path-is-absolute	1.0.1	MIT
path-is-inside	1.0.2	(WTFPL OR MIT)
path-parse	1.0.6	MIT
path-to-regexp	0.1.7	MIT
path-type	1.1.0	MIT
path	0.12.7	MIT
pbkdf2	3.1.1	MIT
performance-now	0.2.0	MIT

Package	Version	License
performance-now	2.1.0	MIT
picomatch	2.2.2	MIT
pify	2.3.0	MIT
pify	3.0.0	MIT
pinkie-promise	2.0.1	MIT
pinkie	2.0.4	MIT
portfinder	1.0.28	MIT
posix-character-classes	0.1.1	MIT
postcss-calc	5.3.1	MIT
postcss-colormin	2.2.2	MIT
postcss-convert-values	2.6.1	MIT
postcss-discard-comments	2.0.4	MIT
postcss-discard-duplicates	2.1.0	MIT
postcss-discard-empty	2.1.0	MIT
postcss-discard-overridden	0.1.1	MIT
postcss-discard-unused	2.2.3	MIT
postcss-filter-plugins	2.0.3	MIT
postcss-load-config	1.2.0	MIT
postcss-load-options	1.2.0	MIT
postcss-load-plugins	2.3.0	MIT
postcss-loader	1.3.3	MIT
postcss-merge-idents	2.1.7	MIT
postcss-merge-longhand	2.0.2	MIT

Package	Version	License
postcss-merge-rules	2.1.2	MIT
postcss-message-helpers	2.0.0	MIT
postcss-minify-font-values	1.0.5	MIT
postcss-minify-gradients	1.0.5	MIT
postcss-minify-params	1.2.2	MIT
postcss-minify-selectors	2.1.1	MIT
postcss-modules-extract-imports	1.2.1	ISC
postcss-modules-local-by-default	1.2.0	MIT
postcss-modules-scope	1.1.0	ISC
postcss-modules-values	1.3.0	ISC
postcss-normalize-charset	1.1.1	MIT
postcss-normalize-url	3.0.8	MIT
postcss-ordered-values	2.2.3	MIT
postcss-reduce-idents	2.4.0	MIT
postcss-reduce-initial	1.0.1	MIT
postcss-reduce-transforms	1.0.4	MIT
postcss-selector-parser	2.2.3	MIT
postcss-svgo	2.1.6	MIT
postcss-unique-selectors	2.0.2	MIT
postcss-url	5.1.2	MIT
postcss-value-parser	3.3.1	MIT
postcss-zindex	2.2.0	MIT
postcss	5.2.18	MIT

Package	Version	License
postcss	6.0.23	MIT
prepend-http	1.0.4	MIT
preserve	0.2.0	MIT
pretty-error	2.1.2	MIT
process-nextick-args	2.0.1	MIT
process	0.11.10	MIT
promise	7.3.1	MIT
protractor	5.1.2	MIT
proxy-addr	2.0.6	MIT
prr	1.0.1	MIT
pseudomap	1.0.2	ISC
psl	1.8.0	MIT
public-encrypt	4.0.3	MIT
punycode	1.3.2	MIT
punycode	1.4.1	MIT
punycode	2.1.1	MIT
q	1.4.1	MIT
q	1.5.1	MIT
qjobs	1.2.0	MIT
qs	6.4.0	BSD-3-Clause
qs	6.5.2	BSD-3-Clause
qs	6.7.0	BSD-3-Clause
query-string	4.3.4	MIT

Package	Version	License
querystring-es3	0.2.1	MIT
querystring	0.2.0	MIT
querystringify	2.2.0	MIT
randomatic	3.1.1	MIT
randombytes	2.1.0	MIT
randomfill	1.0.4	MIT
range-parser	1.2.1	MIT
raw-body	2.4.0	MIT
raw-loader	0.5.1	MIT
read-pkg-up	1.0.1	MIT
read-pkg	1.1.0	MIT
readable-stream	1.0.34	MIT
readable-stream	2.3.7	MIT
readable-stream	3.6.0	MIT
readdirp	2.2.1	MIT
readdirp	3.5.0	MIT
redent	1.0.0	MIT
reduce-css-calc	1.3.0	MIT
reduce-function-call	1.0.3	MIT
reflect-metadata	0.1.13	Apache-2.0
regenerator-runtime	0.11.1	MIT
regex-cache	0.4.4	MIT
regex-not	1.0.2	MIT

Package	Version	License
relateurl	0.2.7	MIT
remove-trailing-separator	1.1.0	ISC
renderkid	2.0.5	MIT
repeat-element	1.1.3	MIT
repeat-string	0.2.2	MIT
repeat-string	1.6.1	MIT
repeating	2.0.1	MIT
request	2.81.0	Apache-2.0
request	2.88.2	Apache-2.0
require-directory	2.1.1	MIT
require-from-string	1.2.1	MIT
require-main-filename	1.0.1	ISC
require-main-filename	2.0.0	ISC
requires-port	1.0.0	MIT
resolve-url	0.2.1	MIT
resolve	1.20.0	MIT
restore-cursor	2.0.0	MIT
ret	0.1.15	MIT
right-align	0.1.3	MIT
rimraf	2.2.8	MIT
rimraf	2.7.1	ISC
ripemd160	2.0.2	MIT
rsvp	3.2.1	MIT

Package	Version	License
rsvp	3.6.2	MIT
run-async	2.4.1	MIT
rx-lite-aggregates	4.0.8	Apache License, Version 2.0
rx-lite	4.0.8	Apache License, Version 2.0
rxjs	5.5.12	Apache-2.0
safe-buffer	5.1.2	MIT
safe-buffer	5.2.1	MIT
safe-regex	1.1.0	MIT
safer-buffer	2.1.2	MIT
sass-graph	2.2.5	MIT
sass-loader	6.0.7	MIT
saucelabs	1.3.0	MIT*
sax	0.6.1	BSD*
sax	1.2.4	ISC
schema-utils	0.3.0	MIT
script-loader	0.7.2	MIT
scss-tokenizer	0.2.3	MIT
select-hose	2.0.0	MIT
selenium-webdriver	2.53.3	Apache-2.0
selenium-webdriver	3.0.1	Apache-2.0
semver-dsl	1.0.1	MIT
semver	4.3.6	ISC
semver	5.0.3	ISC

Package	Version	License
semver	5.3.0	ISC
semver	5.7.1	ISC
semver	6.3.0	ISC
send	0.17.1	MIT
serve-index	1.9.1	MIT
serve-static	1.14.1	MIT
set-blocking	2.0.0	ISC
set-value	2.0.1	MIT
setimmediate	1.0.5	MIT
setprototypeof	1.1.0	ISC
setprototypeof	1.1.1	ISC
sha.js	2.4.11	(MIT AND BSD-3-Clause)
shallow-clone	1.0.0	MIT
signal-exit	3.0.3	ISC
silent-error	1.1.1	ISC
snapdragon-node	2.1.1	MIT
snapdragon-util	3.0.1	MIT
snapdragon	0.8.2	MIT
sntp	1.0.9	BSD
socket.io-adapter	0.5.0	MIT
socket.io-adapter	1.1.2	MIT
socket.io-client	1.7.3	MIT
socket.io-client	2.4.0	MIT

Package	Version	License
socket.io-parser	2.3.1	MIT
socket.io-parser	3.3.2	MIT
socket.io-parser	3.4.1	MIT
socket.io	1.7.3	MIT
socket.io	2.4.1	MIT
sockjs-client	1.1.2	MIT
sockjs	0.3.18	MIT
sort-keys	1.1.2	MIT
source-list-map	1.1.2	MIT
source-list-map	2.0.1	MIT
source-map-loader	0.2.4	MIT
source-map-resolve	0.5.3	MIT
source-map-support	0.4.18	MIT
source-map-url	0.4.1	MIT
source-map	0.1.43	BSD
source-map	0.4.4	BSD-3-Clause
source-map	0.5.7	BSD-3-Clause
source-map	0.6.1	BSD-3-Clause
source-map	0.7.3	BSD-3-Clause
spdx-correct	3.1.1	Apache-2.0
spdx-exceptions	2.3.0	CC-BY-3.0
spdx-expression-parse	3.0.1	MIT
spdx-license-ids	3.0.7	CC0-1.0

Package	Version	License
spdy-transport	2.1.1	MIT
spdy	3.4.7	MIT
split-string	3.1.0	MIT
sprintf-js	1.0.3	BSD-3-Clause
sshpk	1.16.1	MIT
static-extend	0.1.2	MIT
statuses	1.5.0	MIT
stdout-stream	1.4.1	MIT
stream-browserify	2.0.2	MIT
stream-http	2.8.3	MIT
strict-uri-encode	1.1.0	MIT
string-width	1.0.2	MIT
string-width	2.1.1	MIT
string-width	3.1.0	MIT
string_decoder	0.10.31	MIT
string_decoder	1.1.1	MIT
stringstream	0.0.6	MIT
strip-ansi	3.0.1	MIT
strip-ansi	4.0.0	MIT
strip-ansi	5.2.0	MIT
strip-bom	2.0.0	MIT
strip-bom	3.0.0	MIT
strip-indent	1.0.1	MIT

Package	Version	License
strip-json-comments	2.0.1	MIT
style-loader	0.13.2	MIT
stylus-loader	3.0.2	MIT
stylus	0.54.8	MIT
supports-color	2.0.0	MIT
supports-color	3.2.3	MIT
supports-color	5.5.0	MIT
svgo	0.7.2	MIT
symbol-observable	1.0.1	MIT
tapable	0.2.9	MIT
tar	2.2.2	ISC
temp	0.8.3	MIT
through	2.3.8	MIT
time-stamp	2.2.0	MIT
timers-browserify	2.0.12	MIT
tmp	0.0.24	MIT
tmp	0.0.30	MIT
tmp	0.0.31	MIT
tmp	0.0.33	MIT
to-array	0.1.4	MIT
to-arraybuffer	1.0.1	MIT
to-fast-properties	1.0.3	MIT
to-object-path	0.3.0	MIT

Package	Version	License
to-regex-range	2.1.1	MIT
to-regex-range	5.0.1	MIT
to-regex	3.0.2	MIT
toidentifier	1.0.0	MIT
toposort	1.0.7	MIT
tough-cookie	2.3.4	BSD-3-Clause
tough-cookie	2.5.0	BSD-3-Clause
trim-newlines	1.0.0	MIT
trim-right	1.0.1	MIT
true-case-path	1.0.3	Apache-2.0
ts-node	3.0.6	MIT
tsconfig	6.0.0	MIT
tsickle	0.21.6	MIT
tslib	1.14.1	0BSD
tslint	5.3.2	Apache-2.0
tsutils	2.29.0	MIT
tty-browserify	0.0.0	MIT
tunnel-agent	0.6.0	Apache-2.0
tweetnacl	0.14.5	Unlicense
type-is	1.6.18	MIT
typescript	2.3.4	Apache-2.0
uglify-js	2.3.6	BSD*
uglify-js	2.8.29	BSD-2-Clause

Package	Version	License
uglify-js	3.4.10	BSD-2-Clause
uglify-to-browserify	1.0.2	MIT
ultron	1.0.2	MIT
union-value	1.0.1	MIT
uniq	1.0.1	MIT
uniqs	2.0.0	MIT
universalify	0.1.2	MIT
unpipe	1.0.0	MIT
unset-value	1.0.0	MIT
upath	1.2.0	MIT
upper-case	1.1.3	MIT
uri-js	4.4.1	BSD-2-Clause
urix	0.1.0	MIT
url-loader	0.5.9	MIT
url-parse	1.5.1	MIT
url	0.11.0	MIT
use	3.1.1	MIT
user-home	1.1.1	MIT
useragent	2.3.0	MIT
util-deprecate	1.0.2	MIT
util	0.10.3	MIT
util	0.10.4	MIT
util	0.11.1	MIT

Package	Version	License
utila	0.4.0	MIT
utils-merge	1.0.1	MIT
uuid	2.0.3	MIT
uuid	3.4.0	MIT
v8flags	2.1.1	MIT
validate-npm-package-license	3.0.4	Apache-2.0
vary	1.1.2	MIT
vendors	1.0.4	MIT
verror	1.10.0	MIT
vlq	0.2.3	MIT
vm-browserify	1.1.2	MIT
void-elements	2.0.1	MIT
walk-sync	0.3.4	MIT
watchpack-chokidar2	2.0.1	MIT
watchpack	1.7.5	MIT
wbuf	1.7.3	MIT
webdriver-js-extender	1.0.0	MIT
webdriver-manager	12.1.8	MIT
webpack-dev-middleware	1.12.2	MIT
webpack-dev-server	2.4.5	MIT
webpack-merge	2.6.1	MIT
webpack-sources	0.2.3	MIT
webpack-sources	1.4.3	MIT

Package	Version	License
webpack	2.4.1	MIT
websocket-driver	0.7.4	Apache-2.0
websocket-extensions	0.1.4	Apache-2.0
when	3.6.4	MIT
whet.extend	0.9.9	MIT
which-module	1.0.0	ISC
which-module	2.0.0	ISC
which	1.3.1	ISC
wide-align	1.1.3	ISC
window-size	0.1.0	MIT
wordwrap	0.0.2	MIT*
wordwrap	0.0.3	MIT
wordwrap	1.0.0	MIT
wrap-ansi	2.1.0	MIT
wrap-ansi	5.1.0	MIT
wrappy	1.0.2	ISC
ws	1.1.2	MIT
ws	1.1.5	MIT
ws	4.1.0	MIT
ws	7.4.4	MIT
wtf-8	1.0.0	MIT
xml2js	0.4.23	MIT
xml2js	0.4.4	MIT

Package	Version	License
xmlbuilder	11.0.1	MIT
xmldom	0.1.31	MIT*
xmlhttprequest-ssl	1.5.3	MIT
xmlhttprequest-ssl	1.5.5	MIT
xtend	4.0.2	MIT
y18n	3.2.2	ISC
y18n	4.0.1	ISC
yallist	2.1.2	ISC
yargs-parser	13.1.2	ISC
yargs-parser	4.2.1	ISC
yargs	13.3.2	MIT
yargs	3.10.0	MIT
yargs	6.6.0	MIT
yeast	0.1.2	MIT
yn	2.0.0	MIT
zone.js	0.8.29	MIT

Third-Party Licenses

See Flix Server Licenses for a list of 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 14 AND 15 WHERE WE LIMIT OUR LIABILITY TO USERS OF OUR SOFTWARE PRODUCTS; (C) CLAUSE 18.2 REGARDING THE DATA WE MAY COLLECT AND HOW WE MAY USE IT; AND (D) CLAUSE 18.3

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 to use any Software product(s) of Foundry and/or in connection with any Maintenance and Support Agreement as defined in clause 8.5.

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"**) and 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 in writing. 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 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.

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 the terms of 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 purchases 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).

(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 the terms of Foundry's Team Login Licensing Rules as published on its 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 purchases an Individual License for Modo on a subscription basis, 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 an Individual License for Mari on a subscription basis 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 a License 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 Policy published on its website (the "Educational Policy"), Licensee warrants and represents to Foundry as a condition of the Educational License that: (i) (if Licensee is a natural person) he or she 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) (if the Licensee is not a natural person) it is an organization that will use the Software only for the purpose of training and instruction, and for no other purpose, and (iii) Licensee will at all times comply with the Educational Policy (as such policy may be amended from time to time). 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 employment or other such collaborations with studios or similar organisations. Foundry reserves the right to require evidence of graduation by the Licensee. Each Graduate License shall be deemed to be a one (1) year Rental License after which it shall automatically expire.

(l) **"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.

(m) **"Modo Steam Edition"**

A version of Modo with limited functionality as described in the Documentation is available to purchase on discount terms through Valve Corporation's Steam store. If Licensee has purchased such version, Licensee warrants and represents to Foundry as a condition of the Agreement that: (i) Licensee is a natural person; and (ii) Licensee will use the Software strictly through Steam and only for personal, recreational and non-commercial use, except only that if Licensee uses the Software to create assets and content Licensee may sell such assets and content through Valve's Steam Workshop.

(n) **"Modo indie"** and **"Mari indie"**

Variants of Modo and Mari with limited functionality as described in the Documentation are available to purchase on discount terms through Valve Corporation's Steam store. 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 either variant; 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 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 and on only one computer at a time.

(p) **"Subscription License"**

Any reference to a Subscription License shall mean a Modo Subscription License, Nuke Indie License or a Mari Individual Subscription License, as the case may be.

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 except Modo (in respect of which there is no limit on the amount of Headless Rendering allowed), 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"**). 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 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, 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 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.

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 as expressly set out in clause 8.8, 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 authorised 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 Autorenewal Period for a Subscription License, the License Fee as notified by Foundry (or its reseller) on or about the applicable Renewal Date. Unless stated otherwise, any Licence Fee notified to the Licensee by Foundry (or its reseller) is exclusive of 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.

7.3 In the cases of Non-Commercial NUKE or Trial Licenses for the avoidance of doubt, 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 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.

8.3 Unless Licensee opts out of auto-renewal in accordance with clause 8.6 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 22. The reminder email will be sent not less than thirty (30) days prior to the relevant Subscription Renewal Date.

8.5 Subject to Licensee's timely payment of the applicable License Fee, a Subscription Licence shall include access to certain maintenance and support services for the Subscription Licence in accordance with the terms of the Maintenance and Support Agreement which is available on Foundry's website (the **"Maintenance and Support Agreement"**).

8.6 **Opting Out of Auto-renewal.** If Licensee wishes to opt out of auto-renewal then you must email licenses@foundry.com providing details of the Subscription Licences 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 8.6 then your Subscription License will not auto-renew and shall expire at the end of the then-current Subscription Period. If you require further Maintenance and Support then please refer to Foundry's Maintenance and Support Policy as published on its website from time to time.

8.7 **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.

8.8 **Payment in Installments.** If Licensee is paying the License Fee for the Subscription License in instalments (as shall be noted in the applicable invoice), then the License Fee shall be owing on the Subscription Start Date and any/all Subscription Renewal Dates and shall be payable in twelve (12) equal monthly instalments thereafter, or on termination of this agreement if earlier. By placing an order for a Subscription License payable in instalments, Licensee requests and authorizes Foundry (or its agents) to take one twelfth of the applicable annual License Fee from the means of payment provided by Licensee every month during the Subscription Period. The Subscription License will terminate automatically if payment cannot be taken from the means of payment provided by Licensee for any one month. In the event of termination, Licensee shall remain liable for the balance of the License Fee which shall become payable immediately and in full.

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. MAINTENANCE AND SUPPORT

If the Licensee has purchased maintenance and support services from Foundry for any Product licensed under this EULA, or if the Licensee is entitled to receive maintenance and support services for a Subscription Licence in accordance with clause 8.5, then Foundry shall provide those services subject to the terms of its Maintenance and Support Agreement available on its website. Foundry may from time to time and at its sole discretion vary the terms and conditions of the Maintenance and Support Agreement. If Licensee allows their maintenance support services to lapse for a period of time (**"Lapsed Period"**) and then seeks to purchase further maintenance and support services, Foundry reserves the right to require Licensee to back-pay fees for the Lapsed Period.

11. 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.

12. LIMITED WARRANTY

12.1 Subject to clause 12.3, 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.

12.2 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.

12.3 If Licensee purchases a license of the Software that is of a fixed term duration, the Warranty Period in clause 12.1 shall apply only to Licensee's first purchase of such license and not to any subsequent renewal(s) even if a renewal involves another download.

13. INDEMNIFICATION

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 modification of the Software or Documentation by Licensee or any person connected with Licensee infringes the intellectual property rights or other proprietary rights of any third party.

14. 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 requirements. The Software and Documentation is supplied only for Licensee's internal use for its business, and not for any re-sale purposes or for the provision of the Software (whether directly or indirectly) to third parties. Foundry shall not under any circumstances whatever be liable to Licensee, its affiliates, officers, directors, shareholders, employees, agents or other representatives, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with the Agreement for loss of profits, sales, business, or revenue, business interruption, loss of anticipated savings, loss or corruption of data or information, loss of business opportunity, goodwill or reputation (in each case whether the loss is direct or indirect) or any indirect or consequential loss or damage. In respect of any other losses, Foundry's maximum aggregate liability under or in connection with the Agreement whether in contract, tort (including negligence) or otherwise, shall in all circumstances be limited to the greater of US$5,000 (five thousand USD) and a sum equal to the License Fee. Nothing in the Agreement shall limit or exclude Foundry's liability for death or personal injury resulting from our negligence, for fraud or fraudulent misrepresentation or for any other liability that cannot be excluded or limited by applicable law. This EULA sets out the full extent of our obligations and liabilities in respect of the supply of the Software and Documentation. Except as expressly stated in writing in this EULA, there are no conditions, warranties, representations or other terms, express or implied, that are binding on Foundry. Any condition, warranty, representation or other term concerning the supply of the Software and Documentation which might otherwise be implied into, or incorporated in, the Agreement, whether by statute, common law or otherwise, is excluded to the fullest extent permitted by law.

15. 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 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, or loss of business opportunity. 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. Loss or damage is foreseeable if they were an obvious consequence of a breach or if they were contemplated by Licensee and Foundry at the time of forming the Agreement. 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 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.

16. TERM; TERMINATION

16.1 The Agreement is effective upon Licensee's download of the Software, and the Agreement will remain in effect until termination or expiry. Licensee may terminate the Agreement on written notice to Foundry if Foundry is in material breach of this Agreement and fails to cure the breach within 10 (ten) working days of receiving notice of such breach. If Licensee breaches the Agreement, Foundry may terminate the License immediately by notice to Licensee.

16.2 If the Agreement expires or is terminated, the License will cease immediately and Licensee will immediately cease use of any Software and Documentation and either return to Foundry all copies of the Software and Documentation in Licensee's possession, custody or power or, if Foundry directs in writing, destroy all such copies. In the latter case, if requested by Foundry, Licensee shall provide Foundry with a certificate confirming that such destruction has been completed.

16.3 Foundry reserves the right to terminate and/or suspend the License as it deems reasonable in its sole discretion by notice to Licensee if 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 Maintenance and Support Agreement or if the Licensee is otherwise in breach of or fails to comply with any term of the Agreement.

16.4 Foundry may also terminate this EULA if Licensee becomes subject to bankruptcy proceedings, becomes insolvent, or makes an arrangement with Licensee's creditors. This EULA will terminate automatically without further notice or action by Foundry if Licensee goes into liquidation.

17. 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.

18. INSPECTION AND INFORMATION

18.1 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.

18.2 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.

18.3 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 18.2(a) to (e) above.

19. 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 (c) 2001 - 2022 The Foundry Visionmongers Ltd. All Rights Reserved.

* Unpublished-rights reserved under the Copyright Laws of the United Kingdom.

20. SURVIVAL

Clause 6, clause 7 and clauses 11 to 23 inclusive shall survive any termination or expiration of the Agreement.

21. IMPORT/EXPORT CONTROLS

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 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; (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) for use in any nuclear, chemical or biological weapons, or missile technology end-use.

22. 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 Notices. Licensee shall notify Foundry of an email address for the provision of any notices and 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 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.

23. 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 3 November 2022.

Copyright © 3 November 2022 The Foundry Visionmongers Ltd.
All Rights Reserved. Do not duplicate.