GUIDE TO FLIX

Version 6.3.7

FOUNDRY.

The Foundry
5 Golden Square
London
W1F 9HT
Rev: Wednesday, November 10, 2021

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

macOS	Big Sur (11), Catalina (10.15) or Mojave (10.14) A 64 bit processor @ 2GHz minimum (2 CPU cores minimum, 12 CPU cores recommended, 12 vcpu* for a VM) 8GB of RAM minimum, 32GB of RAM recommended *tested using VMWare vSphere. Minimum bandwidth requirements: 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.
Linux	CentOS/RHEL 6.10 or CentOS/RHEL 7 (recommended) A 64 bit processor @ 2GHz minimum (2CPU cores minimum, 12 CPU cores recommended, 12 vcpu* for a VM) 8GB of RAM minimum, 32GB of RAM recommended

	*tested using VMWare vSphere. Minimum bandwidth requirements: 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.

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.

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

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

Installing and Running MySQL 5.7

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 MySQL user Flix Server is assigned 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.

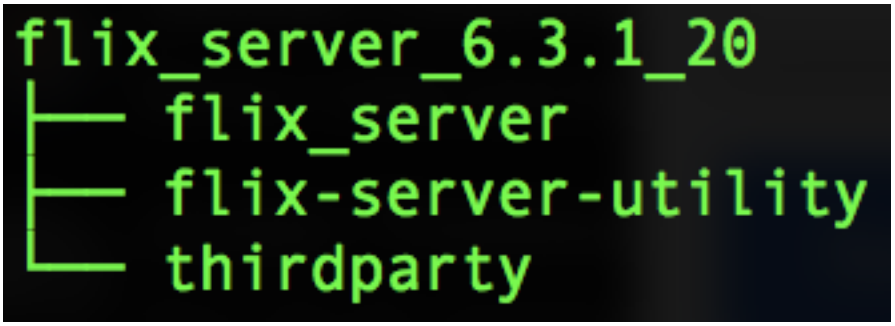
   > **Note:** You must be logged in to your Foundry account to download Flix Server.

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:

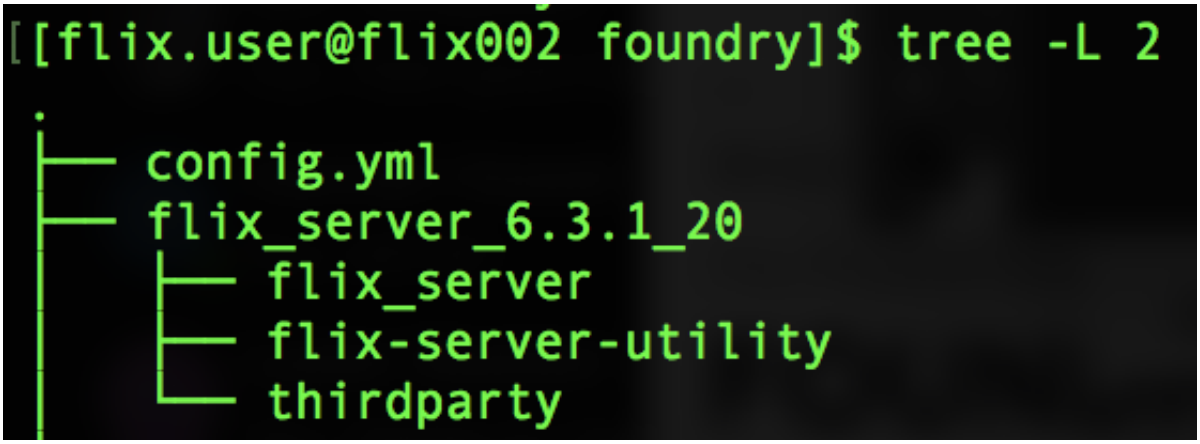


3. Place these files wherever you want Flix Server to be installed.

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 port 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.

Setting Up User Authentication

Flix provides three ways to authenticate users: LDAP (Lightweight Directory Access Protocol) , OAuth and the User Management system.

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. For guidance on formatting, please refer to the example at the end of this section.

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.

Example of a LDAP subsection in a 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
```

Article: For more information on setting up LDAP and for troubleshooting tips, please refer to this Knowledge Base Article.

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.

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:

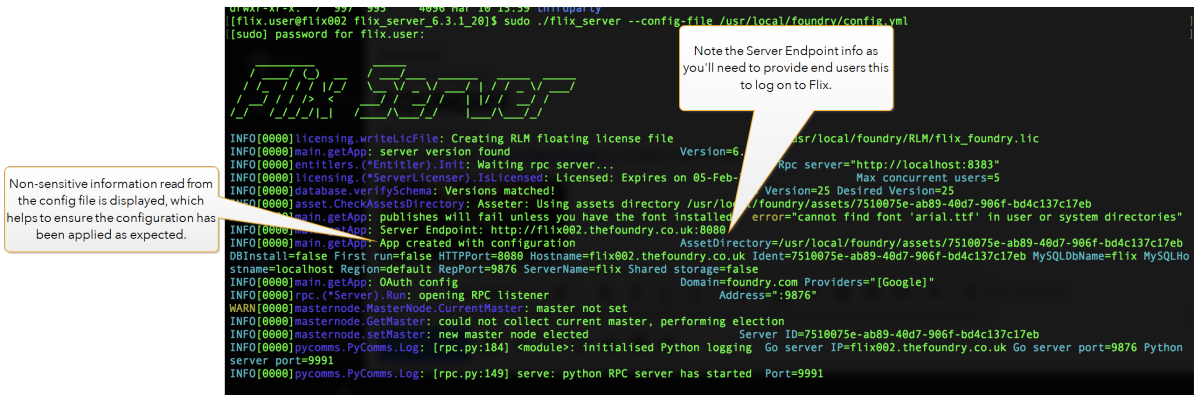
```
chmod +x flix_server
```

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

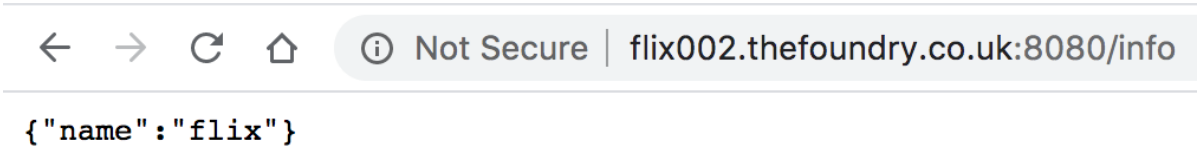
Now that Flix Server is running, it's a good idea to ensure it's accessible by other computers, as end users only access it from another computer at the studio (on the same network), or remotely (over VPN).

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:


   ```
   {"name":"flix"}
   ```

   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.

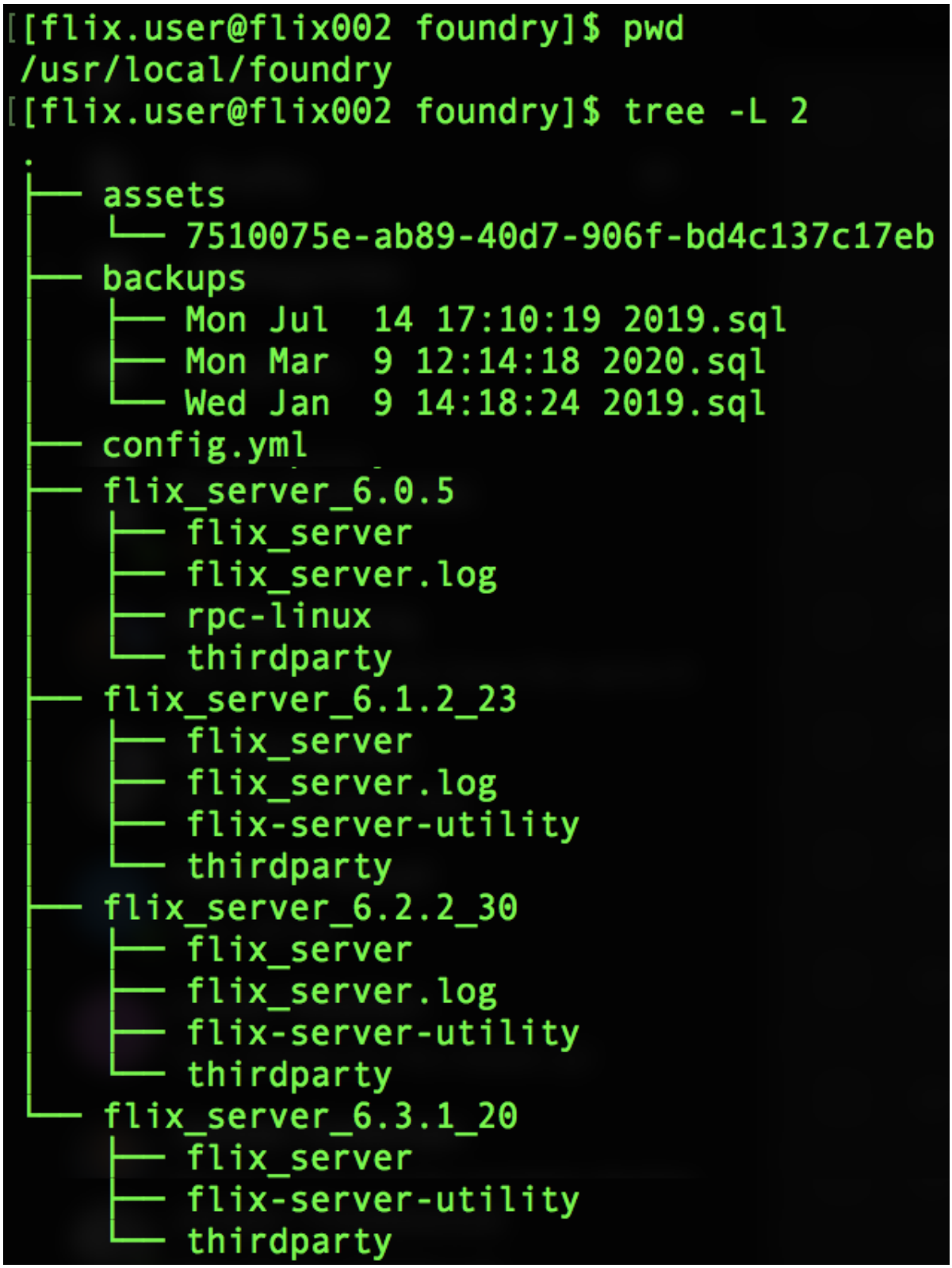
Managing Your Flix Deployment

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.

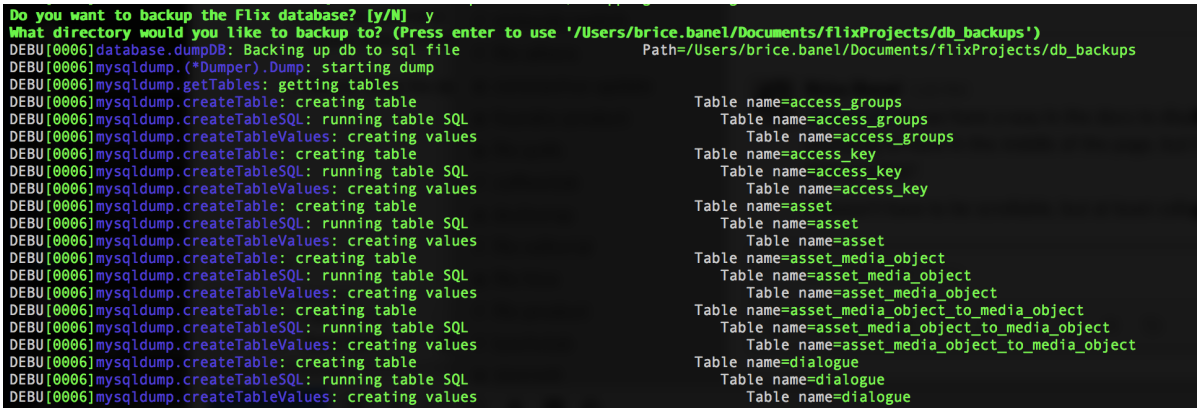
   If you want to upgrade your database later, you can use the **--db-upgrade** mode to mutate the database schema to the latest version:

```
./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.

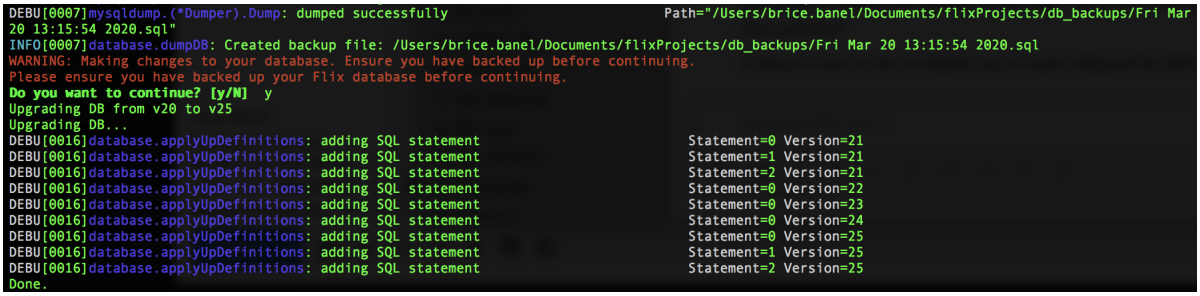
Example of database backup:



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:



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.

Flix Server Version	Required Database Version
6.0.0	4
6.0.1 - 6.0.5	5
6.1.0 - 6.1.2	12
6.2.0 - 6.2.2	20
6.3.0 - 6.3.2	25
6.3.3 - 6.3.4	26
6.3.5 - 6.3.6	28
6.3.7	29

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**

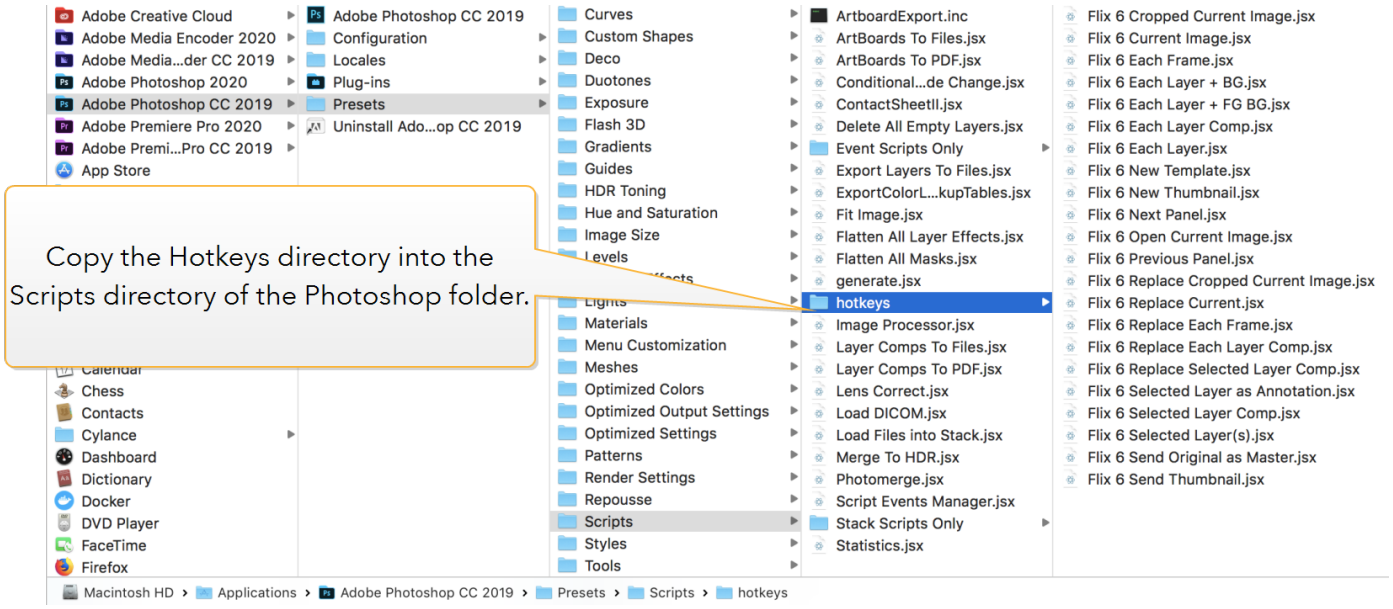
- Windows: **resources\flix.zxp**

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, for Photoshop CC 2019:
   - Mac: copy the **hotkeys** folder from **/Library/Application Support/Adobe/CEP/extensions** > **/Applications/Adobe Photoshop CC 2019/Presets/Scripts/**
   - Windows: copy the **hotkeys** folder from **C:\Program Files (x86)\Common Files\Adobe\CEP\extensions** > **C:\Program Files\Adobe\Adobe Photoshop CC 2019\Presets\Scripts**

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



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`

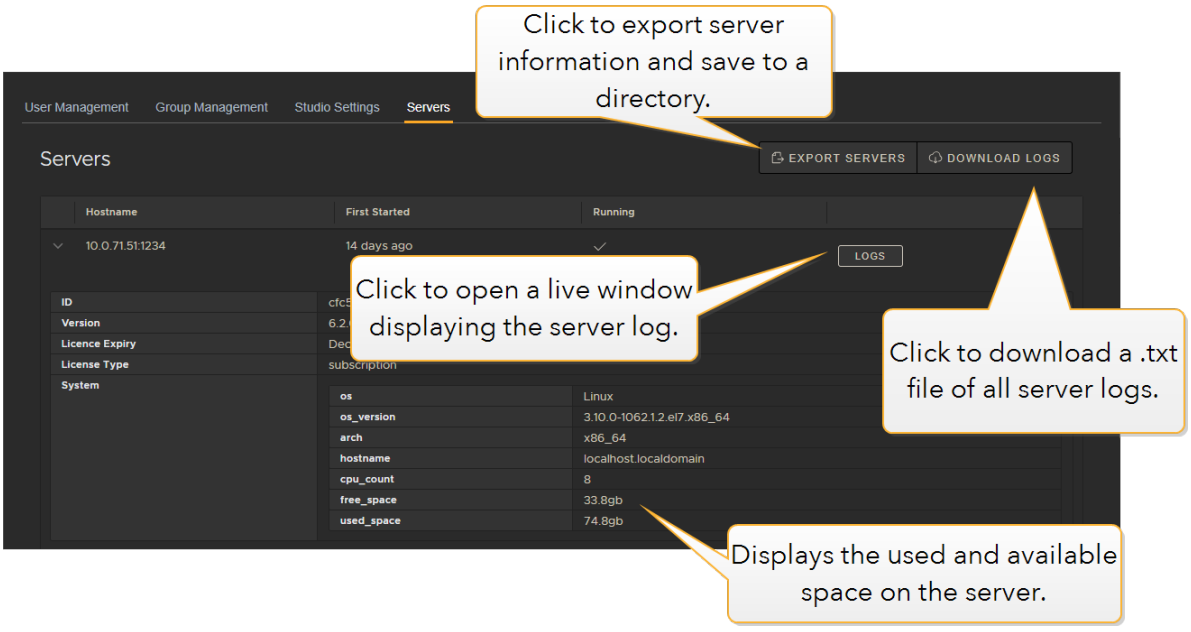
You can repeat these steps to add additional servers.

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 (seeSetting 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

Mac	Big Sur (11), Catalina (10.15) or Mojave (10.14)

	1GB of RAM
Windows	Windows 10 An Intel Pentium 4 processor or later that's SSE2 capable 1GB of RAM

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

Installing and Launching Flix Client

To download and install Flix Client:

Windows:

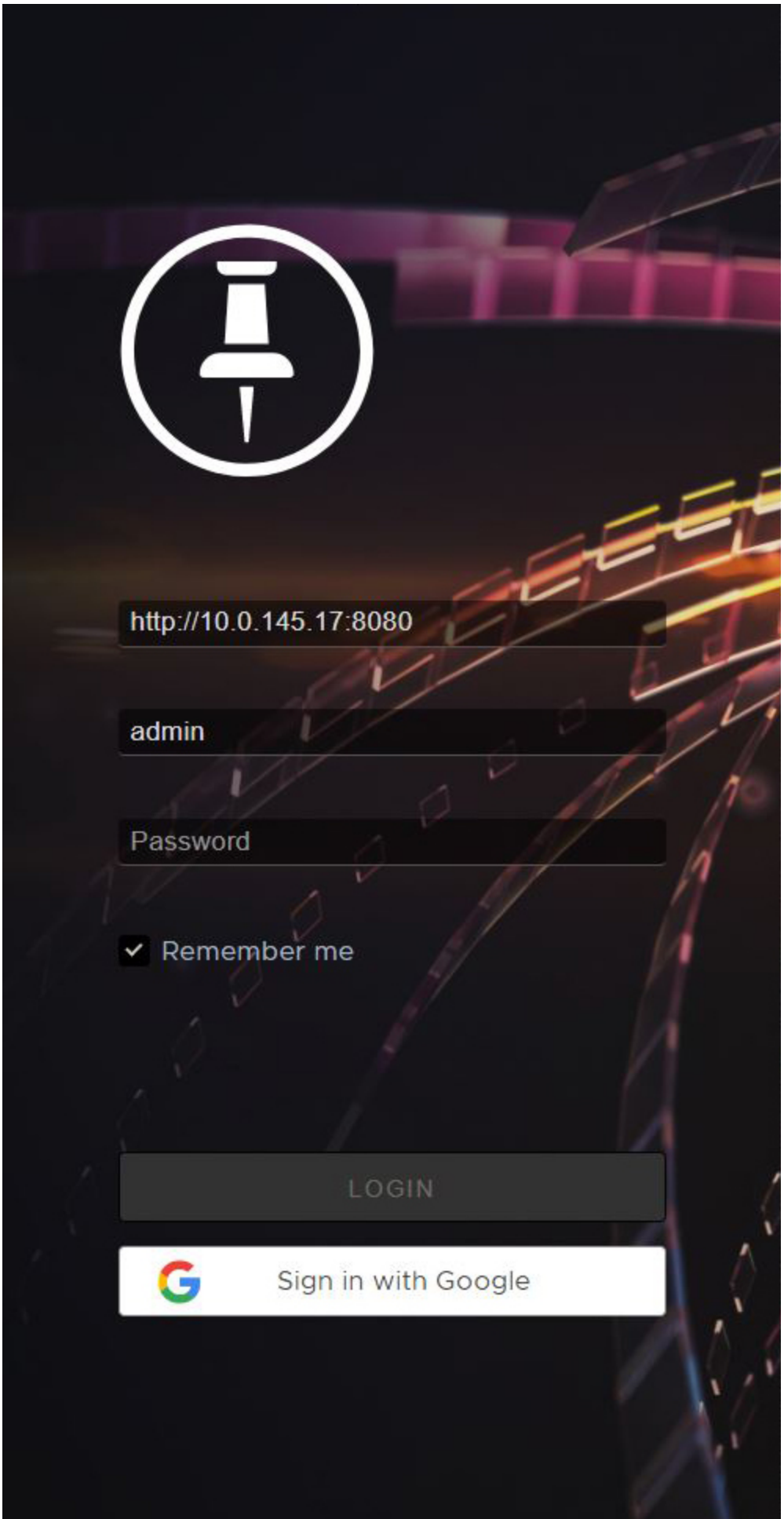
1. Download and unzip the **.zip** file located here.
2. In the unzipped folder, double-click or open **Flix.exe**.

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.

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.


http://10.0.145.17:8080
admin
Password
Remember me
LOGIN
Sign in with Google

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 User Management.

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

Flix opens 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 toSetting Up HTTPS for more information on setting up Flix Server with HTTPS.

User Management

Creating a User Account

As an administrator using the **Flix User Management** system you can create user accounts and assign them to specific groups. Only admin users can create or edit users and groups.

To create a user account:

1. Navigate to **File** > **Management Console**.
2. In the **User Management** tab, in the **Create user** section, fill in the **User name**, **Email address, Password**, **Confirm password** and **Groups** fields.

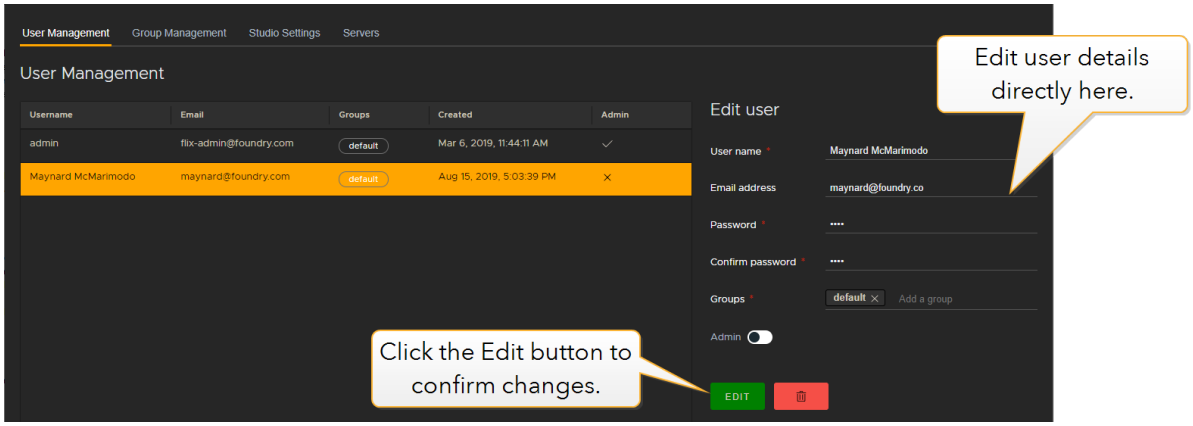
Note: To create a group, see Creating, Editing and Deleting a User Group.

3. Select whether you want to give the user an **Admin** status.
   The toggle turns green to indicate the admin permissions.
4. 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:

1. Navigate to **File** > **Management Console**.
2. In the **User Management** tab, select the user account in the table.
   The selected user account is highlighted in yellow. To deselect a group click it again.



3. To edit a user's details, enter changes directly into the **Edit user** section.
   The **Edit** button turns green to show that it has become active.

Note: Only Admin users can edit user details or make accounts Admin accounts.

4. Click the **Edit** button to confirm the changes.
   The user account is updated with your edits.

To delete a user account:

1. Navigate to **File** > **Management Console**.
2. In the **User Management** tab, select the user account in the table.
   The selected user account is highlighted in yellow.
3. In the **Edit user** section, click the Trash button.
   The user account is deleted.

Creating, Editing and Deleting a User Group

You can use groups to allow users to access certain shows.

1. Navigate to **File** > **Management Console**.
2. In the **Group Management** tab, in the **Create Group** section, fill in the **Group Title** field and click **Create**.

To edit a user group:

1. Select the group in the table.
   The selected group is highlighted in yellow. To deselect a group click it again.
2. Edit its Title in the **Edit Group** section.
3. Click **Edit**.
   The group is updated.

To delete a user group, select the group in the table and click the Trash button.

Flix User Guide

Getting Started

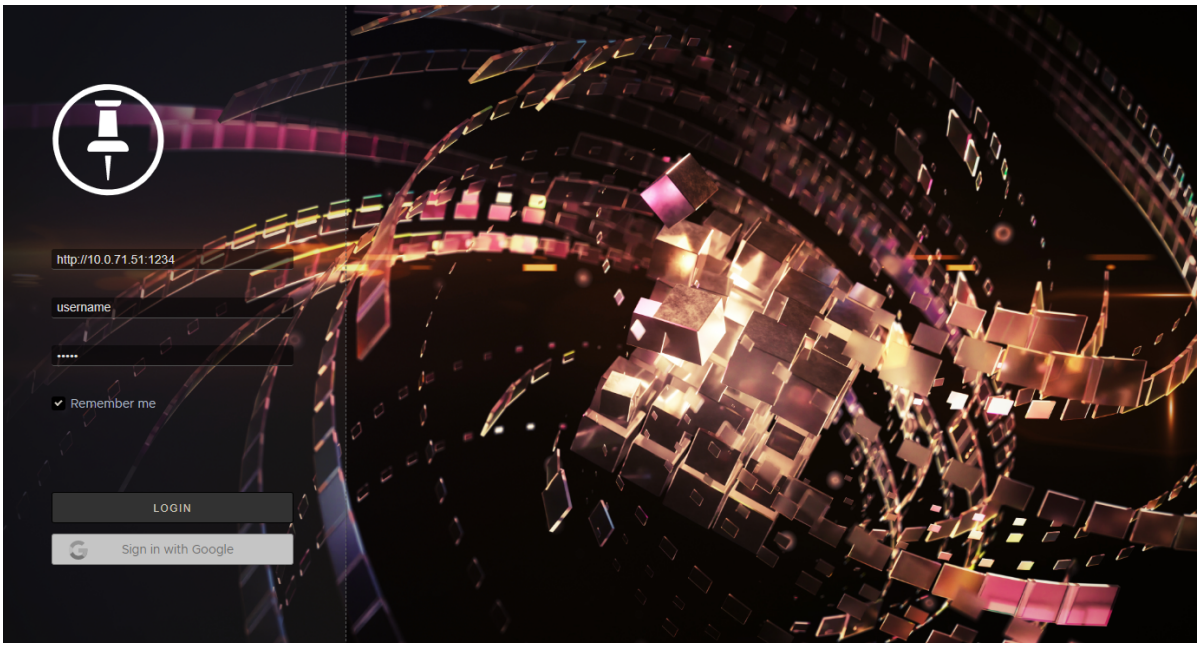
Flix is a story development hub for animated film and TV, gaming, and other visual narrative mediums. Watch the video below for a brief overview of how Flix works.

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**

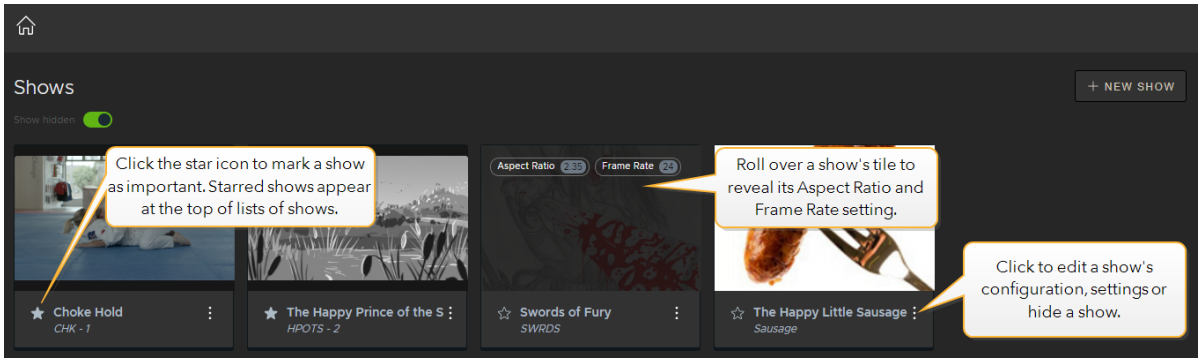
   This opens the login page.



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**.

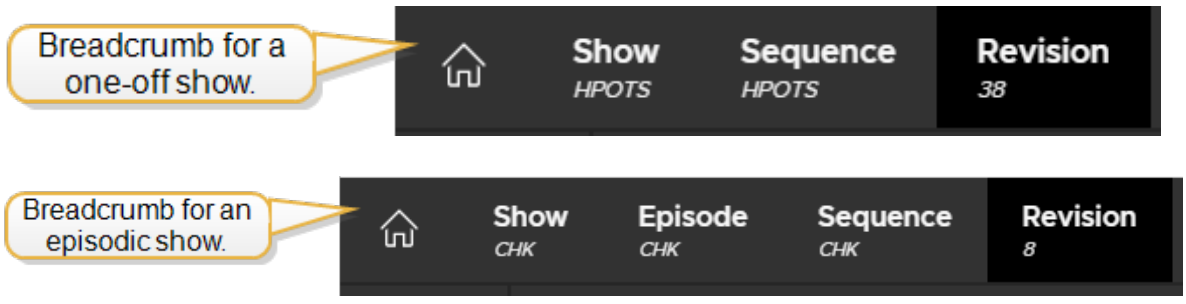
Tip: Check the **Remember me** box to make your next login easier.

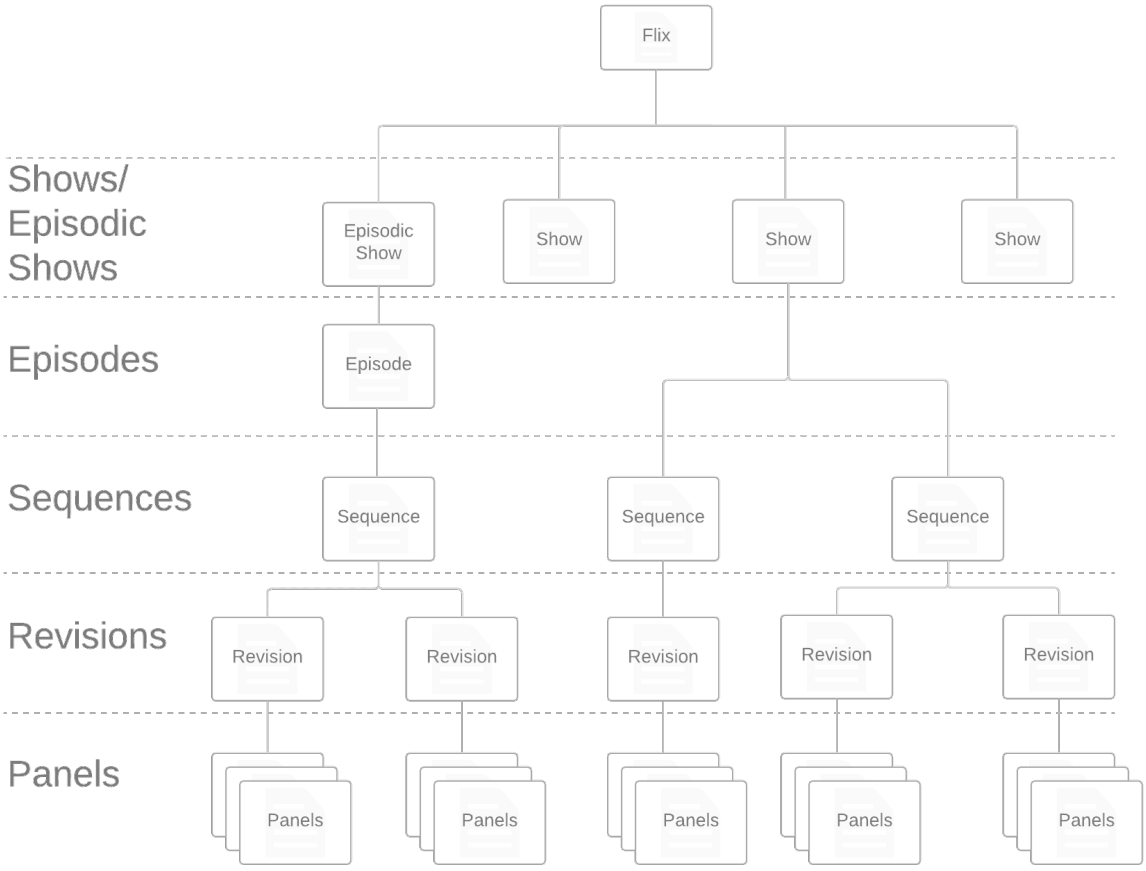
Flix opens at the **Show** level. From here you can open an existing show, create a new one or set user preferences for shows.



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.



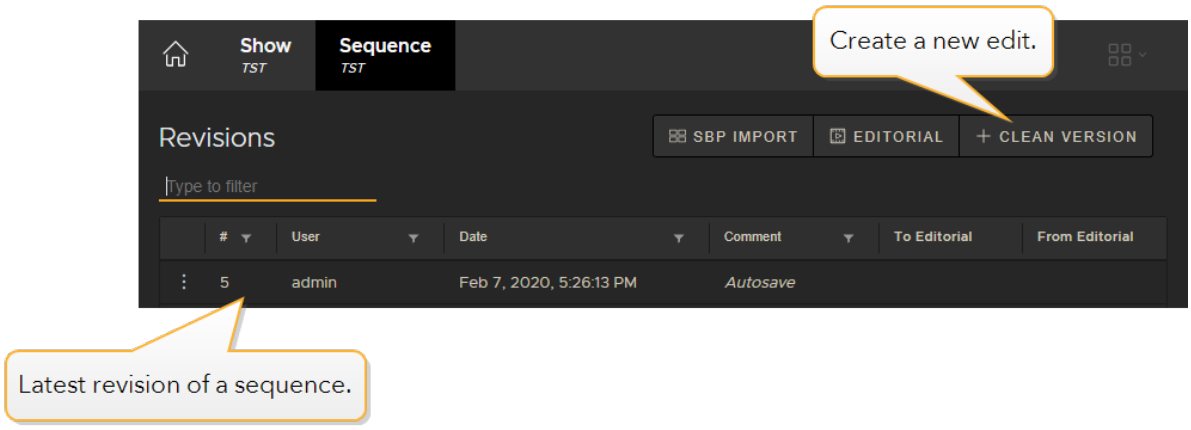


Loading an Existing Sequence

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.



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 to establish the settings for all Flix users first.

Studio Level 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:

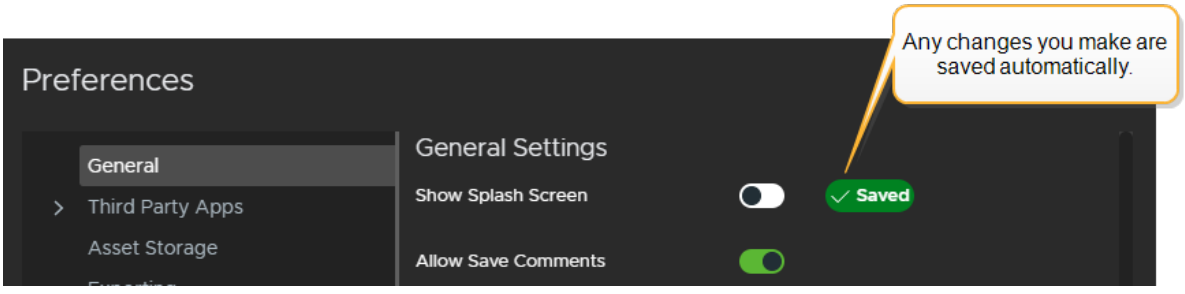
1. Navigate to the **Shows** level, click the More Options button of the required show and click **Settings**.
   This opens the **Show Settings** dialog for that specific show.
2. Edit the required preferences and enable their corresponding checkboxes under **Enforce at Show Level**.
   This overrides the preferences set in the Flix Preferences dialog and at the studio level for that specific show.

Note: The Studio Settings can only be changed by admin users. The Show Settings can only be changed by the owner of the show or an admin user.

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 between common ratios such as 1.77:1 (16:9) and 2.39:1.
Permissions	
Groups	Specifies the groups of users who can access this show. For more information on creating groups, please refer to User Management.

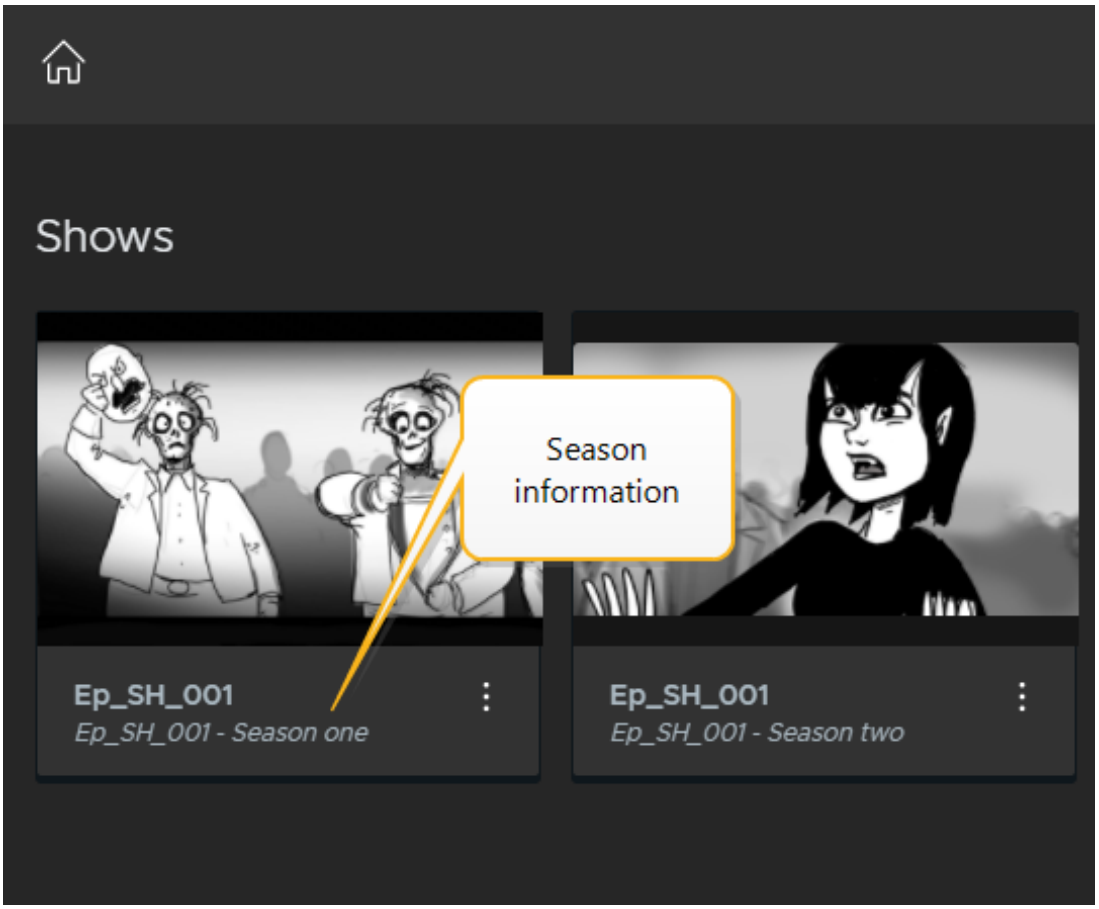
3. Click **Create**.
   Your new show is added at the **Shows** level.

Creating Additional Seasons

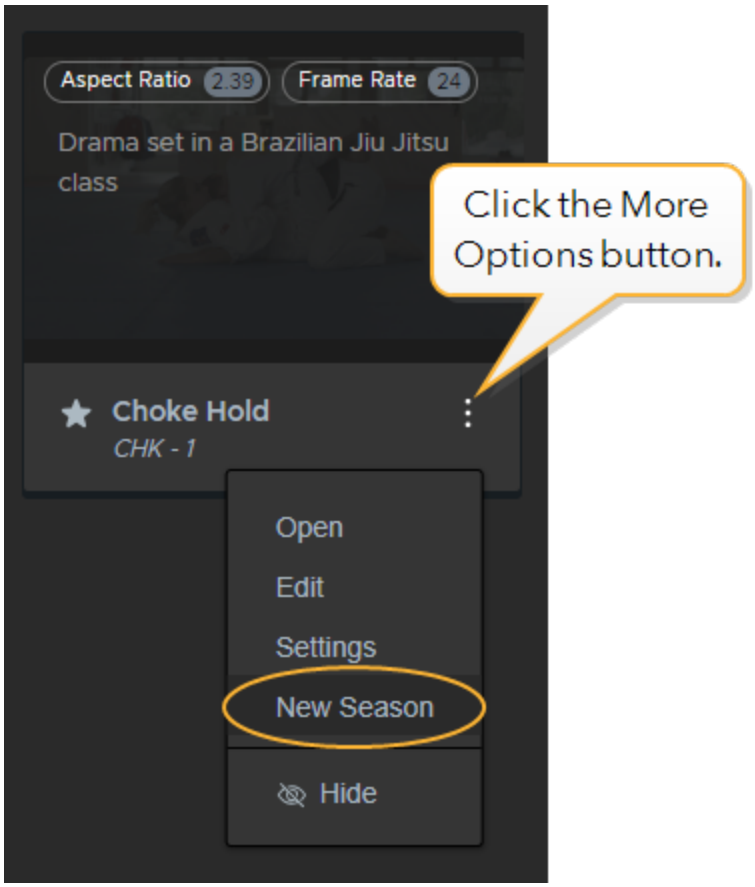
A season of a show is stored at the **Shows** level and contains a number of episodes.

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**.



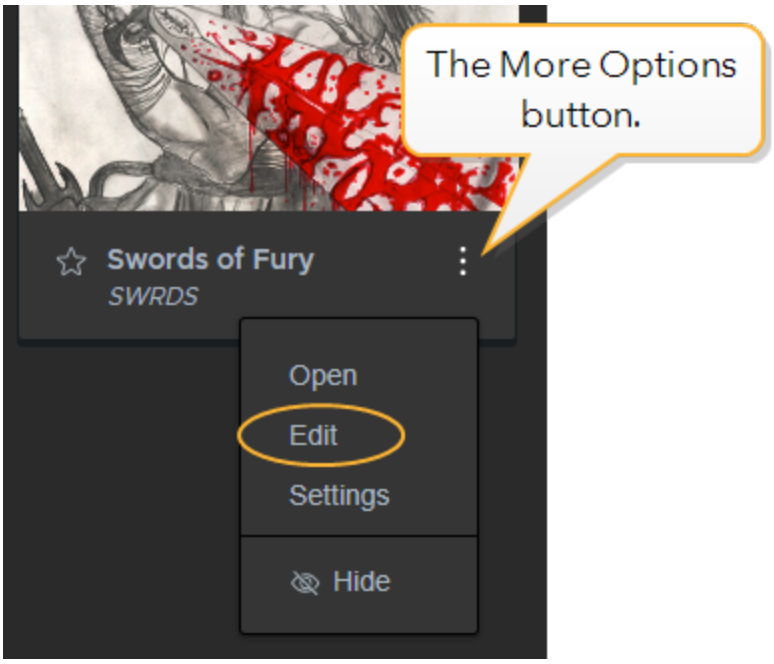
This opens the **Create Show** dialog. The details are pre-filled with the information you provided for the previous season.

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.

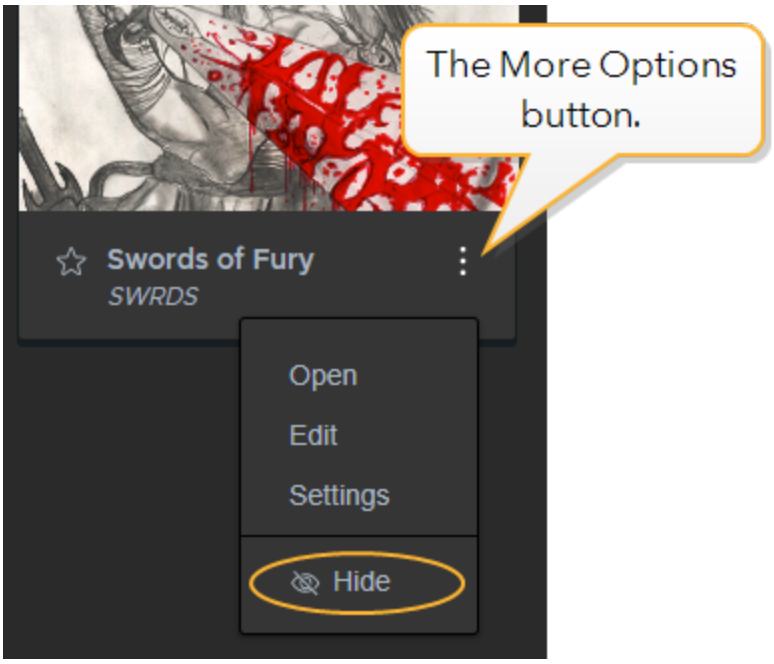
3. Click **Update** to save your edits.

Hiding a Show

Admin users 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: Admin users can toggle the **Show hidden** button ON to display all hidden shows. This button is set to OFF by default.

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.

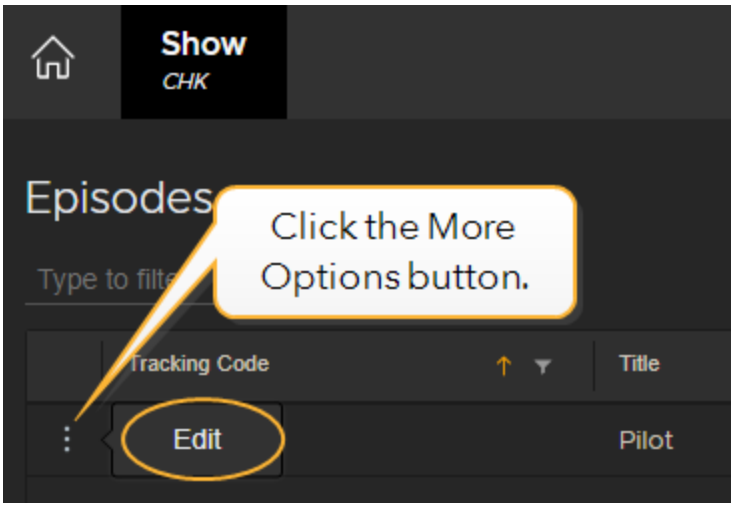
3. Click **Create**.
   You can continue creating additional episodes in this window. Click **Close** when finished.

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**.

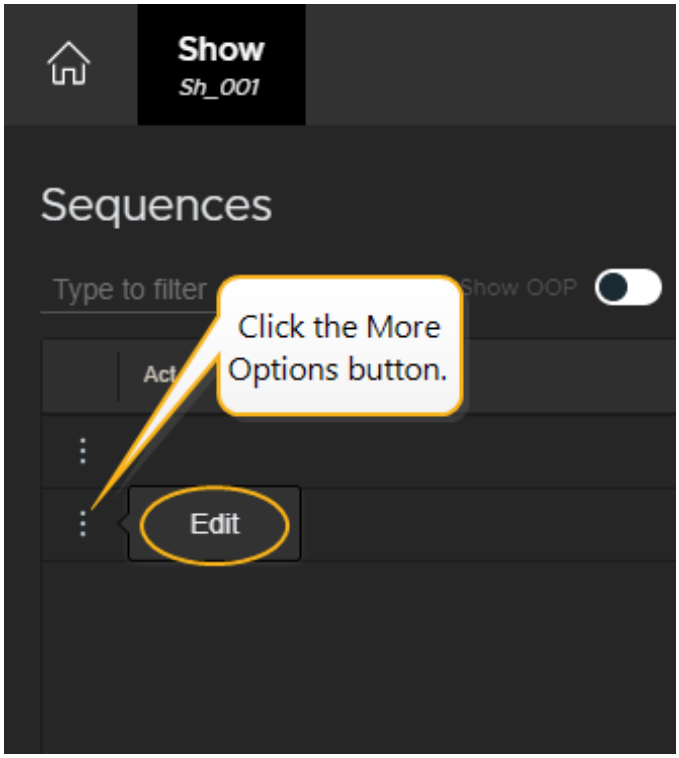
   The **New Sequence** menu opens.

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**.



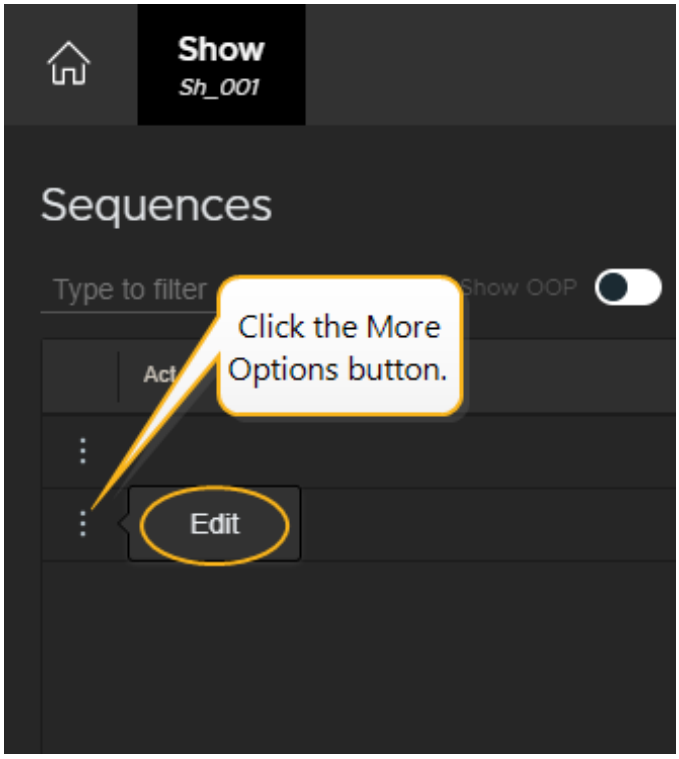
This opens the **Edit Sequence** dialog.

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**.

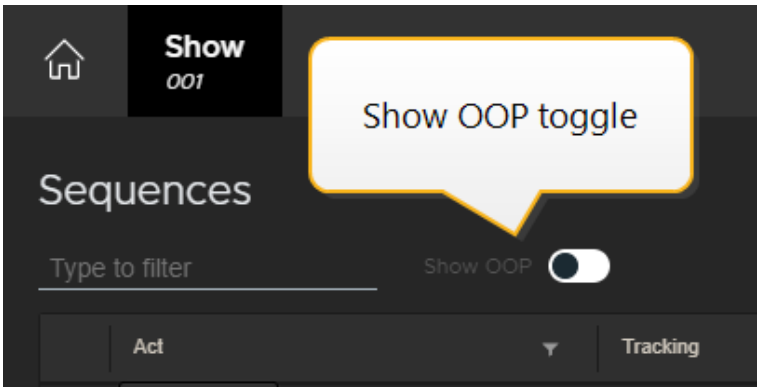


This opens the **Edit Sequence** menu.

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.

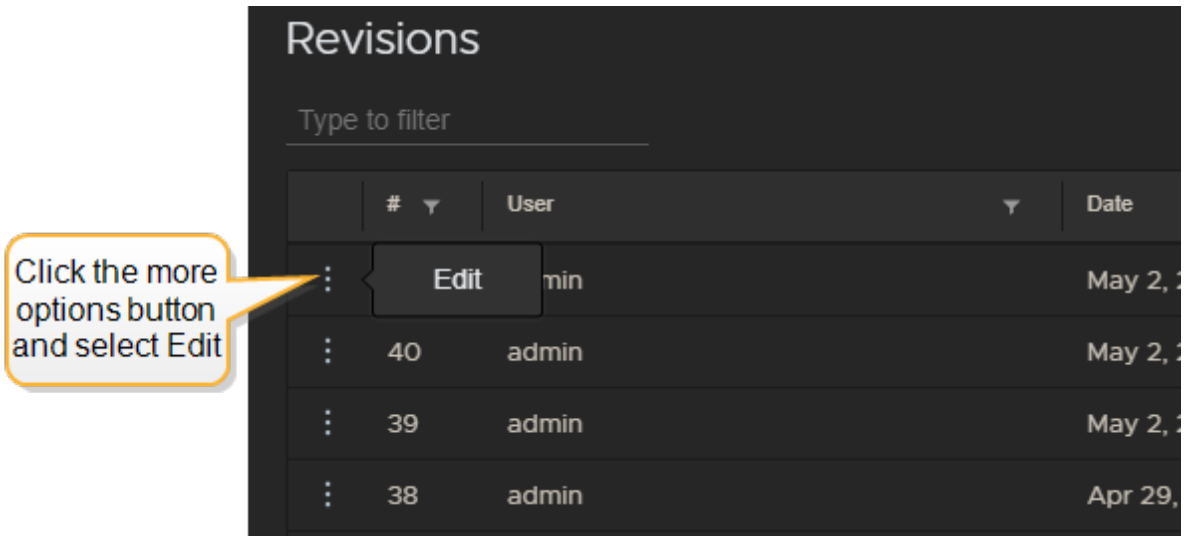
Tip: Duplicate selected panels with the **Duplicate** button.

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**.

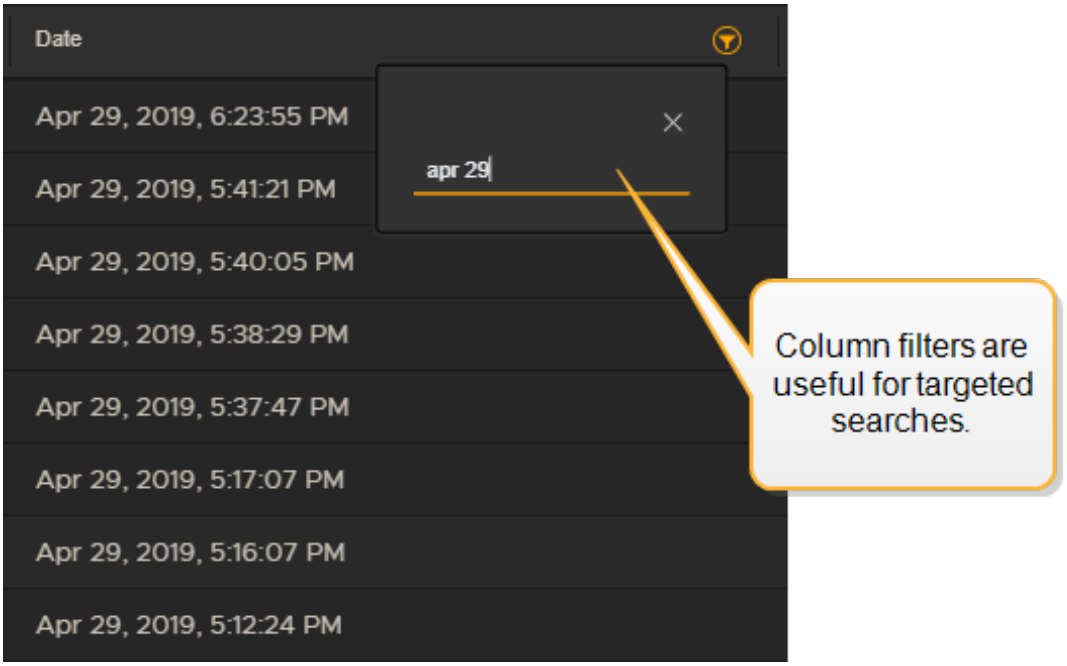


3. Type in the comment "Use for screening" and click **Save**.
4. Above the Revisions list, type the word "screening" into the filter.

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.

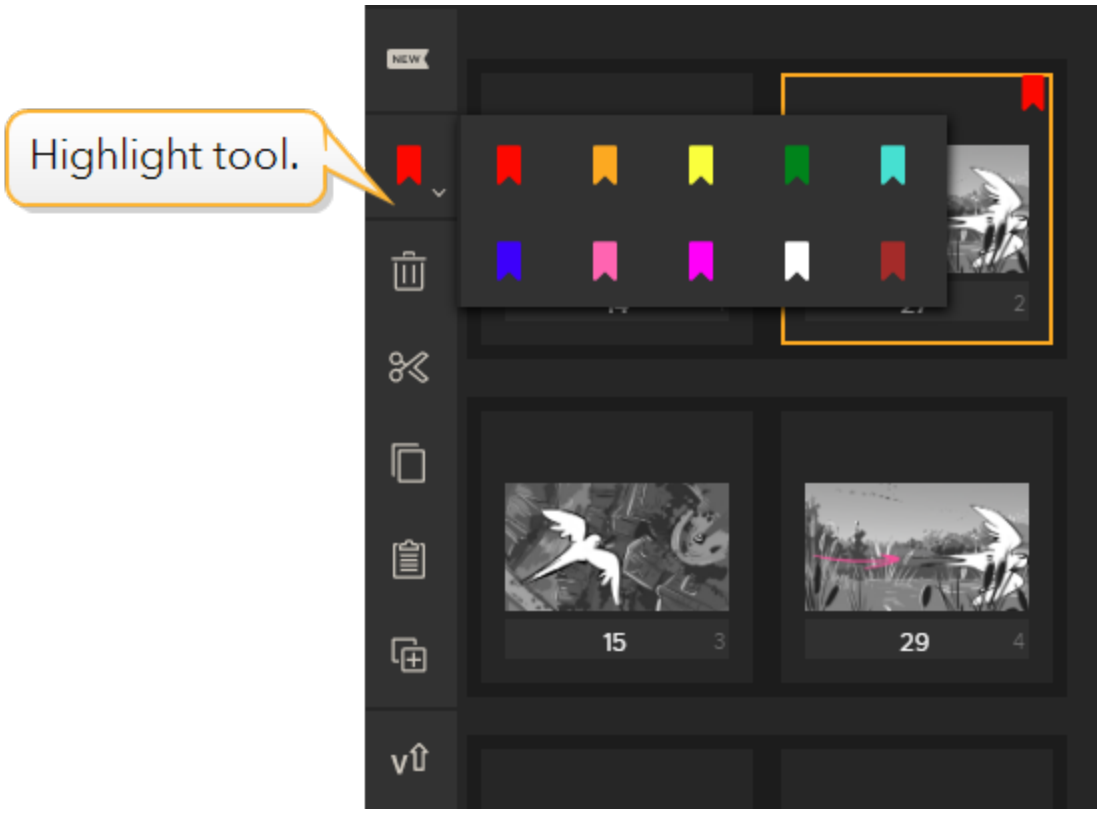
The **User**, **Date** and **Comments** columns each have their own filters, so you can narrow down your search results even further. See below.



Highlighting Panels

To highlight panels in your sequence:

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



To remove highlights:

1. Select the highlighted panels.
2. In the left toolbar, click the Highlight button.
   This removes the highlight from the selection.

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.

• **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.

Article: Read the Using Premiere Markers in Flix Knowledge Base article for more information on how markers are used in Flix.

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

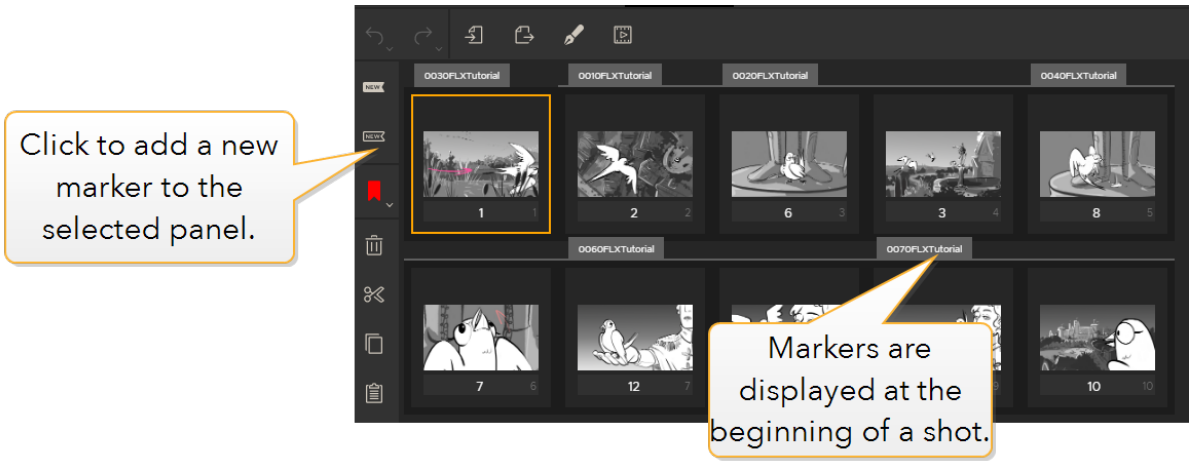
To display markers in the panel browser:

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.

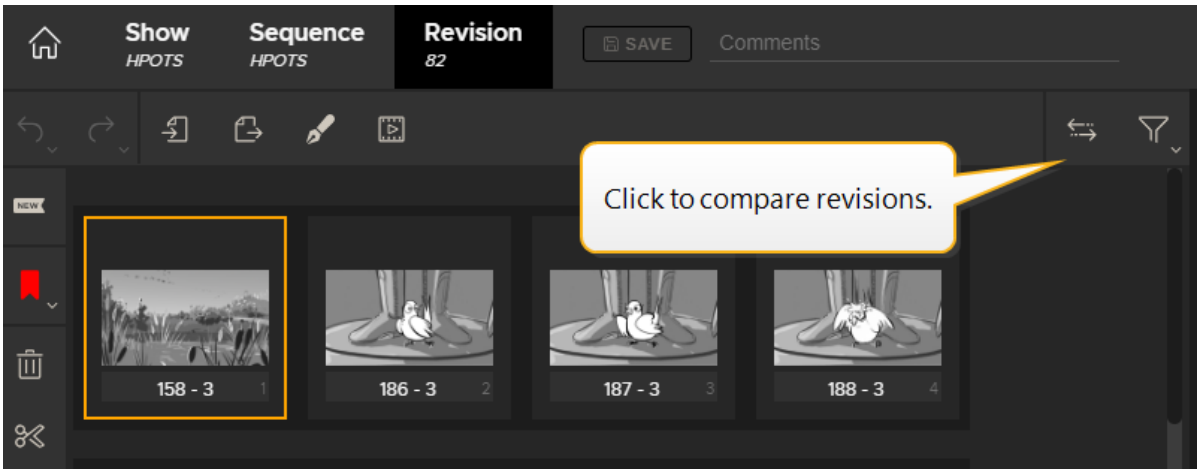
A marker appears on the selected panel.

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



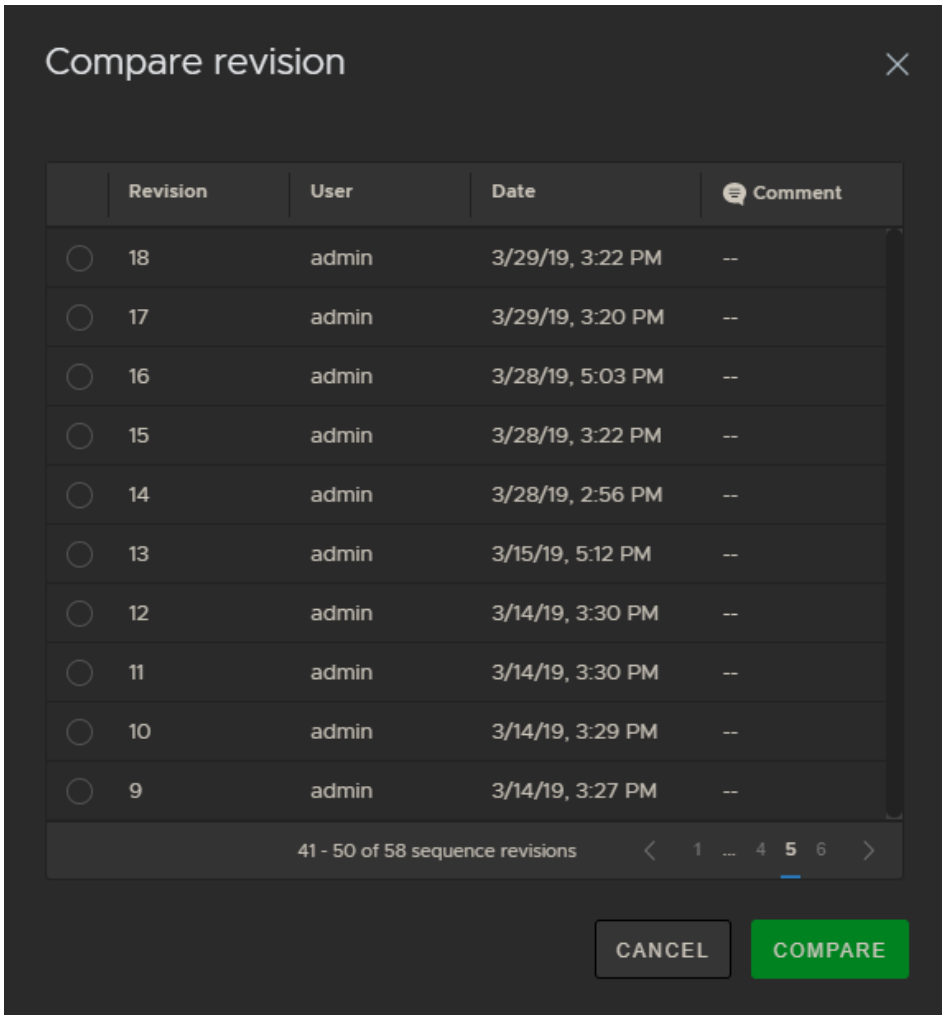
Comparing Edits

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.

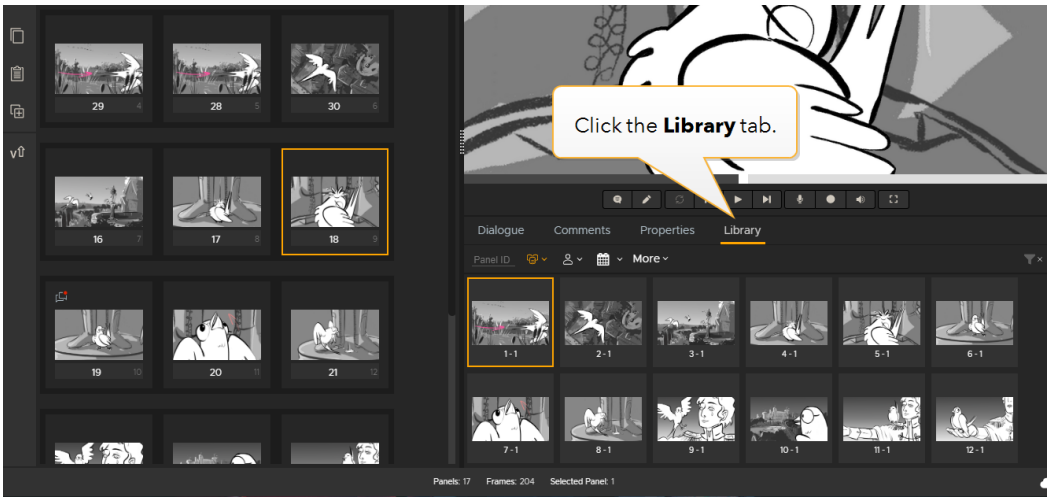
Note: icons, which indicate that panels have been deleted, only display when comparing an old sequence revision to a newer one, for example comparing v4 with v10.

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.



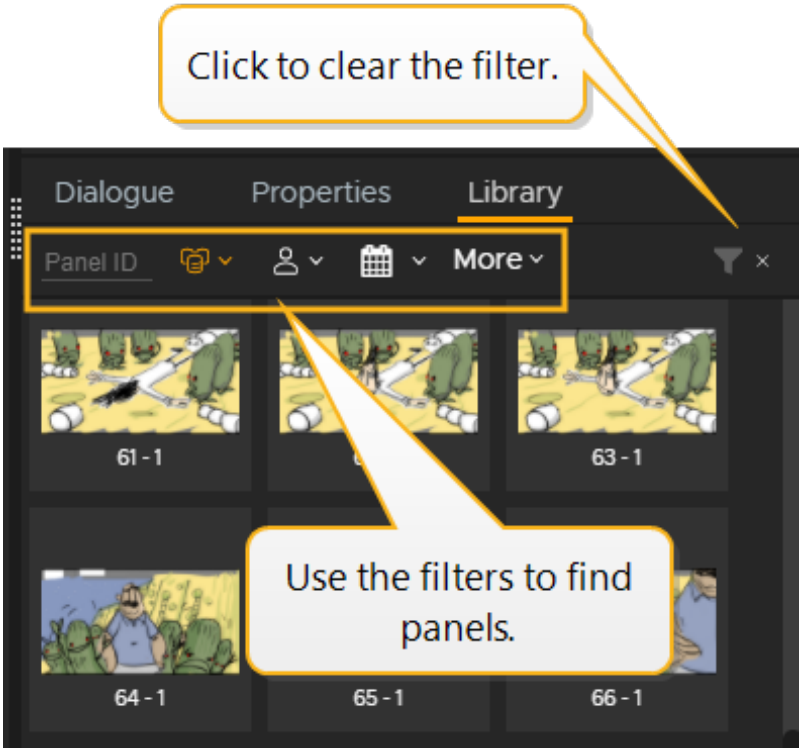
2. Click the Filter by Sequence button and select a specific sequence or episode.

   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.

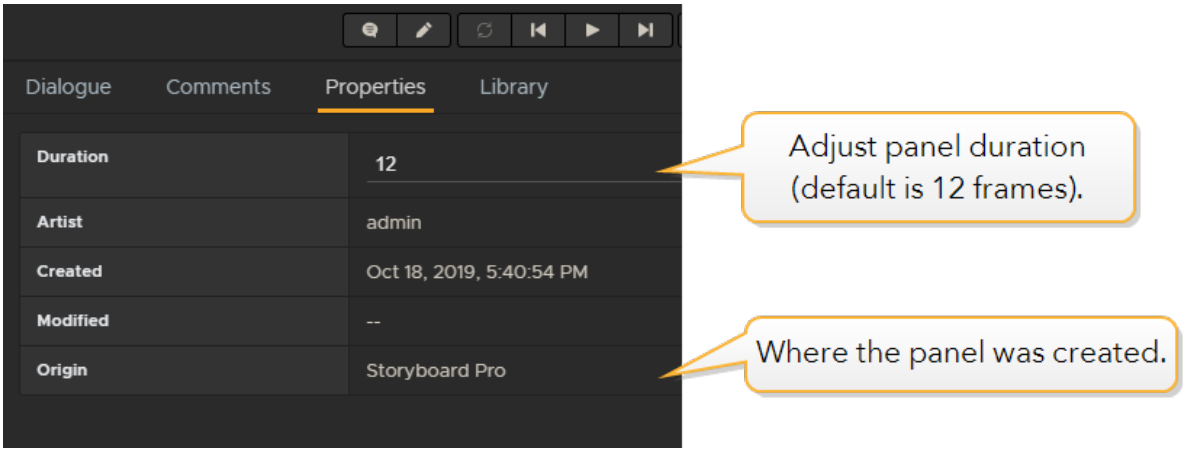
Note: Click the Reset Filters button to clear all applied filters.



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).

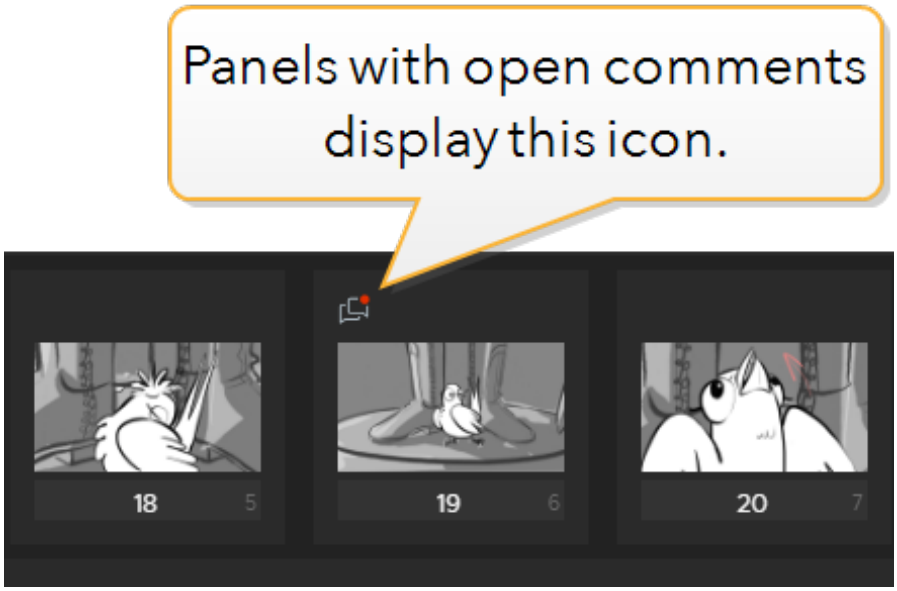
Adding Comments to Panels

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.

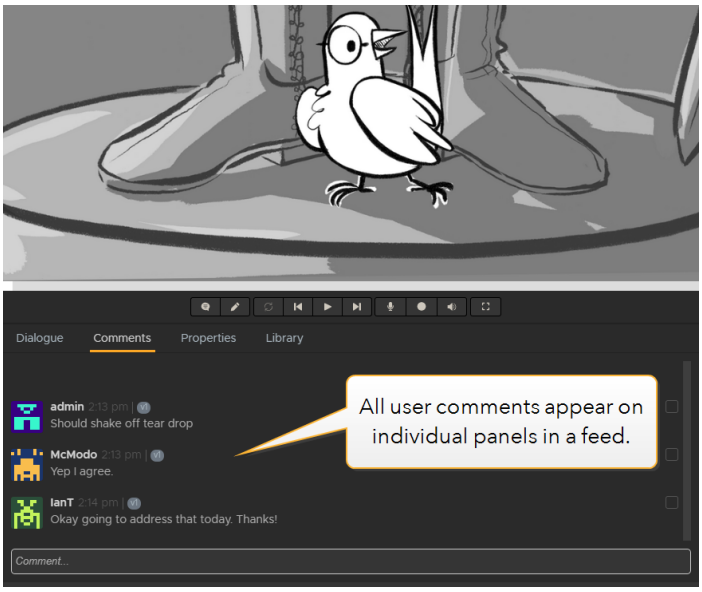
To add a comment:

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

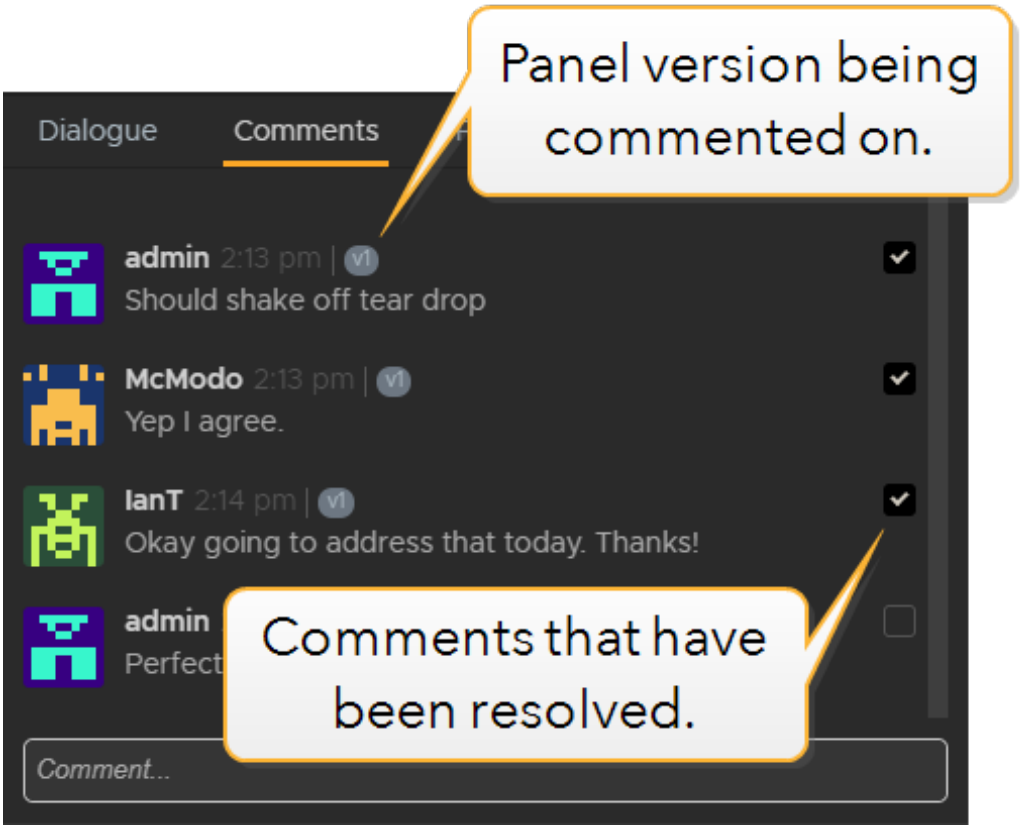
   The panel on which you've commented now displays a comment icon. Hover over the icon to see the latest comment.



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.

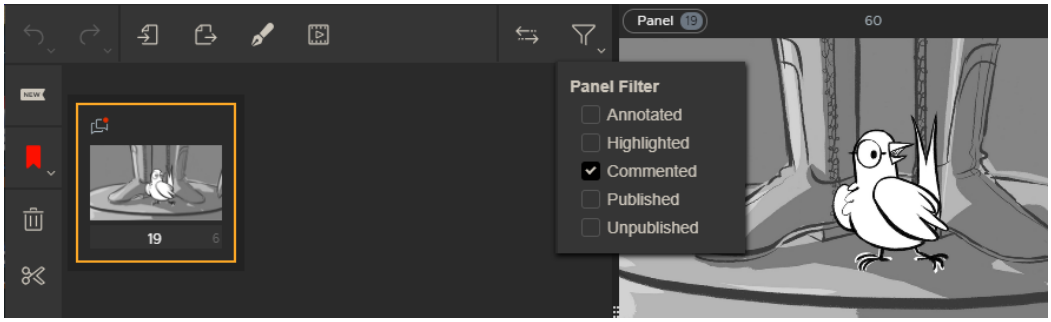
Filtering a Sequence by Comment

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**

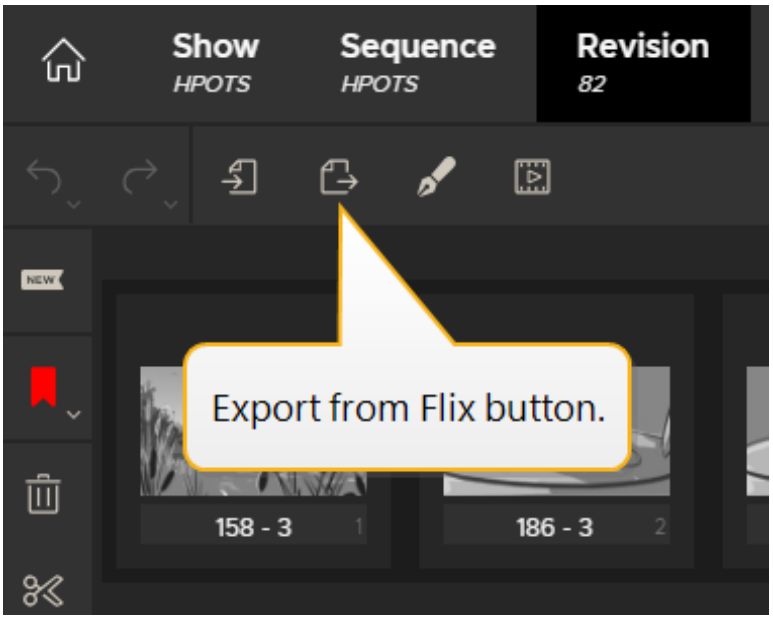
   Flix displays only panels with active comments on them.



Exporting an Edit as a QuickTime

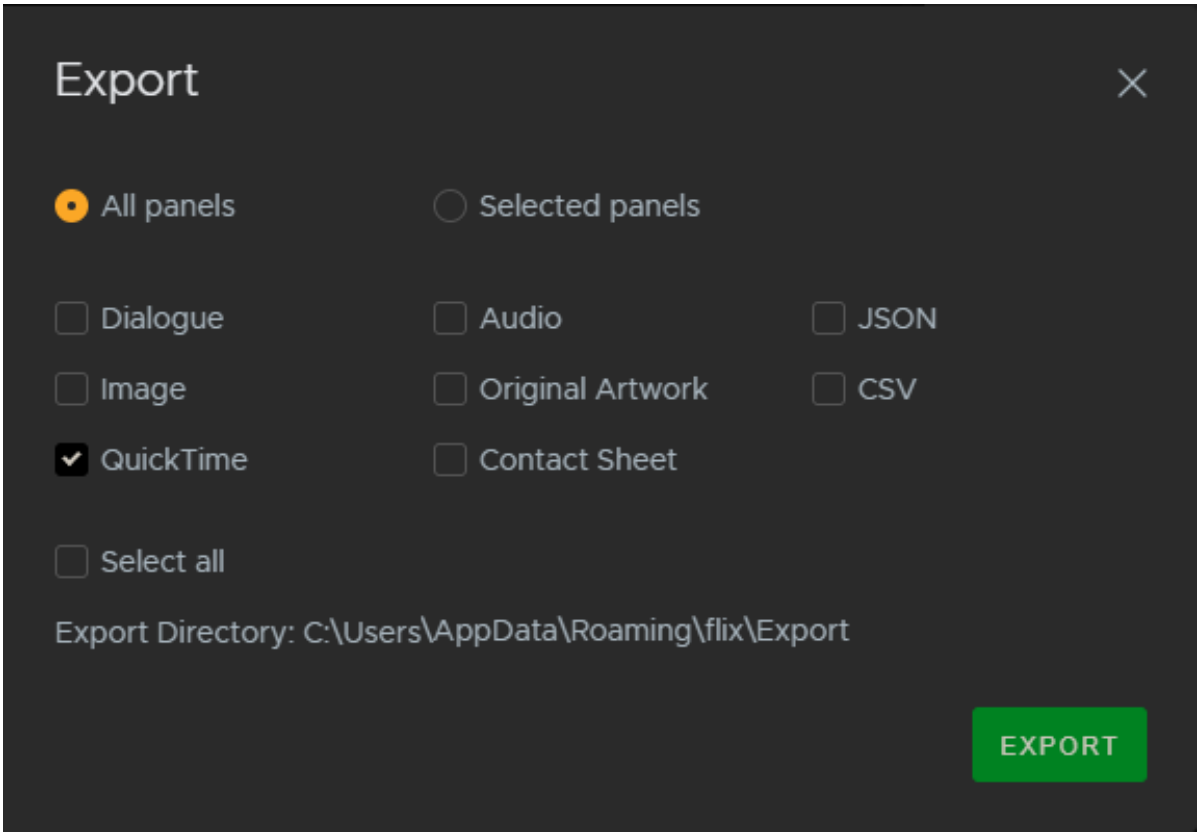
Tip: Before exporting, navigate to **File** > **Preferences** > **Exporting** to set the directory path and the filename structure for your exported files. See Flix Preferences.

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.



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

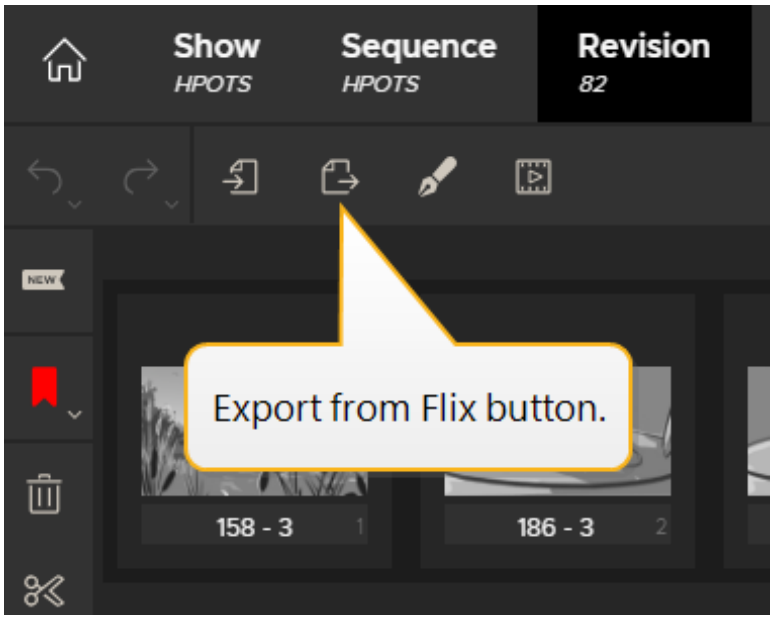
You can also export your dialogue as subtitles in QuickTime if you have the **Include Dialogue** option turned on in **File** > **Preferences** > **Exporting** > **QuickTime Export**.

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

Exporting Panels out of Flix

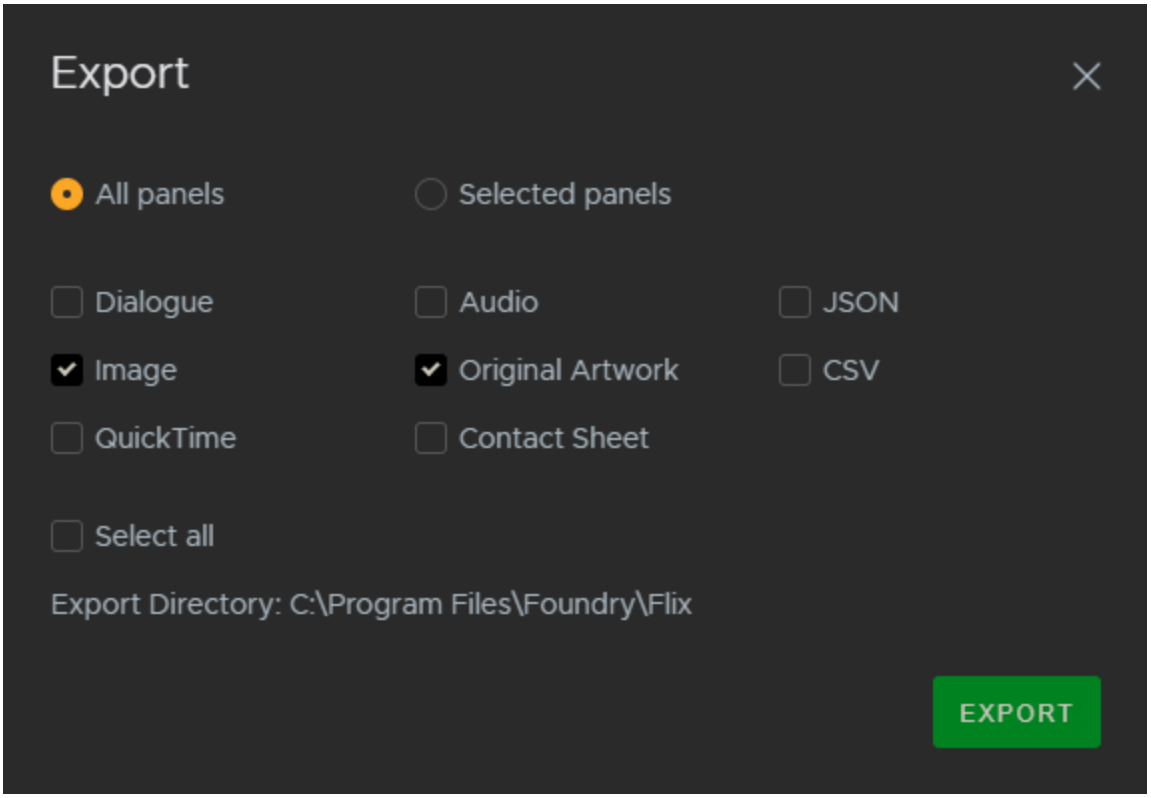
Tip: Before exporting, navigate to **File** > **Preferences** > **Exporting** to set the directory path and the filename structure for your exported files. See Flix Preferences.

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.



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.

Flix for Story

Top Five Things to Onboard New Story Artists

Flix is designed to speed up and manage the various workflows feeding into the creation of a story. Storyboards, Dialogue, Notes from the director, annotations, every version sent and received from Editorial; it's all fed in and managed within Flix. As a Story Artist, you'll be working primarily in your sketching app, like Storyboard Pro. Flix makes your life easier as it tracks every version of a story sequence, acting as the hub that brings together everyone working on the story.

Here are the top five things that will help Story Artists get up and running quickly, so they can get back to focusing on creative tasks.

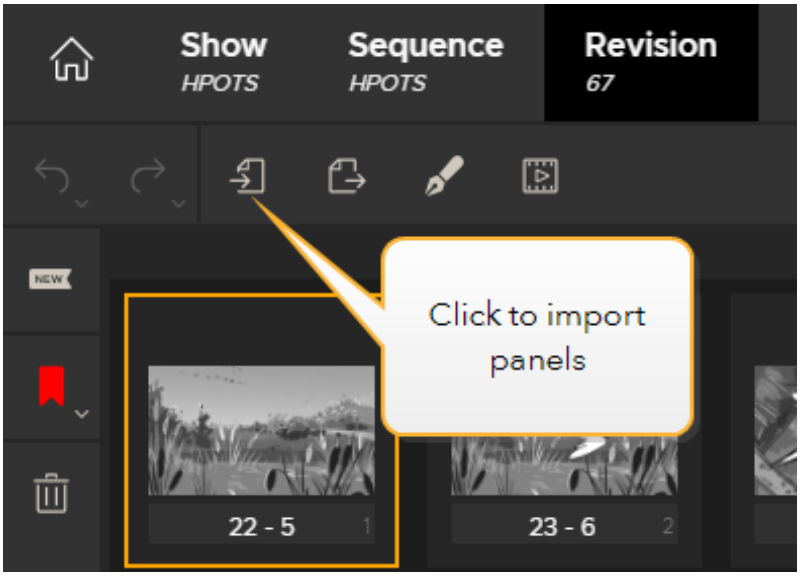
1. Importing Artwork into Flix

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** files from Storyboard Pro, if your artwork has layers you'd like to retain.

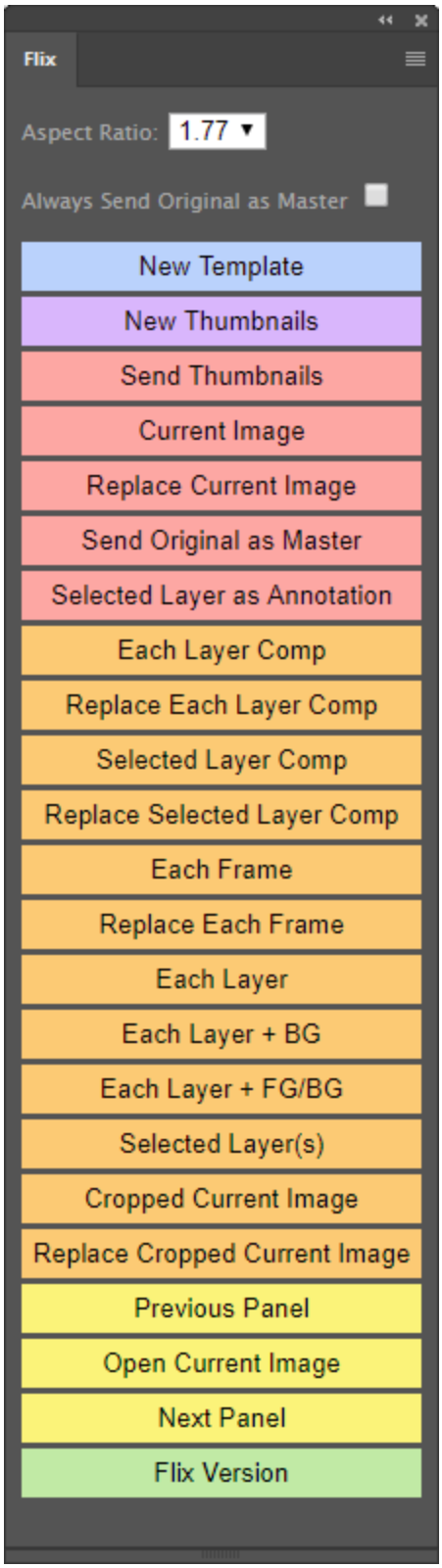
Importing Image Files

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.



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** file. This creates a new sequence revision which contains only panels from the Storyboard Pro file. Panels from other sequence revisions can be found in Flix's Library tab, and drag/dropped into this new revision. See Flix & Storyboard Pro for more information or check out the Flix & Storyboard Pro online video course.

Note: .sboards are the only project file type that Flix supports from Storyboard Pro.

For more information on importing panels and Flix workflows with Photoshop and Storyboard Pro, see Flix for Story. For video tutorials aimed at Story Artists, click here.

2. Panel ID vs Panel Index

What is the difference between a **Panel ID** and a **Panel Index**?

Each panel in Flix is assigned a unique **Panel ID**. This number never changes, so everyone on a production can trust they are always referring to the exact same panel.

Default positions of **Panel ID** and **Panel Index**.

The **Panel Index** is simply the number that indicates the order of panels. It's always sequential. In other words, if you rearrange the order of panels, the indexes will change but the IDs will not.

If you prefer, you can swap the position of the **Panel ID** with the **Panel Index**.

Go to **File** > **Preferences** > **Panel Browser** > **Swap Panel ID with Index** and toggle the switch ON.

The **Panel Index** now appears in the center of the panel and the **Panel ID** is shown on the right.

Panel ID and **Panel Index**in swapped positions.

3. Editing Existing Panels

Throughout the storyboarding process, your director and colleagues will add annotations and comments in Flix to let you know that a panel or sequence revision needs changes. You can just open up the board straight from the sequence in Flix, rather than digging through folders on your local machine to find the original.

To edit the original Photoshop file:

1. Make sure Photoshop is your default sketching app. To check this, go to **File** > **Preferences** > **Third-Party Apps** and check the **Sketching Tool** is set to Photoshop.
2. In Flix's panel browser, select the panel you would like to edit in Photoshop.
3. Double-click the panel or use the **Open in Sketching App** button.

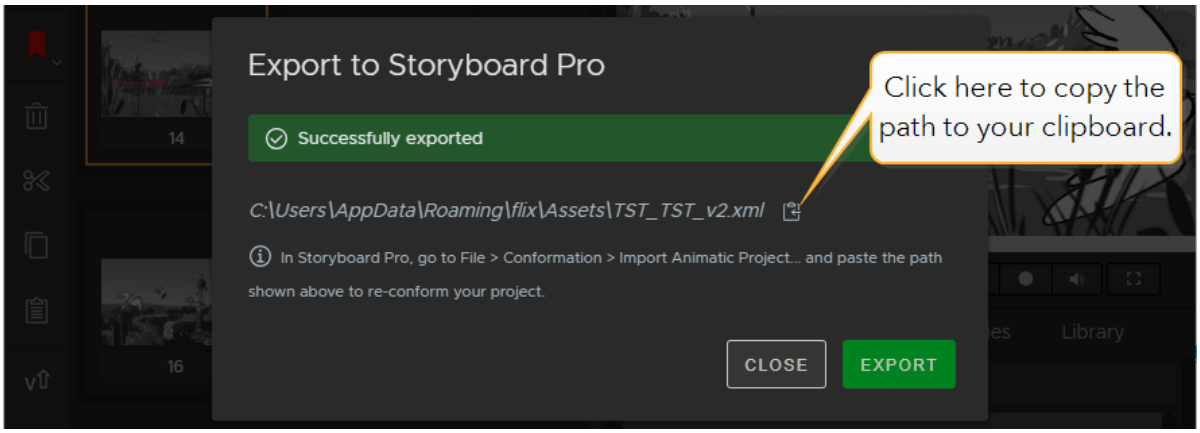
   The master Photoshop document opens in Photoshop.

Note: The **Current Image** option in the Flix/Photoshop toolset sends a flattened **PNG** to Flix, so layers and layer comps won't have been retained when that panel is reopened in Photoshop.

To send existing panels to Storyboard Pro for further editing:

1. Make sure Storyboard Pro is your default sketching app. To check this, go to **File** > **Preferences** > **Third-Party Apps** and check the **Sketching Tool** is set to Storyboard Pro.
2. Select the panel in Flix.
3. Double-click the panel or use the **Open in Sketching App** button.

   Flix prompts you to export the full sequence revision as an **XML** file.
4. Click **Export**.



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

Switching Panel Versions

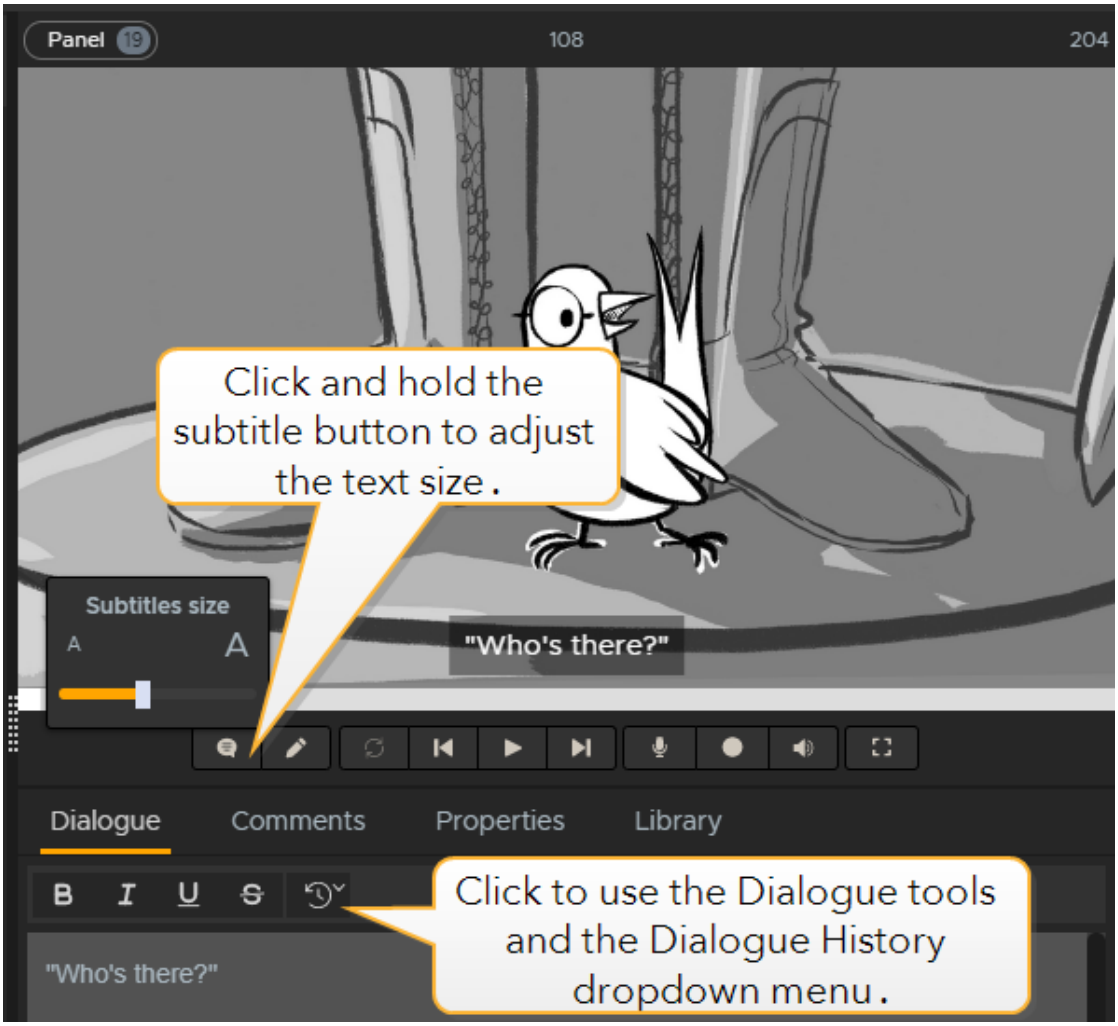
Each time you make changes to a panel within Flix, your previous Flix edits are still available to you as panel versions. If it turns out that your team prefers the previous iteration of a panel, just click on the space at the bottom of the panel and select the desired version from the **Revisions** 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.

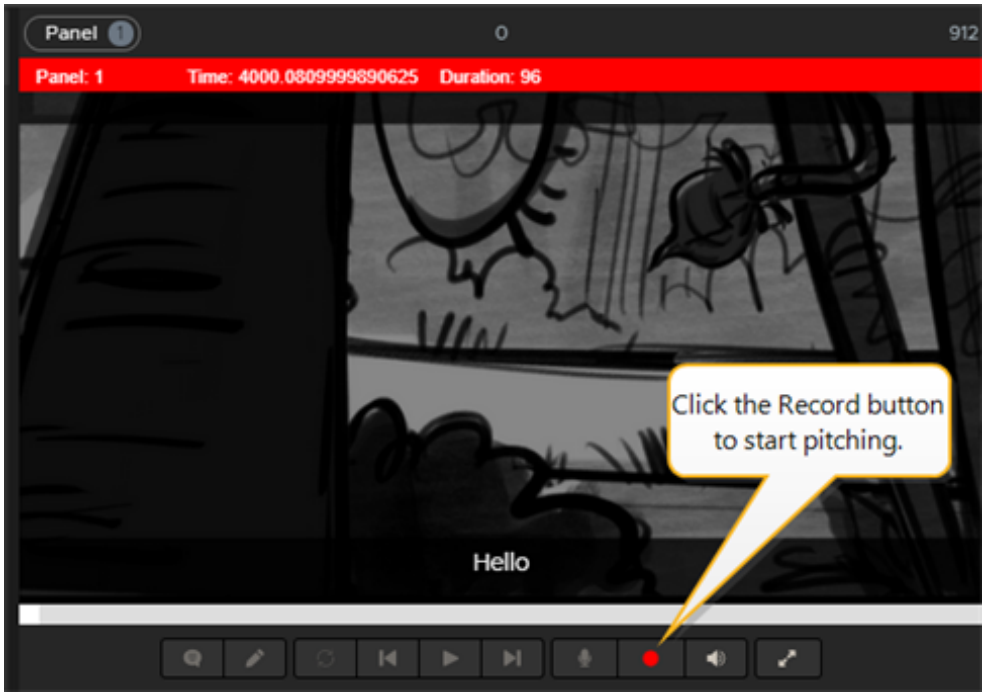
For more information on working with dialogue in Flix, see Adding or Editing Dialogue.

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.

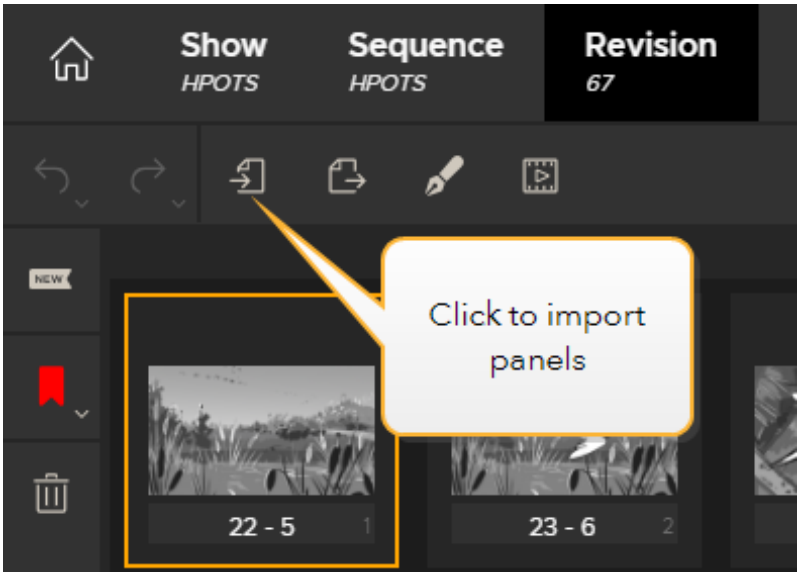
If you prefer to pitch the sequence the traditional way, you can also simply step forward or backward through the sequence, using either your keyboard arrows or the Go-to buttons.

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**.

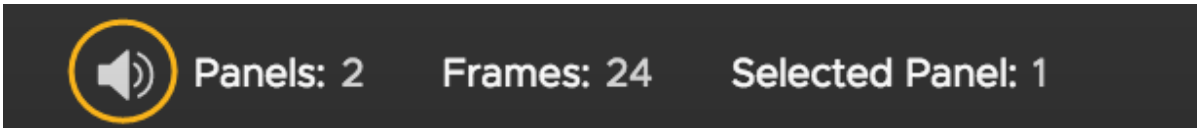
   OR

   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.

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.

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 **Duplicate** button.

Editing Panels Already in Flix

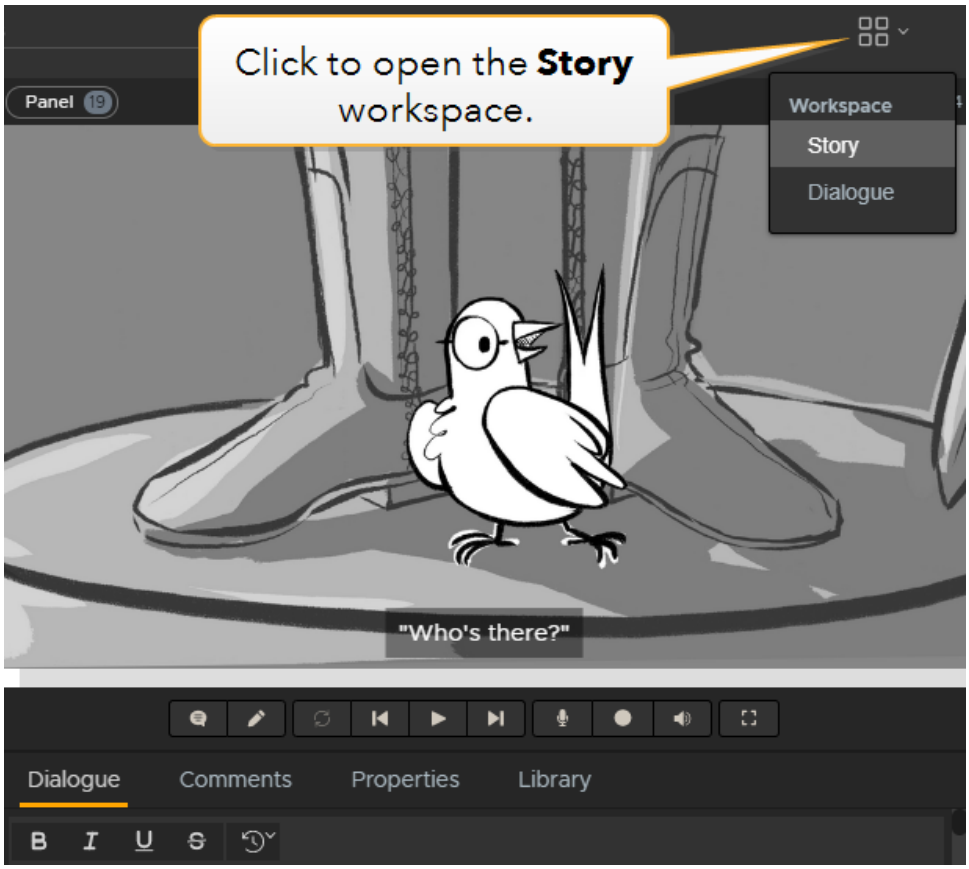
To make changes to panels in your edit (sequence revision), you can open and edit them in third-party applications. See the Flix & Photoshop section.

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.

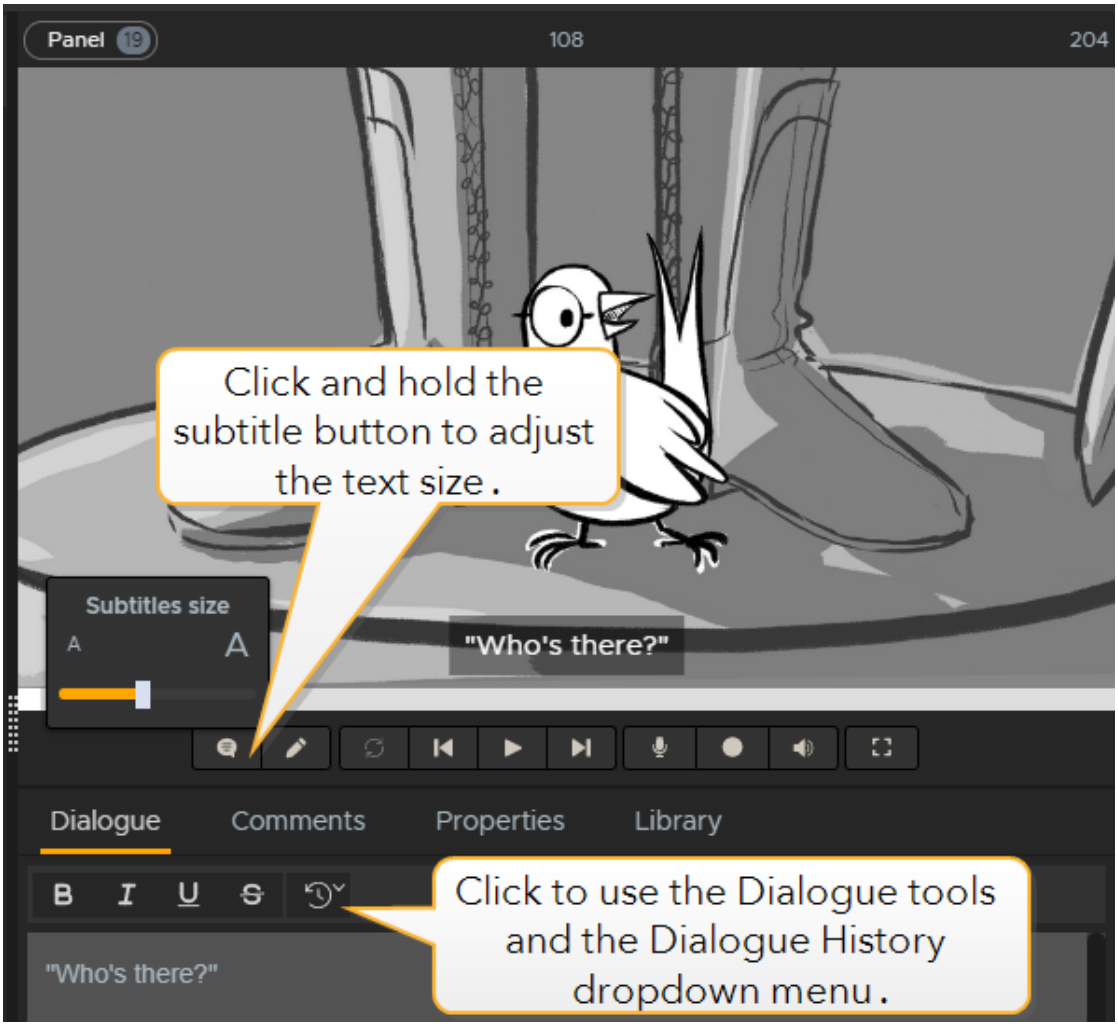


To add and edit dialogue:

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.

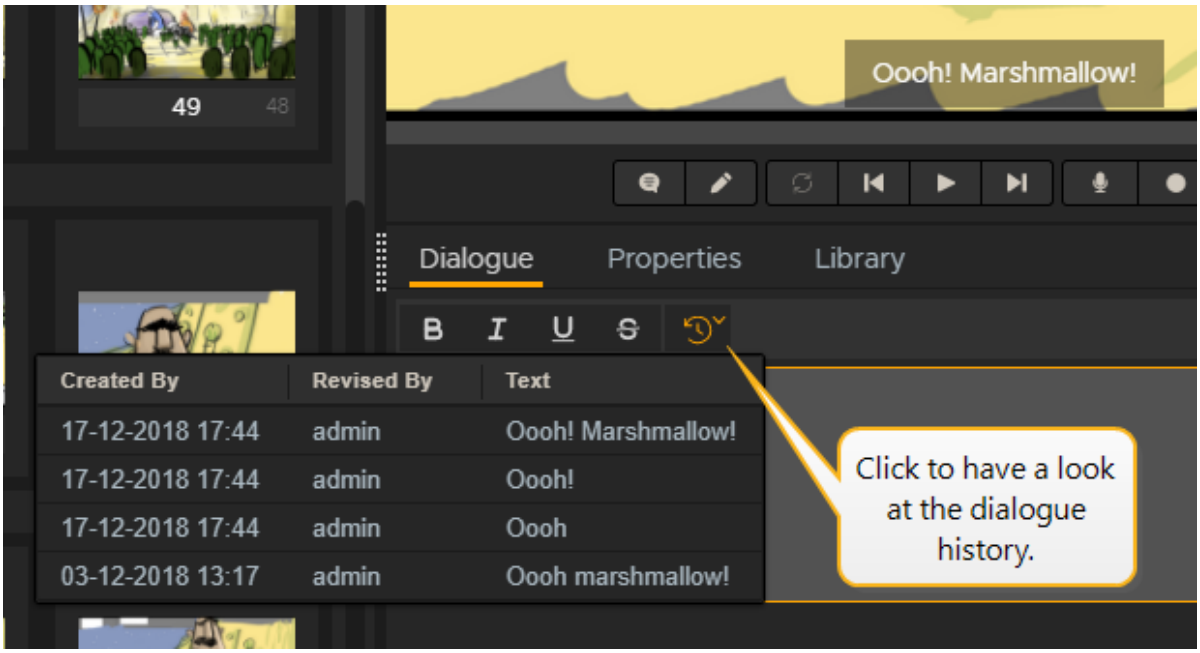
Note: If you select multiple panels, the same dialogue is added to all selected panels.

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.

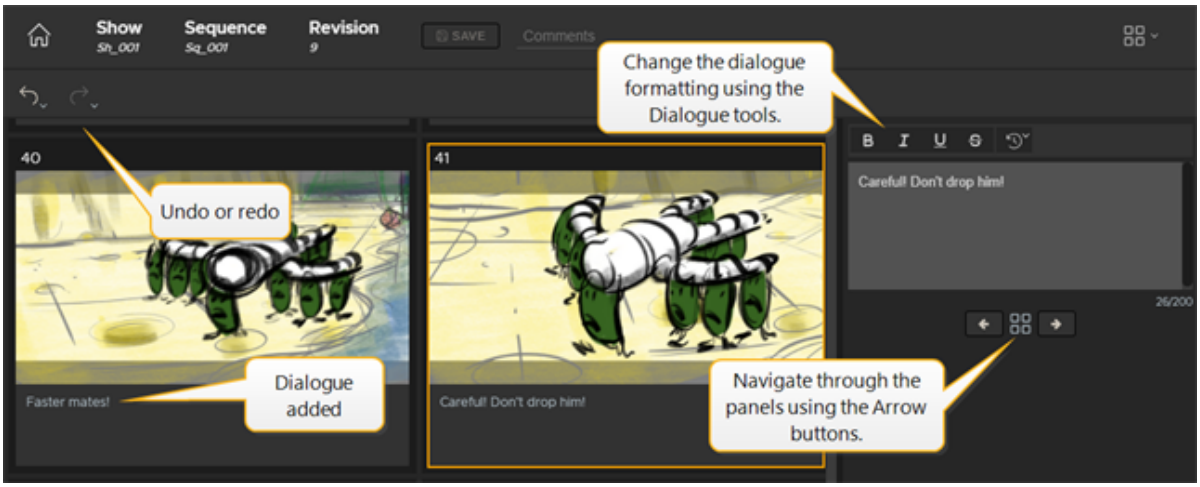
To add and edit dialogue:

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.

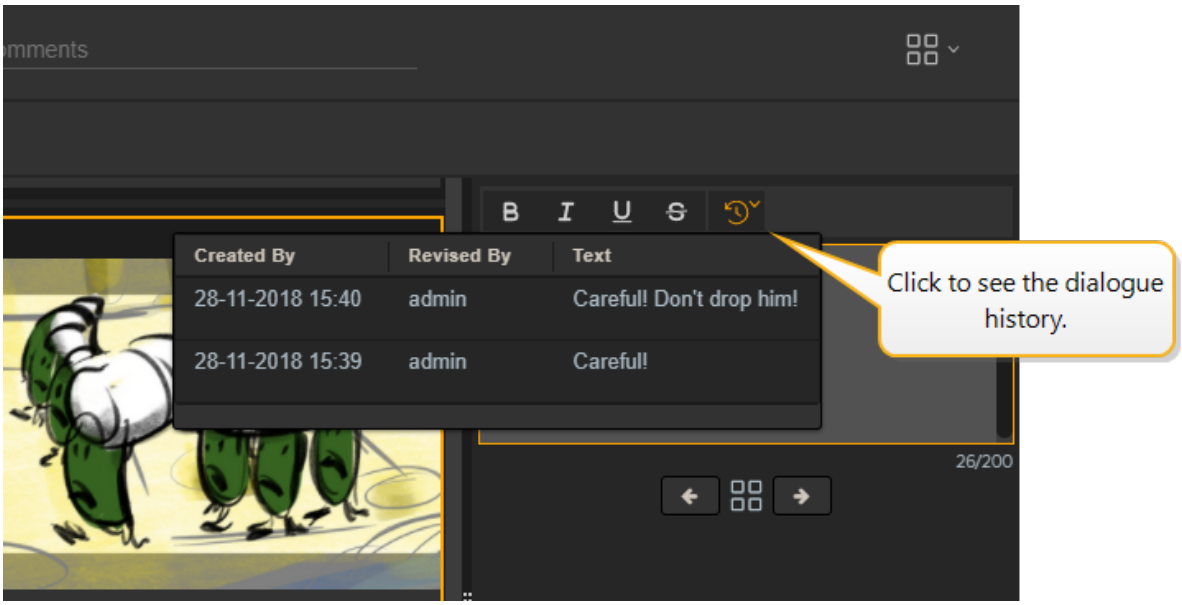
Tip: Press **Tab** to move to the next panel and **Shift+Tab** to move to the previous panel when entering dialogue on the Dialogue workspace.

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.

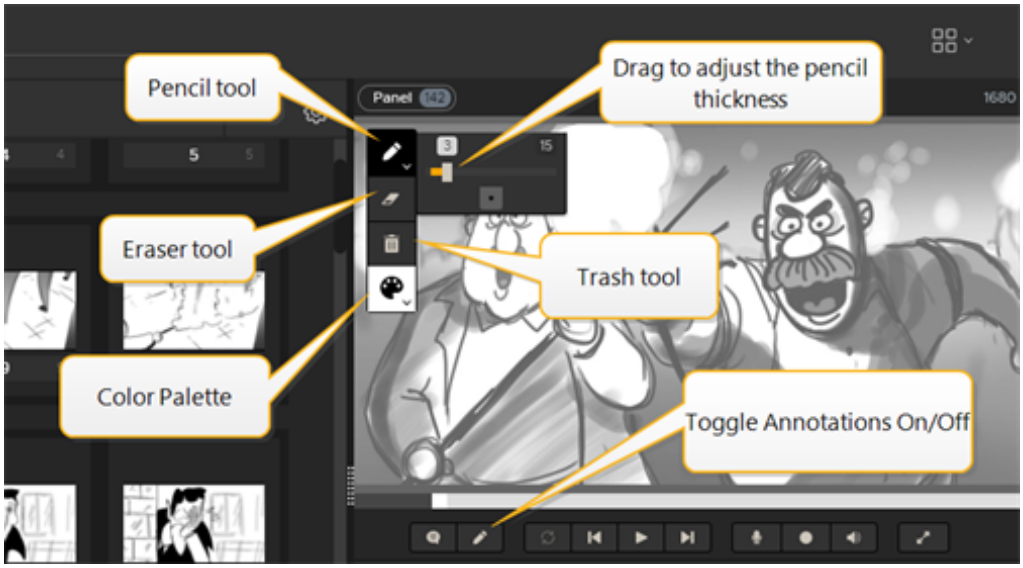


Annotations

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.

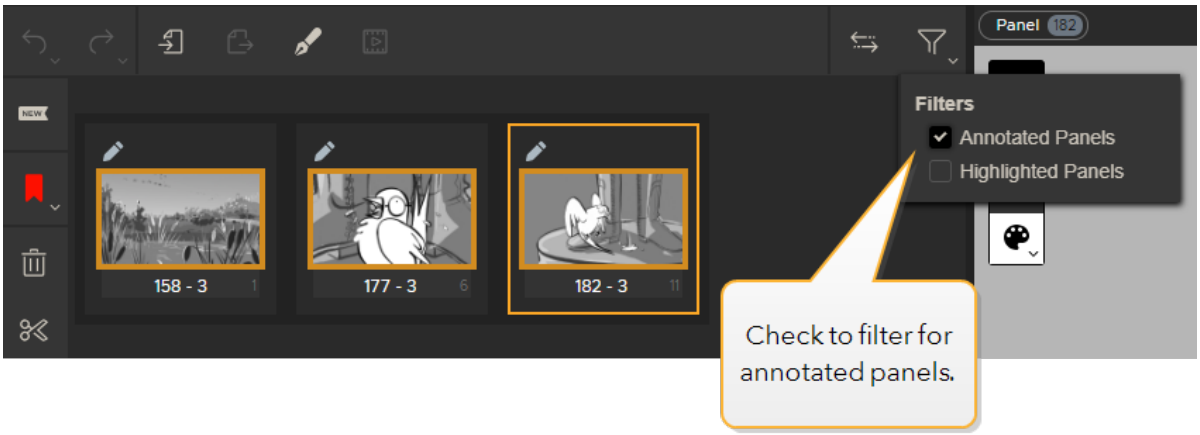
1. In the Panel Properties pane, toggle the annotations on by clicking the Pencil button at the bottom of the Player.



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 button and enable the **Annotations** filter.



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

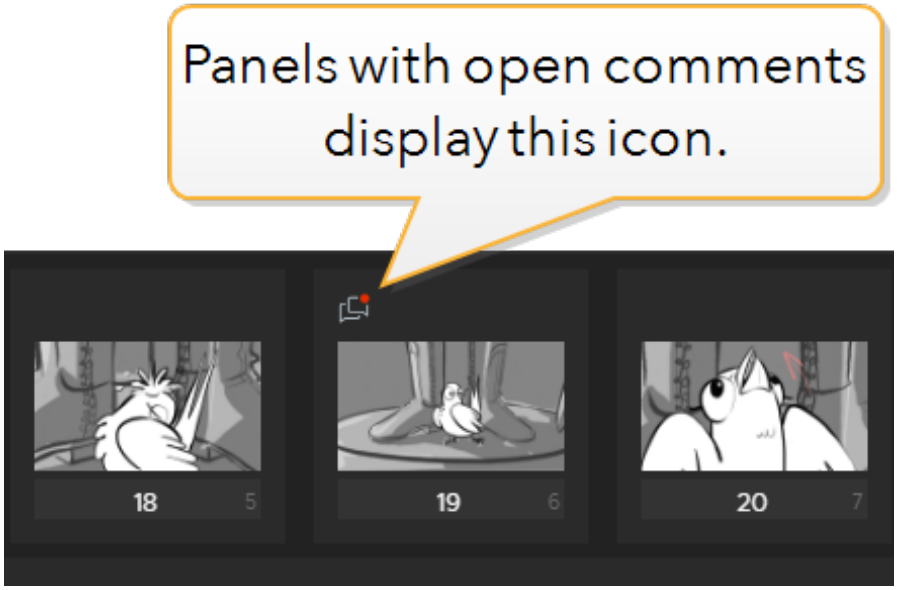
Panel Comments

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.

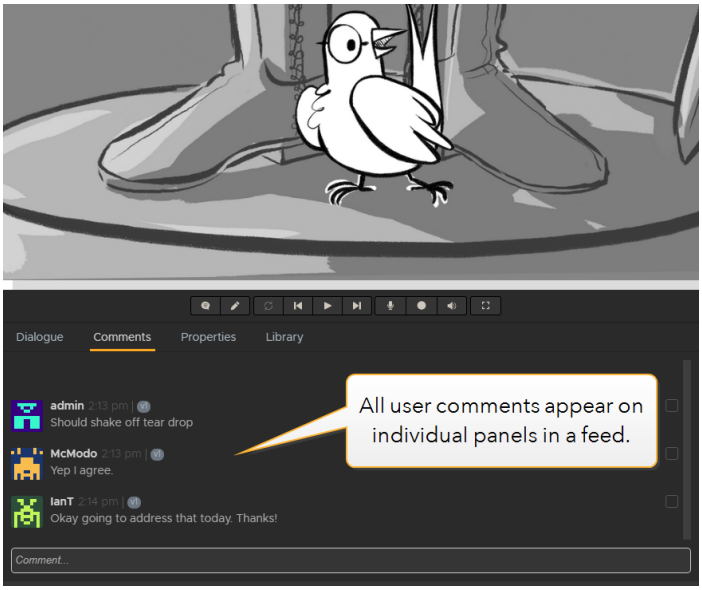
To add a comment:

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

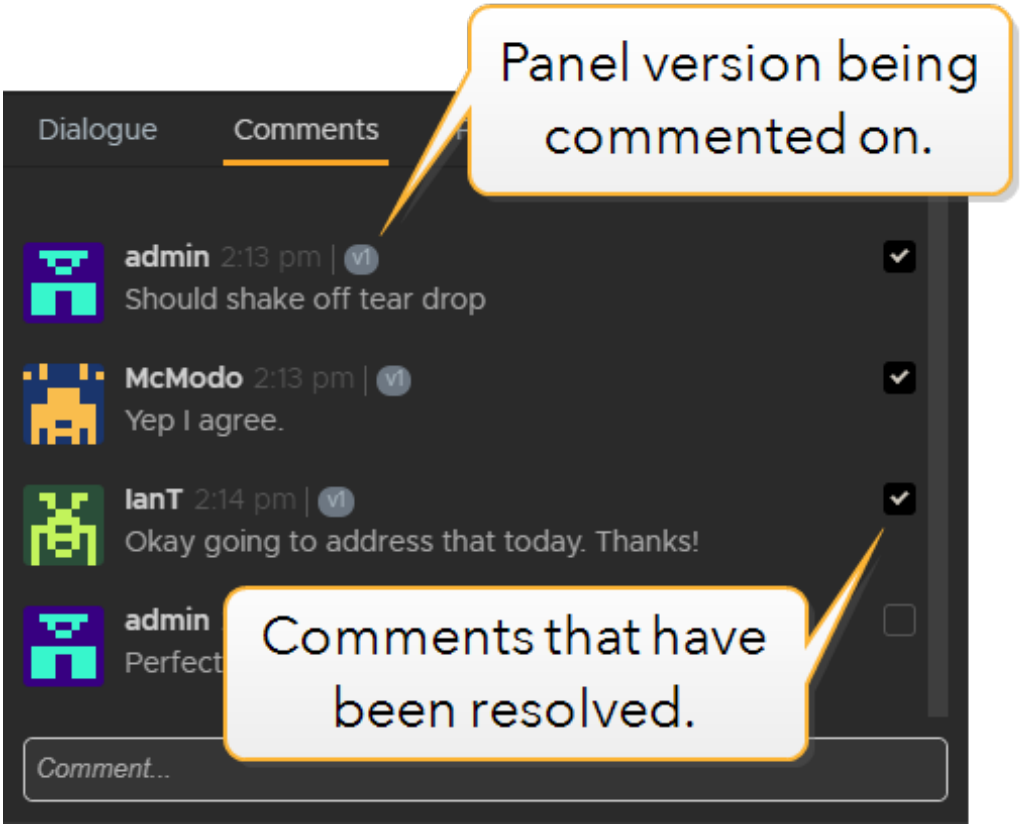
   The panel on which you've commented now displays a comment icon. Hover over the icon to see the latest open comment.



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.



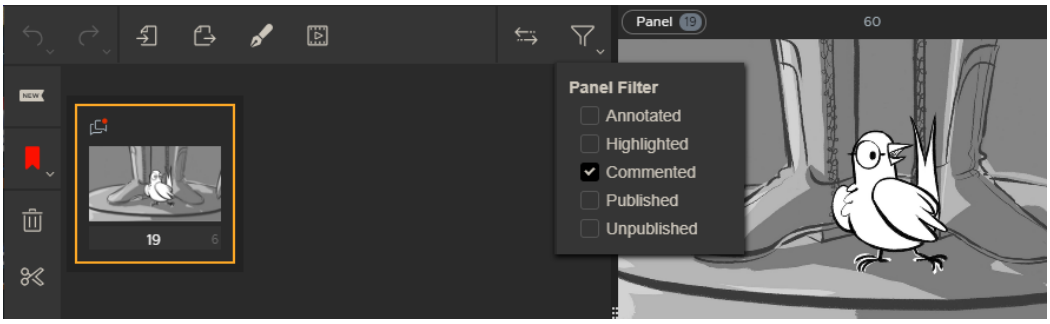
Note: Once all the comments in a feed have been resolved, the comment icon disappears from the panel.

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 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.

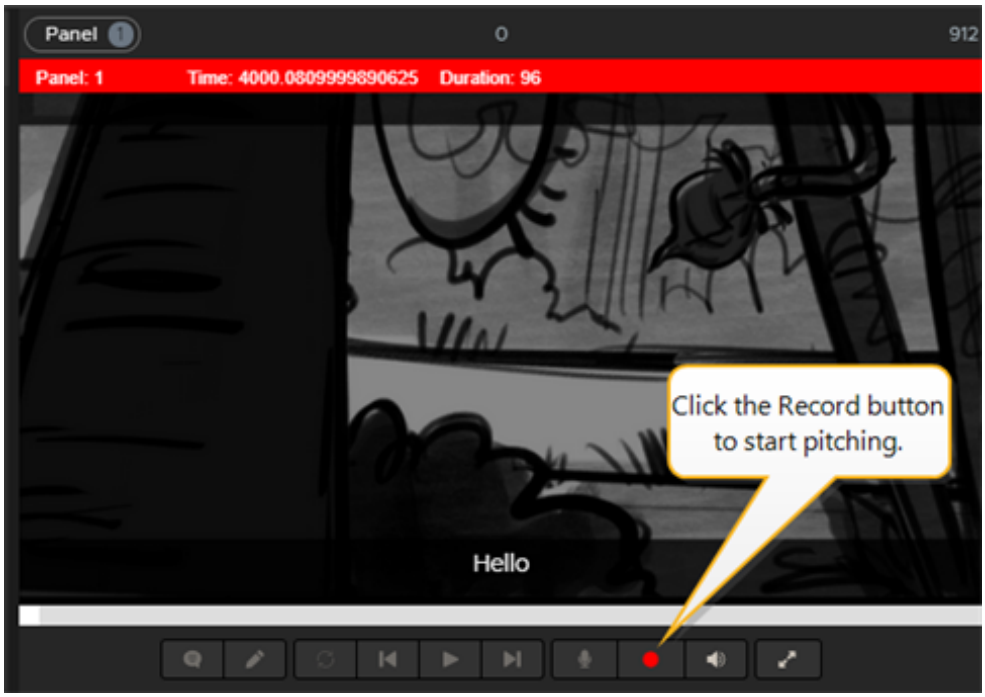
1. If you want to record audio while pitching, navigate to **File** > **Preferences** > **Audio** and select an **Input Device**. If you already have audio in your sequence and don't want to override it, in the Player click the Microphone button to disable it.

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** > **Preferences > Third Party Apps.**
3. In the **Sketching** tab, select **Photoshop.**
4. In the **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 Plugins**. This installs the Photoshop scripts to run the Photoshop actions.

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**.

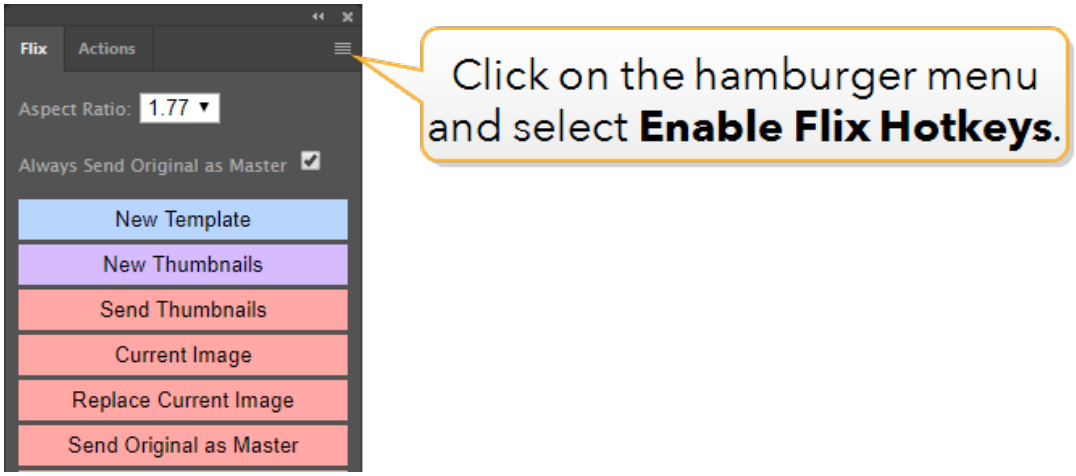
Assigning Keyboard Shortcuts to Flix Commands

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.

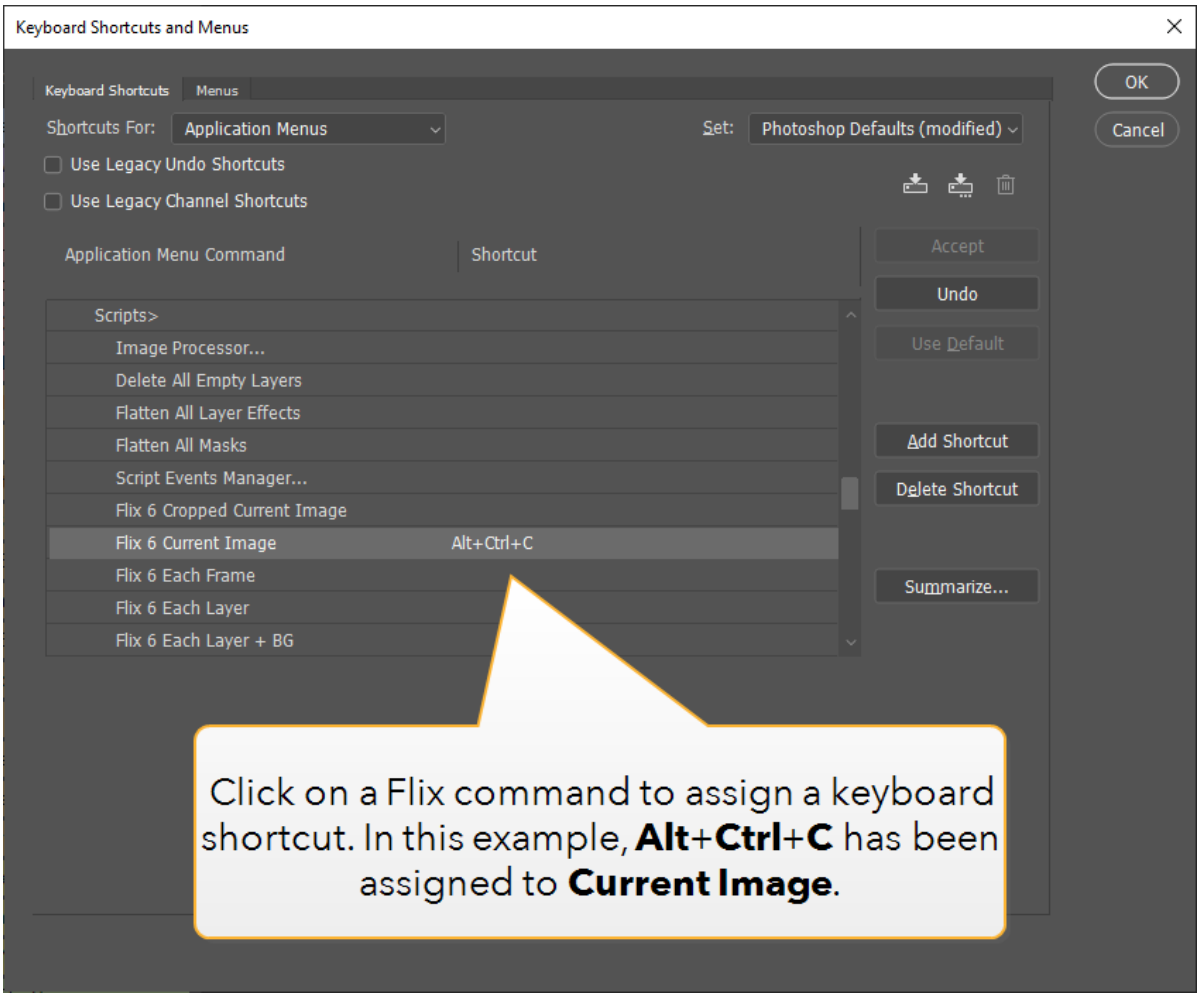
5. Restart Photoshop.

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

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

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).

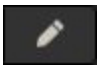
2. Click **New Thumbnails**.
3. Draw on the thumbnails.
4. Click **Send Thumbnails**.

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. In the **Sketching** tab, 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 6\StoryboardPro.exe*.

- Under the **Import** section, choose the behavior for importing Storyboard Pro project files containing **Camera Moves**:
    - **Hold First Frame** - Holds the first frame to render a still panel.
    - **Render All Frames** - Renders all the frames to create an animated panel.

> **Note:** Hold First Frame imports are quicker than imports with Render All Frames set. Storyboard Pro exports PSDs for each panel, which Flix renders on import. If the animation of a panel is not required, choose Hold First Frame, as Flix will only render one frame per panel rather than all frames.

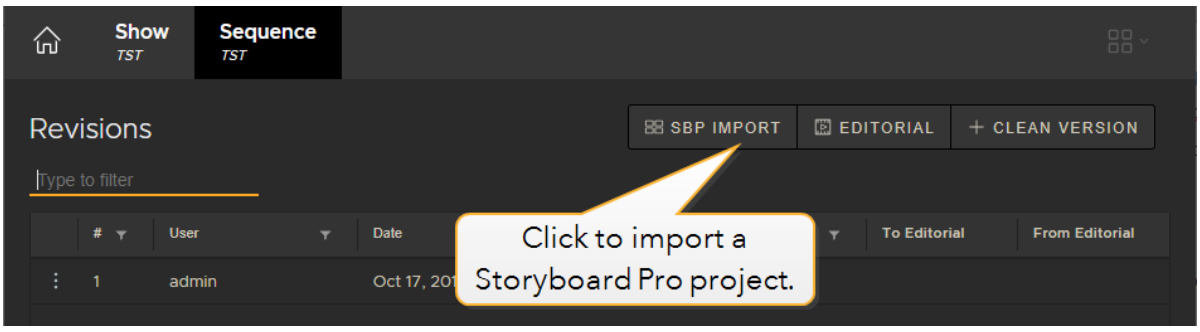
- Under the **Export** section, choose the behavior for exporting Flix sequences to Storyboard Pro.
    - **Export Path** - Set the path for your exported Flix sequence.
    - **Export Dialogue** - Toggle ON to update dialogue in Storyboard Pro.
    - **Export Camera Moves** - Toggle ON to update Camera Moves in Storyboard Pro.
    - **Export Audio** - Toggle ON to include audio from Flix in exports to Storyboard Pro.
    - **Export Markers** - Marker updates are not currently supported in Storyboard Pro.

Importing Storyboard Pro Projects into Flix

You can create a new sequence revision by importing your Storyboard Pro project directly into Flix.

To import your Storyboard Pro project into Flix:

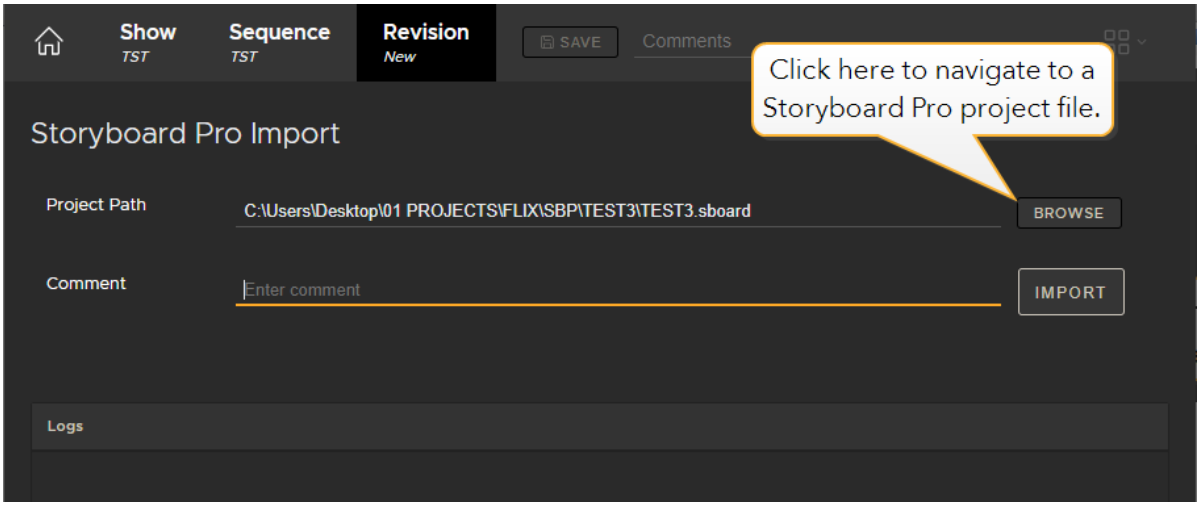
1. Save your Storyboard Pro project.
2. In Flix, navigate to the Revisions page, either by clicking on your sequence or creating a clean version.



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**) file you want to import. Select the file and click **Open**.
   The path to the project file appears in the **Project Path**.



5. 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 a few moments.

6. Click on **Sequence** in the breadcrumb to return to the Revisions page.
   A new sequence revision has been created from your Storyboard Pro import.

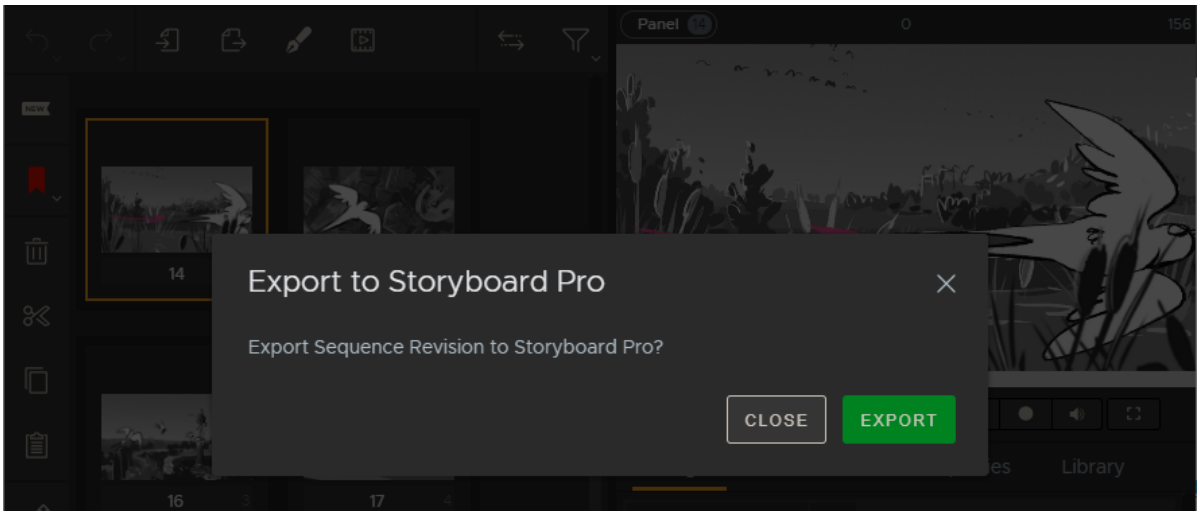
> **Note:** If markers are set to display in the panel browser, scene numbers show as markers on sequences imported from Storyboard Pro.

> **Article:** Dialogue within Storyboard Pro 20 projects does not transfer across to Flix. To learn more about this issue and possible workarounds, check out this Knowledge Base Article.

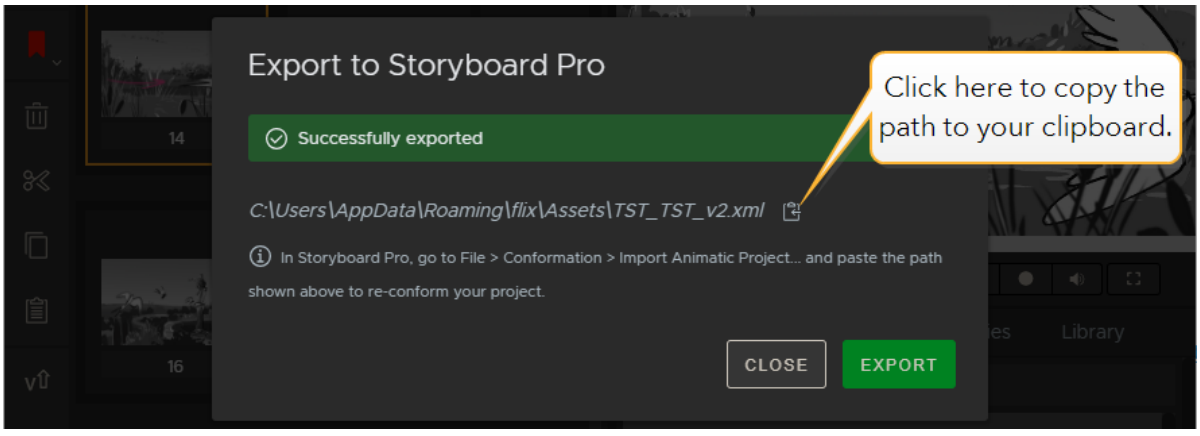
Exporting Flix sequences to Storyboard Pro

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



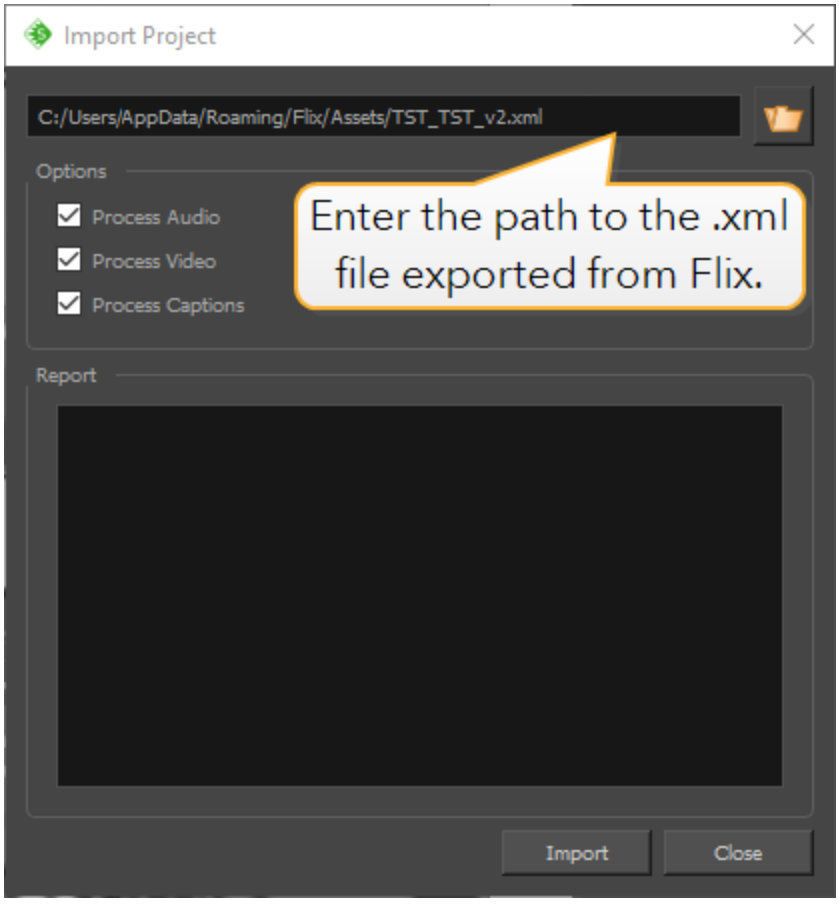
Note: To change the default path for exports, go to **File** > **Preferences** > **Third Party Apps** > **Storyboard Pro** > **Export**.

Re-conforming Your Project in Storyboard Pro

When changes come back from editorial and Flix, you can always update your project in Storyboard Pro and continue working. First, you will need to reconform your original Storyboard Pro project, following these steps.

1. In Storyboard Pro, go to **File** > **Conformation** > **Import Animatic Project**.

   The **Import Project** menu opens.



2. Enter the path to the **.xml** file exported from Flix, or click the  icon and browse for the file. Once located, click **Open**.

> **Tip:** You can copy the path from the Export to Storyboard Pro menu in Flix and paste it in the Import Project menu. See Exporting Flix sequences to Storyboard Pro for more information.

3. 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 export naming conventions.
4. Click **Okay**.

   Your Storyboard Pro sequence updates with any changes that came from Flix.

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 will 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 Timeline or Clip markers to Premiere. • **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.  **Article:** Read the Using Premiere Markers in Flix Knowledge Base article for more information on how markers are used in 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 Media Composer 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 Avid Media Composer.
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 **Avid Media Composer** tab, set your preferences for exporting **.aaf** files to Media Composer. See the table below for details.

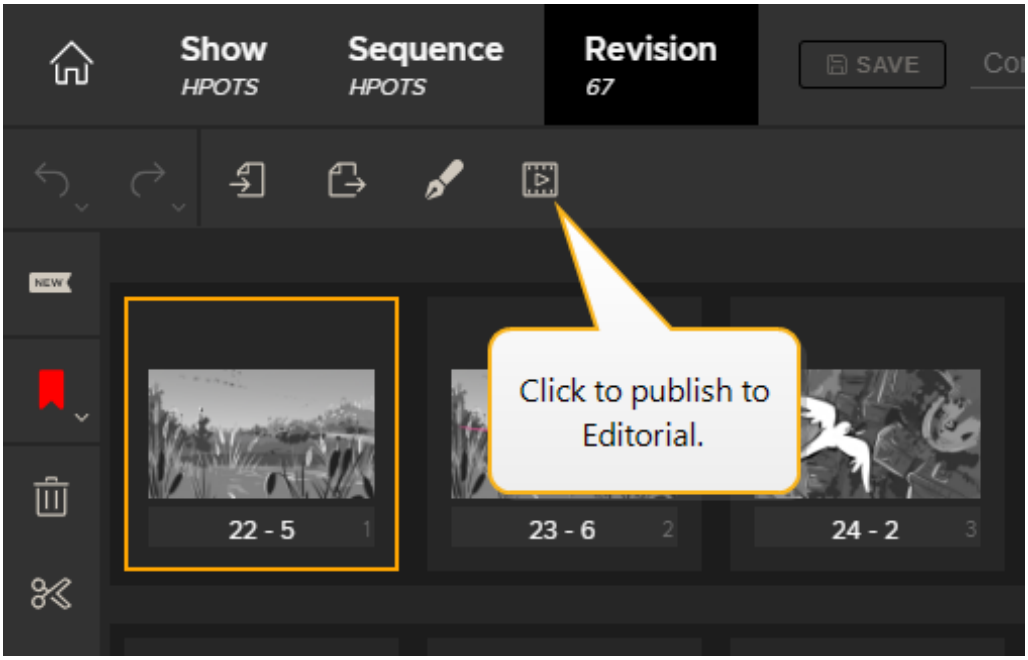
Avid Media Composer	
Avid Clip Name	Specifies the naming convention of the embedded clips in AAFs exported for Media Composer. **Note:** The chips located under the Avid Clip Name preference show what information type you can include in the filename format. Click a chip to add it to your filename convention.
Codec	Specifies the type of Avid video codec used for exporting to Media Composer. Choose between DNxHD 36 for smaller file size and DNxHD 115 for higher bitrate, larger files.
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.
Enable Mark In	Toggle ON to include a Mark In point on each clip.
Mark In	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	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.
Send Animated Panels As Stills	When enabled, a still is sent to Avid instead of a movie file, resulting in a faster publish. **Note:** This does not apply to animated panels originating from Storyboard Pro, which are always sent as stills so editors have control over keyframes.
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.

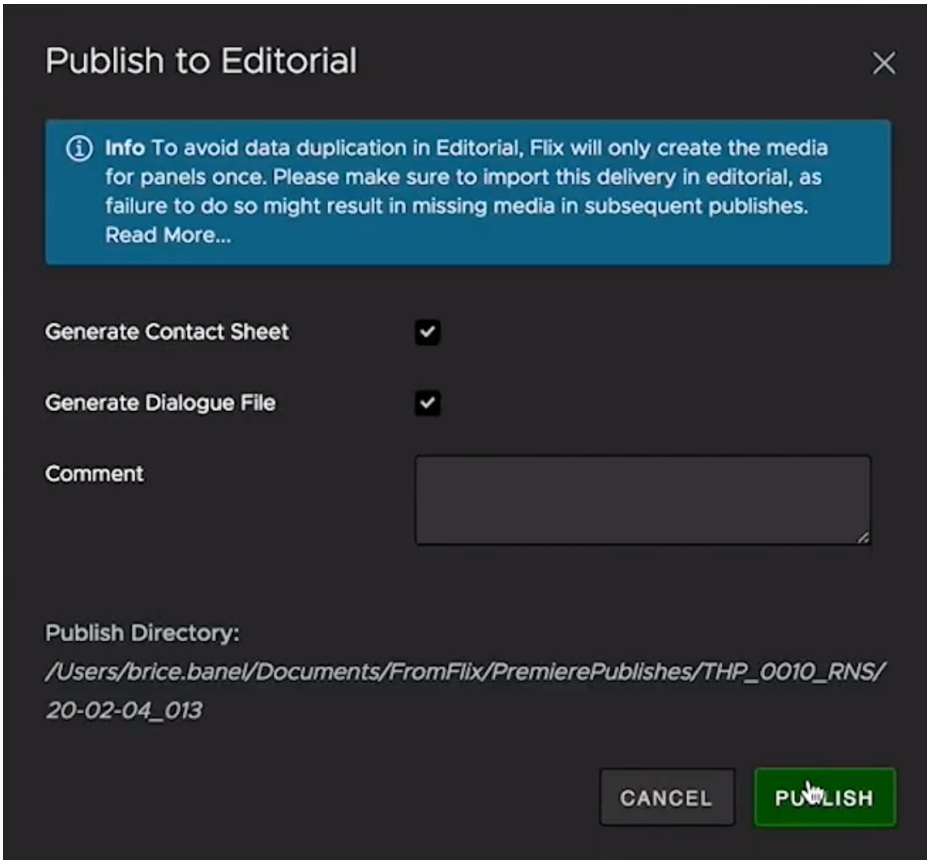
Publishing a Sequence to Editorial

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

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

   - <filename>**_all.xml/.aaf** - This file contains the panel information of your entire sequence as well as any new or existing audio.
   - <filename>**_new.pdf** - This file (Contact Sheet) contains the panel thumbnails of your entire sequence with **New** labels on panels that have been added or edited since the last publish. The Contact Sheet also contains dialogue. This is only created if you enabled **Generate Contact Sheet**.
   - <filename>**_new.xml/.aaf** - This file contains only information and audio of panels that have been added or edited since the last publish.

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

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

Filtering a Sequence by Published/Unpublished Panels

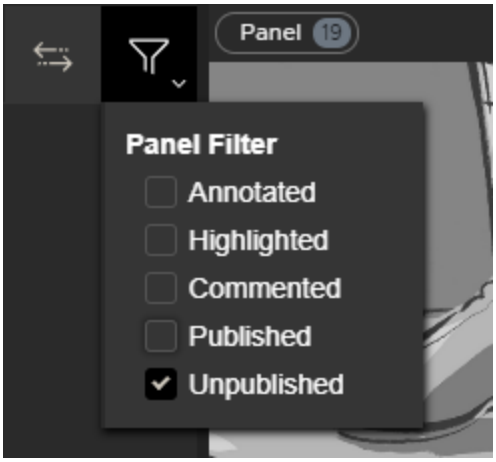
To check which panels have previously been included or excluded from publishes to editorial:

1. Click on the  Filter button in the Panel Browser.
2. Either:

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

OR

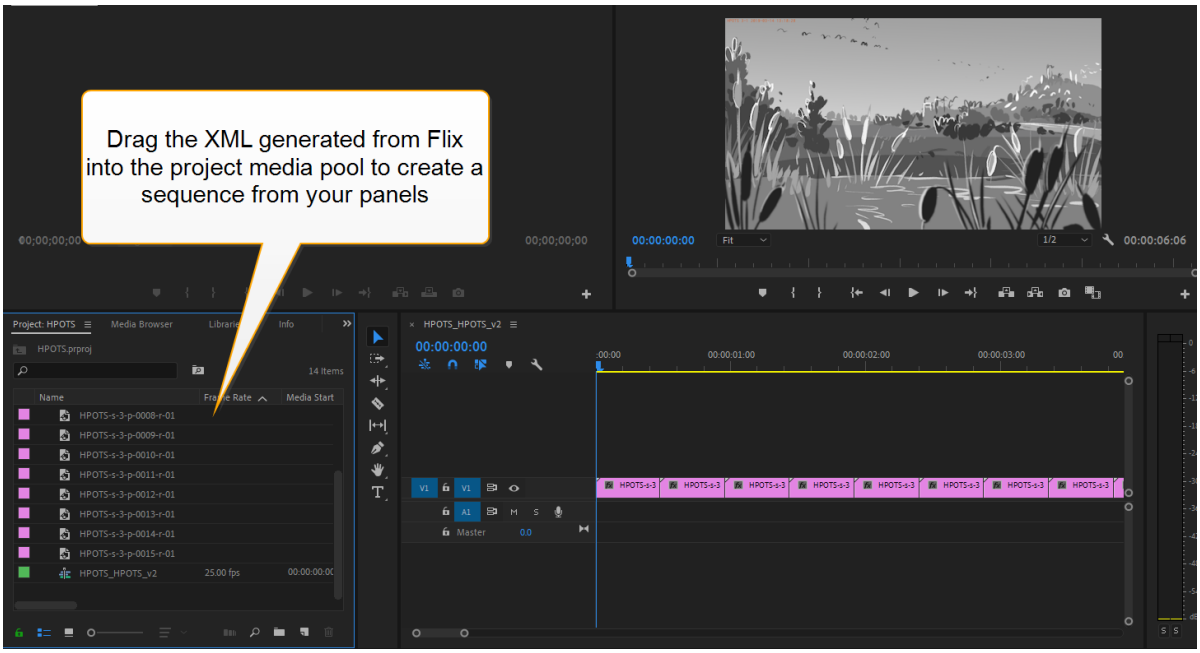
• Check the **Unpublished** option to see only panels that have not been published to editorial previously.



Importing Your Sequence to Adobe Premiere

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

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

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.

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

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.

Article: Have a look at this Using Premiere Markers in Flix Knowledge Base article for more information on how markers are used in Flix.

Exporting Your Sequence Revision from Avid Media Composer

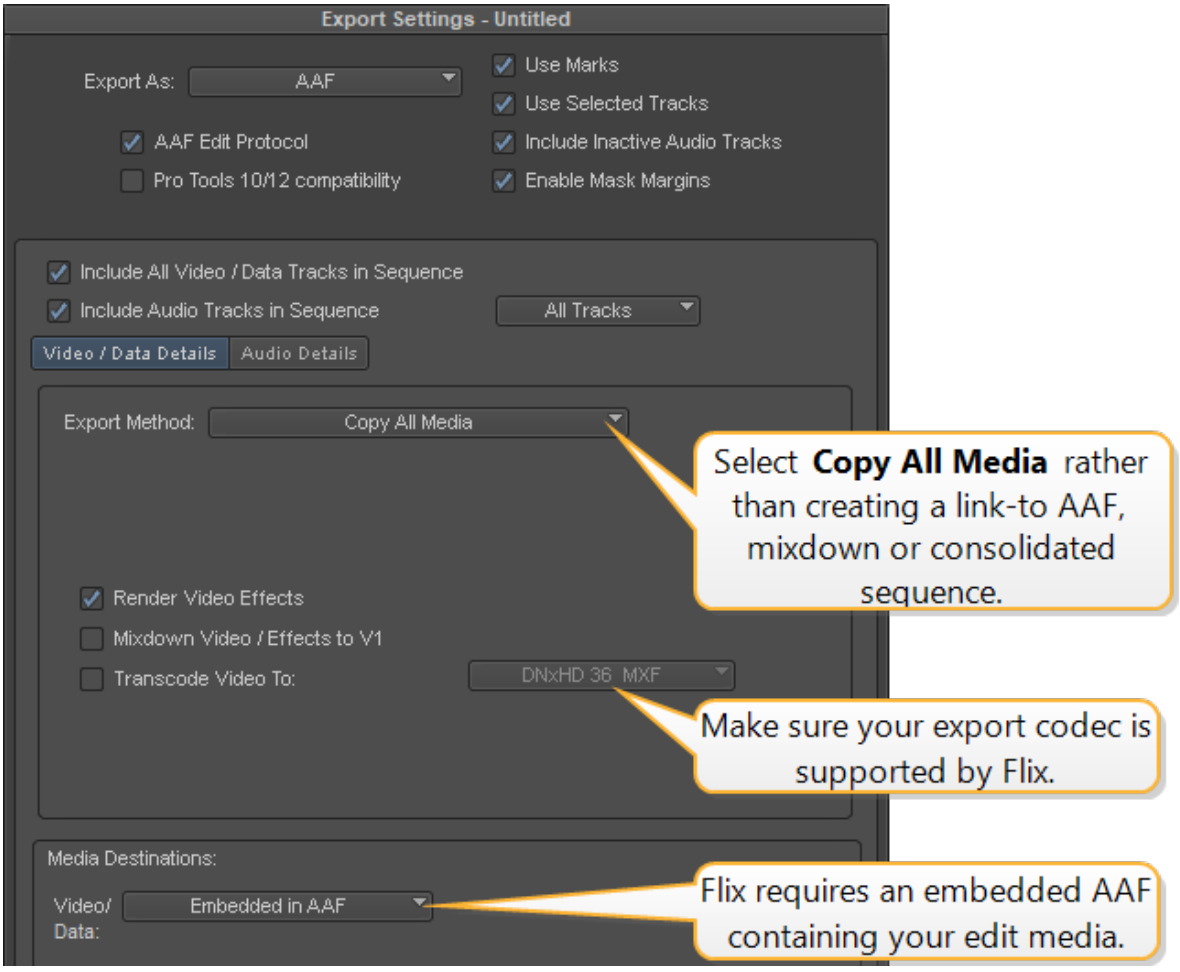
To export an **AAF** of your edit back to Flix:

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.

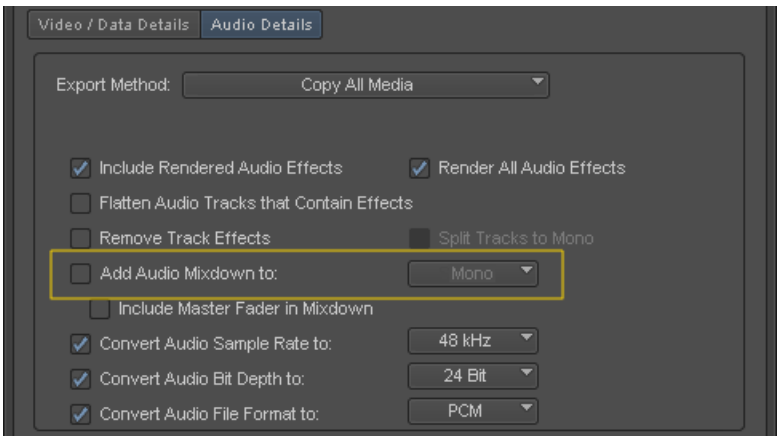


The **Audio Details** tab controls how audio is exported from Avid. Audio can severely impact the file size of AAFs exported from Avid to Flix.

You can:

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

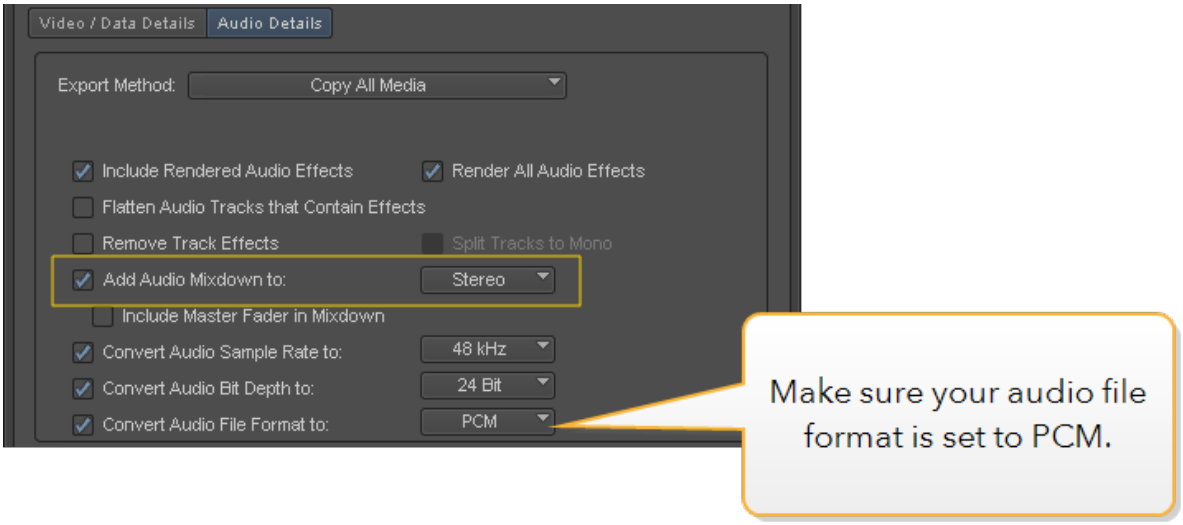
Note: We recommend manually mixing down audio in Avid before exporting the AAF back to Flix. See the Audio Mixdown in Avid Knowledge Base article (Q100581) for more information.



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.

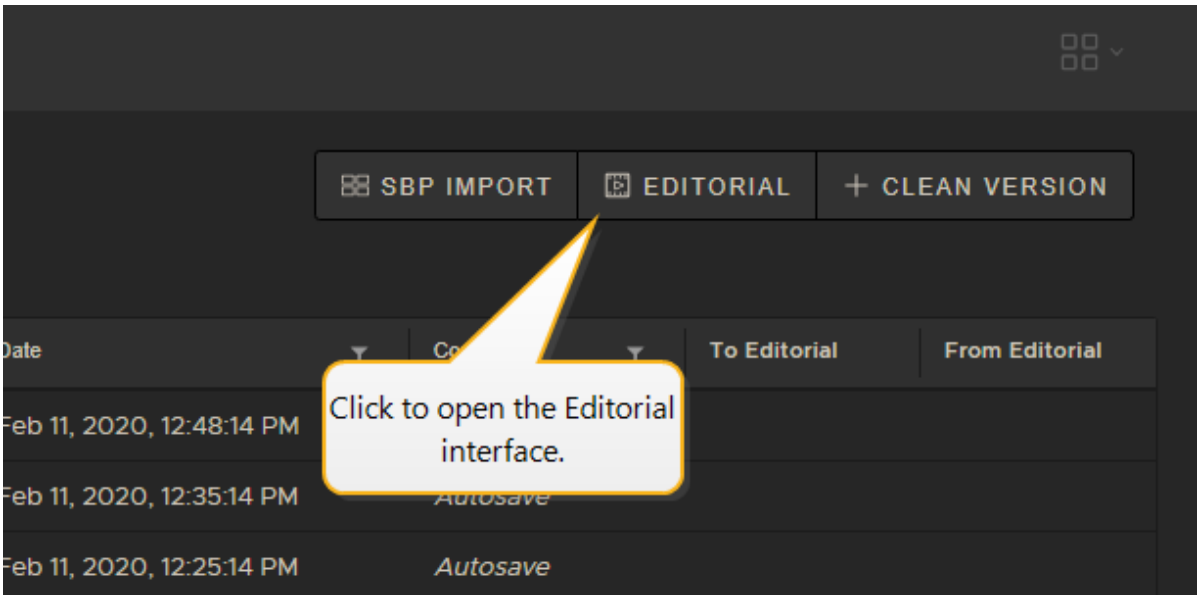


3. Click **Save** to finalize the video and audio export settings.
4. Choose a location to save your file and click **Save**.

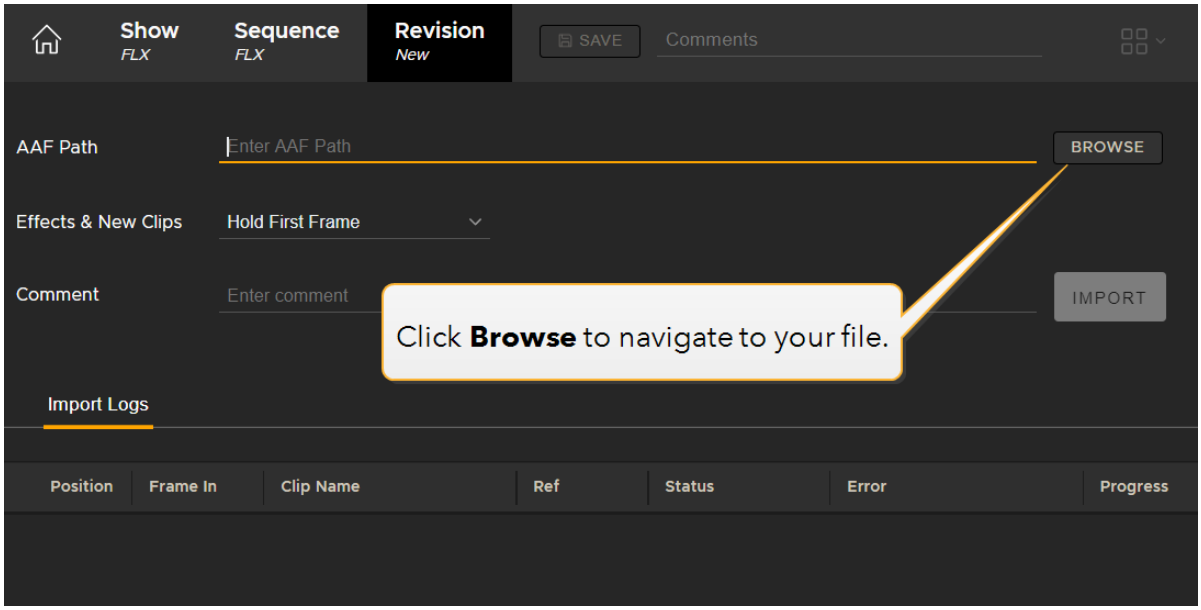
Note: Markers created in Avid Media Composer display in the Flix panel browser at the beginning of the corresponding shot.

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.

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 - Port number for the Flix Server to run on, e.g. 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.

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 toMigrating 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)

Authentication

LDAP (Lightweight Directory Access Protocol)

For guidance on formatting, please refer to the example at the end of this section.

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.

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 the search for the User starts. If this value is not set the **base** will be used.

filter (optional) - Filter to apply when searching the directory. You can filter by any attribute. 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 the search for the Group starts. 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.

Example of a LDAP subsection in a 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
  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
```

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).

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.

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.

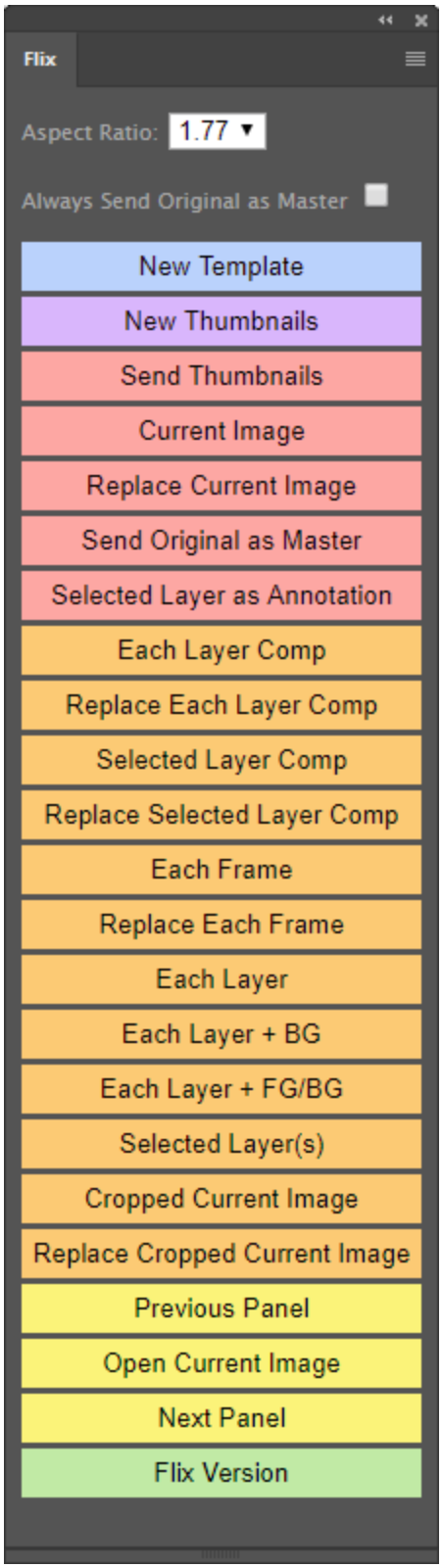
Note: Flix supports Adobe Photoshop CC 2019 and CC 2020. See Third-Party Application Support for more information.

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
Aspect ratio	Specifies the aspect ratio to set for the panels that are sent to Flix.
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 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.

Action	Description
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.

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.

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:

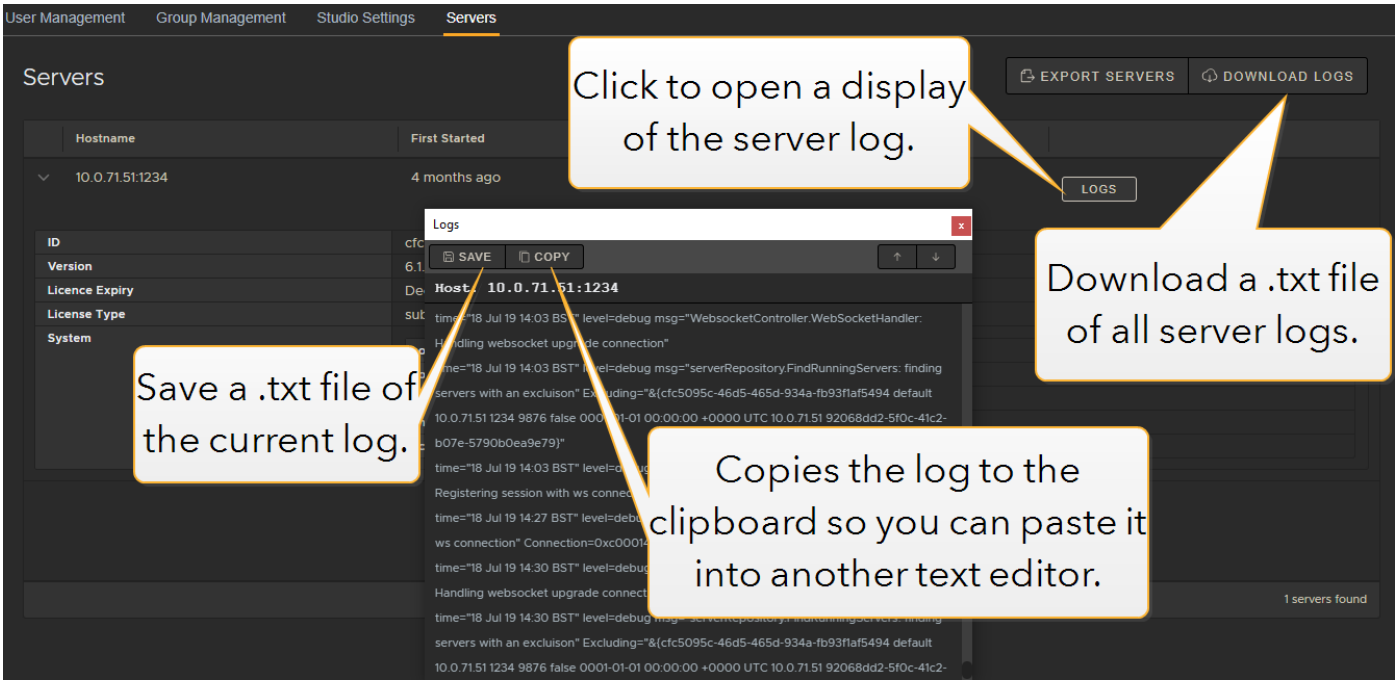
```
log_thread_info:
    user: true
    client_id: true
    server: false
```

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 Tool	Specifies the external application used to edit panels.

	Note: In releases prior to Flix 6.2, Photoshop is the only supported editor.
Editorial Tool	Specifies the external application used for editorial roundtripping.
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.

	Tip: Environment variables can be used to construct publish and export paths. For example, **%USERPROFILE%** on Windows and **${HOME}** on macOS.  **Note:** Flix will automatically create missing directories if they don't already exist.
Adobe Photoshop	
Executable	Specifies which version of Photoshop 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,

	meaning that it opens the **.psd** file with all information including hidden layers and empty groups.  **Note:** This is only relevant if a master image is available.
Send Annotation as Layer	When enabled, annotated panels will open in Photoshop with their annotations shown on an additional layer.
Install Plugins	Installs the Photoshop scripts to run the Photoshop actions. Restarting Photoshop is required after installing the Flix plugins.  **Article:** To remove the Flix 6 extension from Photoshop, see this Knowledge Base Article.
Adobe Premiere	
Marker Type	Choose whether Flix sends Timeline or Clip markers to Premiere. • **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.  **Article:** Read the Using Premiere Markers in Flix Knowledge Base article for more information on how markers are used in 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.
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.
Enable Mark In	Toggle ON to include a Mark In point on each clip.
Mark In	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	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 creates an unreadable AAF.
Send Animated Panels As Stills	When enabled, a still is sent to Avid instead of a movie file, resulting in a faster publish.

	Note: This does not apply to animated panels originating from Storyboard Pro, which are always sent as stills so editors have control over keyframes.
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.
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.
Import	Choose the behavior for importing project files containing **Camera Moves**: • **Hold First Frame** - Holds the first frame to render a still panel. • **Render all Frames** - Renders all the frames to create an animated panel.
Export	Specify the behavior for exporting Flix sequences to Storyboard Pro. • **Export Path** - Set the path for your

	exported Flix sequence. • **Export Dialogue** - Toggle ON to update dialogue in Storyboard Pro. • **Export Camera Moves** - Toggle ON to update Camera Moves in Storyboard Pro. • **Export Audio** - Toggle ON to include audio from Flix in exports to Storyboard Pro. • **Export Markers** - Marker updates are not currently supported in Storyboard Pro.

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

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

User Data Directory	Specifies where Flix stores your preferences, logs and assets.
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	Removes the temporary files stored in the local asset cache.

Keyboard Shortcuts

Project

Open the Preferences	**Ctrl**/**Cmd**+**,** (comma)
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.

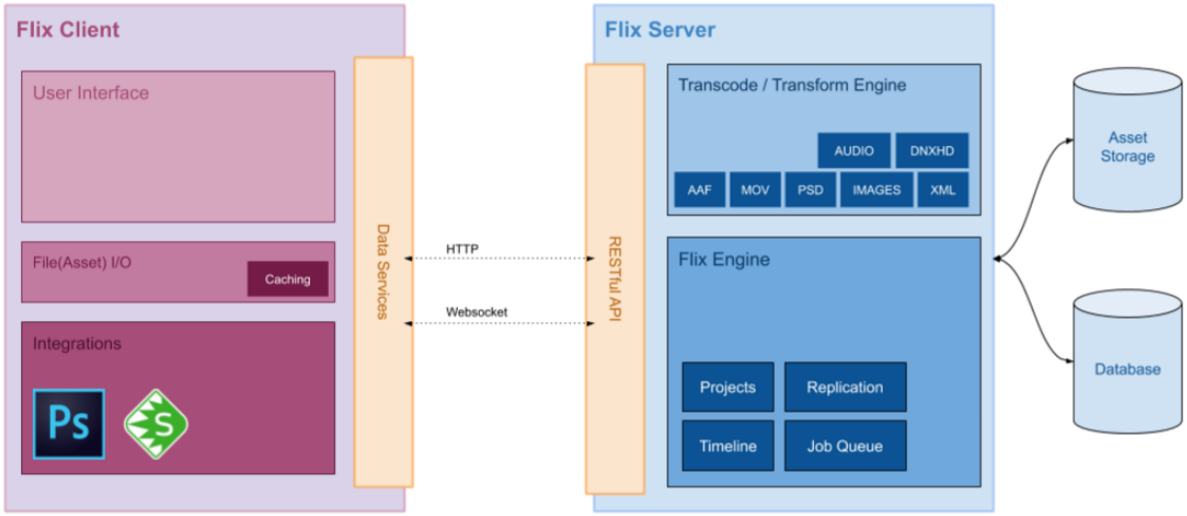
This overview assumes some familiarity with systems administration and client/server architecture.

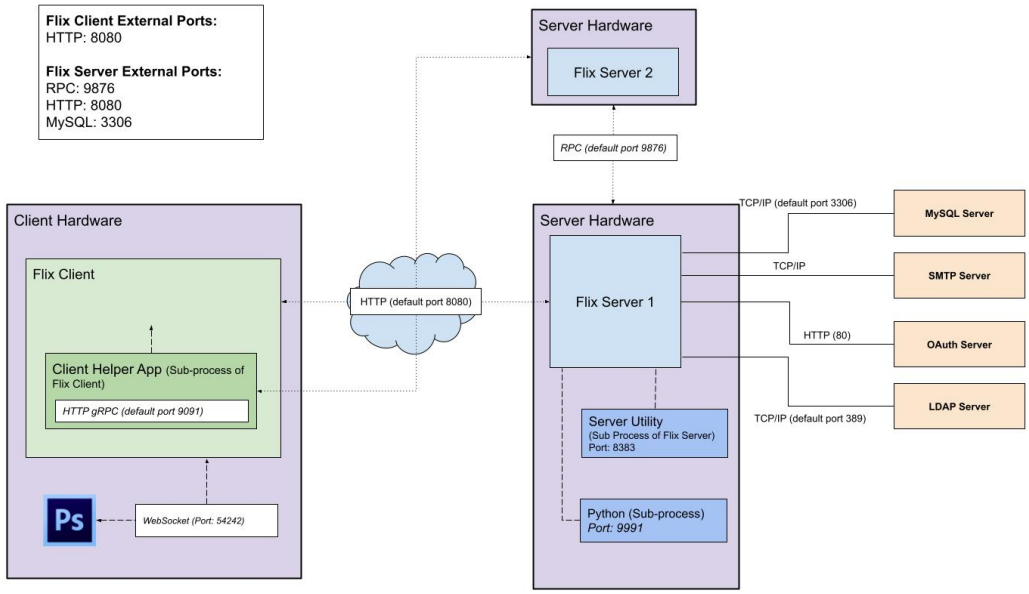
Architecture

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.

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.

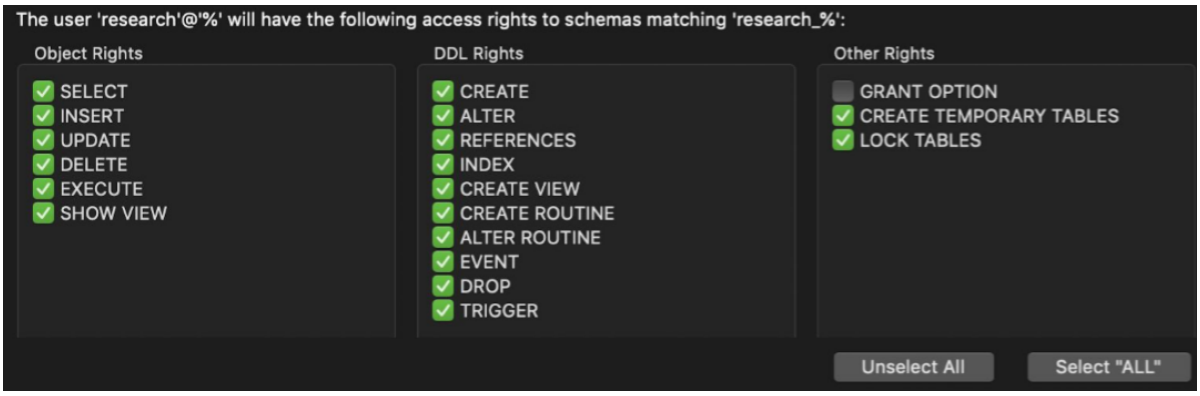
Article: If you want to learn more about what info is stored in the database and how this is connected to the assets and Flix, take a look at this Knowledge Base Article.

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.

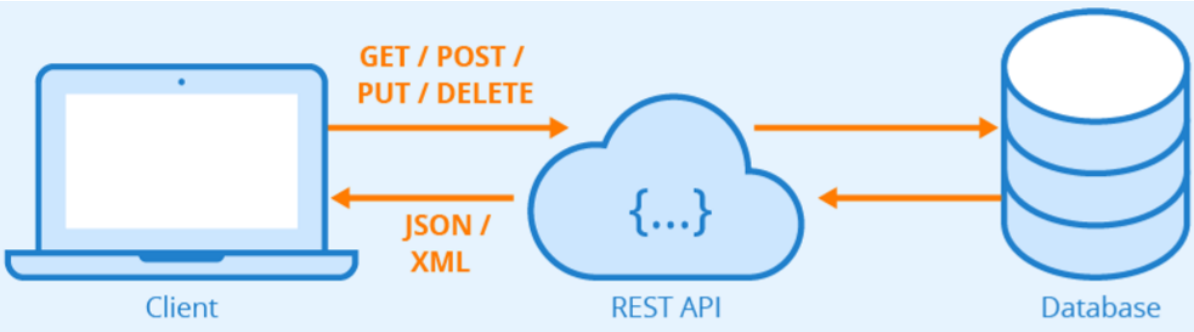
Flix performs a wide variety of actions on the database and requires an extended permission set. Flix checks your permissions on start-up to ensure it has the correct privileges.



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 Import	245.7	0.5	0	>30 min	Warning: chunk upload was a little slow: ~9 seconds	
Manual Import	245.7	1	0	>30 min	Warning: chunk upload was a little slow: ~4.5 seconds	
Manual Import	245.7	2	0	25 min	Warning: chunk upload was a little slow: ~2 seconds	
Manual Import	245.7	2.5	0	22 min	None	**Minimum Bandwidth**
Manual Import	245.7	3	0	17:45	None	
Manual Import	245.7	4	0	13:40	None	
Manual Import	245.7	6	0	10:35	None	
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	

Supported File Formats

The following table lists the supported file formats in Flix. The extensions listed under "Extension" let you specify the file format; use these as the actual filename extensions or the prefix to indicate output format.

Via the Import Button

Format	Extension
Stills	
JPEG	jpeg, jpg
PNG	png
PSD	psd
OGG	.ogg
TIFF	.tiff
Animated	
QuickTime	mov
Audio	
MP3	mp3
WAVE	wav

From Editorial

Format	Extension
Stills	
XML	xml
AAF DNxHD 36, 45 (DNxHD LB) DNxHD 115, 120 and 145 (DNxHD SQ) **Note:** From Media Composer version 2020.4 onwards, DNxHD resolution names have been simplified. See resolution names in brackets above.	.aaf
Animated	
QuickTime	mov
AAF DNxHD 36, 45 (DNxHD LB) DNxHD 115, 120 and 145 (DNxHD SQ) **Note:** From Media Composer version 2020.4 onwards, DNxHD resolution names have been simplified. See resolution names in brackets above.	.aaf

Third-Party Application Support

This page lists the supported applications used in Flix.

Applications	Supported Versions
Adobe Photoshop	CC 2019, CC 2020, 2021
Adobe Premiere	CC 2019, CC 2020
Avid Media Composer	2018, 2019, 2020
Toon Boom Storyboard Pro	6, 7, 20

Note: * Older versions may work, but they haven't been fully tested with Flix 6.3, so they cannot be listed as officially supported. ** Older and newer versions may work, but they haven't been fully tested with Flix 6.3, so they cannot be listed as officially supported.
If you have any problems with a particular application's version, please contact the Foundry Support Team, refer to Contacting Customer Support.

Third-Party Software Notices

This page lists third-party notices and versions used in Flix.

Third-Party Notice Versions

Library	Version	Library	Version
angular-devkit/build-optimizer	0.0.13	json-stringify-safe	5.0.1
angular-devkit/build-optimizer	0.0.15	json3	3.3.2
angular/animations	4.4.6	json5	0.5.1
angular/cli	1.3.1	jsonfile	2.4.0
angular/common	4.4.6	jsonfile	3.0.1

Library	Version	Library	Version
angular/compiler-cli	4.3.5	jsonfile	4.0.0
angular/compiler	4.4.6	jsonify	0.0.0
angular/core	4.4.6	jsonpointer	4.0.1
angular/forms	4.4.6	jspm-config	0.3.4
angular/http	4.4.6	jsprim	1.4.0
angular/platform-browser-dynamic	4.4.6	jsprim	1.4.1
angular/platform-browser	4.4.6	junit-report-builder	1.3.1
angular/router	4.4.6	karma-chrome-launcher	2.2.0
angular/tsc-wrapped	4.3.5	karma-cli	1.0.1
clr/icons	0.12.9	karma-coverage-istanbul-reporter	1.4.0
ngtools/json-schema	1.1.0	karma-jasmine-html-reporter	0.2.2
ngtools/webpack	1.6.1	karma-jasmine	1.1.0
tweenjs/tween.js	17.2.0	karma-json-fixtures-preprocessor	0.0.6
types/chai	4.1.3	karma-junit-reporter	1.2.0
types/core-js	0.9.36	karma-mocha-reporter	2.2.4
types/howler	2.0.5	karma-source-map-support	1.2.0
types/jasmine	2.5.53	karma-sourcemap-loader	0.3.7
types/jquery	2.0.49	karma	1.7.0
types/node	7.0.7	kind-of	2.0.1

Library	Version	Library	Version
types/node	8.10.21	kind-of	3.2.2
types/semver	5.5.0	kind-of	4.0.0
types/webdriverio	4.10.3	klaw	1.3.1
webcomponents/custom-elements	1.0.4	known-css-properties	0.3.0
abbrev	1.1.0	latest-version	3.1.0
accepts	1.3.3	lazy-cache	0.2.7
accepts	1.3.4	lazy-cache	1.0.4
accessibility-developer-tools	2.12.0	lazystream	1.0.0
acorn-dynamic-import	2.0.2	lcid	1.0.0
acorn-jsx	3.0.1	less-loader	4.0.4
acorn	3.3.0	less-loader	4.0.6
acorn	4.0.13	less	2.7.2
acorn	5.1.2	levn	0.3.0
acorn	5.3.0	license-webpack-plugin	0.5.1
adm-zip	0.4.7	line-by-line	0.1.6
after	0.8.2	listify	1.0.0
agent-base	2.1.1	load-json-file	1.1.0
agent-base	4.2.1	load-json-file	2.0.0
ajv-keywords	1.5.1	loader-runner	2.3.0
ajv-keywords	2.1.0	loader-utils	0.2.17
ajv	4.11.8	loader-utils	1.1.0

Library	Version	Library	Version
ajv	5.2.3	locate-path	2.0.0
ajv	5.5.2	lockfile	1.0.3
align-text	0.1.4	lodash.assign	4.2.0
alphanum-sort	1.0.2	lodash.camelcase	4.3.0
amdefine	1.0.1	lodash.capitalize	4.2.1
angular-2-local-storage	1.0.1	lodash.clonedeep	4.5.0
angular-split	0.2.2	lodash.get	4.4.2
angular2-moment	1.9.0	lodash.kebabcase	4.1.1
angular2-virtual-scroll	0.4.12	lodash.memoize	4.1.2
ansi-align	2.0.0	lodash.mergewith	4.6.0
ansi-escapes	1.4.0	lodash.tail	4.1.1
ansi-escapes	3.0.0	lodash.uniq	4.5.0
ansi-html	0.0.7	lodash	3.10.1
ansi-regex	2.1.1	lodash	4.17.10
ansi-regex	3.0.0	lodash	4.17.4
ansi-styles	2.2.1	log-symbols	2.1.0
ansi-styles	3.2.0	log-update	1.0.2
ansi-styles	3.2.1	log4js	0.6.38
any-promise	1.3.0	loglevel	1.6.1
anymatch	1.3.2	longest	1.0.1
app-root-path	2.0.1	loose-envify	1.3.1
append-transform	0.4.0	loud-rejection	1.6.0
aproba	1.1.1	lower-case	1.1.4

Library	Version	Library	Version
aproba	1.1.2	lowercase-keys	1.0.0
archiver-utils	1.3.0	lru-cache	2.2.4
archiver	2.1.1	lru-cache	4.1.1
archy	1.0.0	ltcdr	2.2.1
are-we-there-yet	1.1.4	macaddress	0.2.8
argparse	1.0.9	magic-string	0.22.4
arr-diff	2.0.0	make-dir	1.2.0
arr-flatten	1.1.0	make-error-cause	1.2.2
array-find-index	1.0.2	make-error	1.3.4
array-flatten	1.1.1	map-obj	1.0.1
array-flatten	2.1.1	matcher-collection	1.0.5
array-slice	0.2.3	math-expression-evaluator	1.2.17
array-union	1.0.2	md5.js	1.3.4
array-uniq	1.0.3	md5	2.2.1
array-unique	0.2.1	media-typer	0.3.0
arraybuffer.slice	0.0.6	mem	1.1.0
arrify	1.0.1	memory-fs	0.4.1
asap	2.0.6	meow	3.7.0
asar	0.14.3	merge-descriptors	1.0.1
asn1.js	4.9.1	merge	1.2.0
asn1	0.2.3	methods	1.1.2
assert-plus	0.2.0	micromatch	2.3.11

Library	Version	Library	Version
assert-plus	1.0.0	miller-rabin	4.0.1
assert	1.4.1	mime-db	1.27.0
assertion-error	1.1.0	mime-db	1.29.0
ast-types	0.9.6	mime-db	1.30.0
async-each	1.0.1	mime-db	1.33.0
async-foreach	0.1.3	mime-types	2.1.15
async-limiter	1.0.0	mime-types	2.1.16
async	0.2.10	mime-types	2.1.17
async	0.9.2	mime-types	2.1.18
async	1.5.2	mime	1.3.4
async	2.5.0	mime	1.3.6
async	2.6.0	mime	2.3.1
asynckit	0.4.0	mimic-fn	1.1.0
atoa	1.0.0	minimalistic-assert	1.0.0
atob	1.1.3	minimalistic-crypto-utils	1.0.1
author-regex	1.0.0	minimatch	3.0.4
autoprefixer	6.7.7	minimist	0.0.10
autoprefixer	7.1.1	minimist	0.0.8
aws-sign2	0.6.0	minimist	1.1.3
aws-sign2	0.7.0	minimist	1.2.0
aws4	1.6.0	mixin-object	2.0.1
axios	0.18.0	mkdirp	0.5.0

Library	Version	Library	Version
babel-code-frame	6.22.0	mkdirp	0.5.1
babel-generator	6.25.0	mkpath	0.1.0
babel-messages	6.23.0	mksnapshot	0.3.1
babel-runtime	6.25.0	mocha-junit-reporter	1.18.0
babel-runtime	6.26.0	mocha	4.1.0
babel-template	6.25.0	mocha	5.0.4
babel-traverse	6.25.0	moment	2.19.1
babel-types	6.25.0	moment	2.22.2
babylon	6.17.4	ms	0.7.1
babylon	6.18.0	ms	0.7.2
backo2	1.0.2	ms	2.0.0
balanced-match	0.4.2	multicast-dns-service-types	1.1.0
balanced-match	1.0.0	multicast-dns	6.1.1
base64-arraybuffer	0.1.5	mute-stream	0.0.5
base64-js	1.2.0	mute-stream	0.0.7
base64-js	1.2.1	mydaterangepicker	4.2.1
base64-js	1.3.0	mysql	2.16.0
base64id	1.0.0	mz	2.7.0
batch	0.6.1	nan	2.6.2
bcrypt-pbkdf	1.0.1	ncname	1.0.0
better-assert	1.0.2	negotiator	0.6.1
big.js	3.1.3	ng2-dragula	1.5.0

Library	Version	Library	Version
bignumber.js	4.1.0	ng2-ion-range-slider	2.0.0
binary-extensions	1.9.0	ng2-material-dropdown	0.8.0
binary	0.3.0	ng2-nouislider	1.7.10
bl	1.2.1	ng2-page-scroll	4.0.0-beta.12
blob-to-buffer	1.2.7	ngx-chips	1.5.11
blob-to-stream	1.0.3	ngx-color-picker	6.4.0
blob	0.0.4	no-case	2.3.1
block-stream	0.0.9	node-fetch	1.7.3
bluebird	3.5.1	node-forge	0.6.33
bn.js	4.11.8	node-gyp	3.6.2
body-parser	1.18.2	node-libs-browser	2.0.0
bonjour	3.5.0	node-modules-path	1.0.1
boolbase	1.0.0	node-pre-gyp	0.6.36
boom	2.10.1	node-sass	4.7.2
bootstrap	4.0.0-alpha.6	nodeify	1.0.1
boxen	1.3.0	noms	0.0.0
brace-expansion	1.1.11	nopt	1.0.10
brace-expansion	1.1.7	nopt	3.0.6
brace-expansion	1.1.8	nopt	4.0.1
braces	0.1.5	normalize-package-data	2.4.0
braces	1.8.5	normalize-path	2.1.1

Library	Version	Library	Version
brorand	1.1.0	normalize-range	0.1.2
browser-stdout	1.3.0	normalize-url	1.9.1
browser-stdout	1.3.1	nouislider	11.1.0
browserify-aes	1.0.8	npm-install-package	2.1.0
browserify-cipher	1.0.0	npm-run-path	2.0.2
browserify-des	1.0.0	npmlog	4.1.0
browserify-rsa	4.0.1	npmlog	4.1.2
browserify-sign	4.0.4	nth-check	1.0.1
browserify-zlib	0.1.4	nugget	2.0.1
browserslist	1.7.7	null-check	1.0.0
browserslist	2.4.0	num2fraction	1.2.2
buffer-crc32	0.2.13	number-is-nan	1.0.1
buffer-from	1.1.1	oauth-sign	0.8.2
buffer-indexof	1.1.1	object-assign	4.1.0
buffer-shims	1.0.0	object-assign	4.1.1
buffer-xor	1.0.3	object-component	0.0.3
buffer	4.9.1	object-keys	0.4.0
buffers	0.1.1	object-keys	1.0.11
builtin-modules	1.1.1	object.assign	4.1.0
builtin-status-codes	3.0.0	object.omit	2.0.1
bytes	3.0.0	object.pick	1.3.0
caller-path	0.1.0	obuf	1.1.1
callsite	1.0.0	on-finished	2.3.0

Library	Version	Library	Version
callsites	0.2.0	on-headers	1.0.1
camel-case	3.0.0	once	1.4.0
camelcase-keys	2.1.0	onetime	1.1.0
camelcase	1.2.1	onetime	2.0.1
camelcase	2.1.1	opn	4.0.2
camelcase	3.0.0	opn	5.1.0
camelcase	4.1.0	optimist	0.3.7
caniuse-api	1.6.1	optimist	0.6.1
caniuse-db	1.0.30000713	optionator	0.8.2
caniuse-lite	1.0.30000713	options	0.0.6
caniuse-lite	1.0.30000744	original	1.0.0
capture-stack-trace	1.0.0	os-browserify	0.2.1
caseless	0.11.0	os-homedir	1.0.2
caseless	0.12.0	os-locale	1.4.0
center-align	0.1.3	os-locale	2.1.0
chai-as-promised	5.3.0	os-tmpdir	1.0.2
chai	3.5.0	osenv	0.1.4
chai	4.1.2	p-finally	1.0.0
chainsaw	0.1.0	p-limit	1.1.0
chalk	1.1.3	p-locate	2.0.0
chalk	2.1.0	p-map	1.1.1
chalk	2.3.2	package-json	4.0.1
chalk	2.4.1	pako	0.2.9

Library	Version	Library	Version
charenc	0.0.2	param-case	2.1.1
check-error	1.0.2	parse-asn1	5.1.0
chokidar	1.7.0	parse-author	2.0.0
chromium-pickle-js	0.2.0	parse-glob	3.0.4
cipher-base	1.0.4	parse-json	2.2.0
circular-dependency-plugin	3.0.0	parsejson	0.0.3
circular-dependency-plugin	4.2.1	parseqs	0.0.5
circular-json	0.3.3	parseuri	0.0.5
clap	1.2.0	parseurl	1.3.1
clarity-angular	0.10.25	parseurl	1.3.2
clarity-ui	0.10.25	path-browserify	0.0.0
clean-css	4.1.7	path-exists	2.1.0
cli-boxes	1.0.0	path-exists	3.0.0
cli-cursor	1.0.2	path-is-absolute	1.0.1
cli-cursor	2.1.0	path-is-inside	1.0.2
cli-truncate	1.1.0	path-key	2.0.1
cli-width	2.1.0	path-parse	1.0.5
cliui	2.1.0	path-to-regexp	0.1.7
cliui	3.2.0	path-type	1.1.0
clone-deep	0.3.0	path-type	2.0.0
clone	1.0.2	pathval	1.1.0
clone	2.1.1	pbkdf2	3.0.14
co	4.6.0	pend	1.2.0

Library	Version	Library	Version
coa	1.0.4	performance-now	0.2.0
code-point-at	1.1.0	performance-now	2.1.0
codelyzer	3.1.1	pify	2.3.0
color-convert	1.9.0	pify	3.0.0
color-name	1.1.3	pinkie-promise	2.0.1
color-string	0.3.0	pinkie	2.0.4
color	0.11.4	plist	2.1.0
colormin	1.1.2	pluralize	1.2.1
colors	1.1.2	popsicle-proxy-agent	3.0.0
columnify	1.5.4	popsicle-retry	3.2.1
combine-lists	1.0.1	popsicle-rewrite	1.0.0
combined-stream	1.0.5	popsicle-status	2.0.1
combined-stream	1.0.6	popsicle	9.2.0
commander	2.11.0	portfinder	1.0.13
common-tags	1.4.0	postcss-calc	5.3.1
compare-version	0.1.2	postcss-colormin	2.2.2
component-bind	1.0.0	postcss-convert-values	2.6.1
component-emitter	1.1.2	postcss-discard-comments	2.0.4
component-emitter	1.2.1	postcss-discard-duplicates	2.1.0
component-inherit	0.0.3	postcss-discard-empty	2.1.0
compress-commons	1.2.2	postcss-discard-overridden	0.1.1

Library	Version	Library	Version
compressible	2.0.11	postcss-discard-unused	2.2.3
compression	1.7.1	postcss-filter-plugins	2.0.2
concat-map	0.0.1	postcss-load-config	1.2.0
concat-stream	1.5.0	postcss-load-options	1.2.0
concat-stream	1.6.0	postcss-load-plugins	2.3.0
concat-stream	1.6.2	postcss-loader	1.3.3
configstore	3.1.1	postcss-merge-idents	2.1.7
connect-history-api-fallback	1.3.0	postcss-merge-longhand	2.0.2
connect	3.6.5	postcss-merge-rules	2.1.2
console-browserify	1.1.0	postcss-message-helpers	2.0.0
console-control-strings	1.1.0	postcss-minify-font-values	1.0.5
constants-browserify	1.0.0	postcss-minify-gradients	1.0.5
content-disposition	0.5.2	postcss-minify-params	1.2.2
content-type	1.0.2	postcss-minify-selectors	2.1.1
content-type	1.0.4	postcss-modules-extract-imports	1.1.0
contra	1.9.4	postcss-modules-local-by-default	1.2.0
convert-source-map	1.5.0	postcss-modules-scope	1.1.0
cookie-signature	1.0.6	postcss-modules-values	1.3.0
cookie	0.3.1	postcss-normalize-	1.1.1

Library	Version	Library	Version
		charset	
copyfiles	1.2.0	postcss-normalize-url	3.0.8
core-js	2.4.1	postcss-ordered-values	2.2.3
core-object	3.1.5	postcss-reduce-idents	2.4.0
core-util-is	1.0.2	postcss-reduce-initial	1.0.1
cosmiconfig	2.2.2	postcss-reduce-transforms	1.0.4
crc32-stream	2.0.0	postcss-selector-parser	2.2.3
crc	3.5.0	postcss-svgo	2.1.6
create-ecdh	4.0.0	postcss-unique-selectors	2.0.2
create-error-class	3.0.2	postcss-url	5.1.2
create-hash	1.1.3	postcss-url	7.0.0
create-hmac	1.1.6	postcss-value-parser	3.3.0
cross-env	5.0.1	postcss-zindex	2.2.0
cross-spawn	3.0.1	postcss	5.2.17
cross-spawn	5.1.0	postcss	5.2.18
crossvent	1.5.4	postcss	6.0.8
crypt	0.0.2	prelude-ls	1.1.2
cryptiles	2.0.5	prepend-http	1.0.4
crypto-browserify	3.11.1	preserve	0.2.0
crypto-js	3.1.9-1	pretty-bytes	1.0.4
crypto-random-string	1.0.0	pretty-error	2.1.1

Library	Version	Library	Version
css-color-names	0.0.4	private	0.1.7
css-loader	0.28.4	process-nextick-args	1.0.7
css-parse	1.7.0	process-nextick-args	2.0.0
css-parse	2.0.0	process	0.11.10
css-select	1.2.0	progress-stream	1.2.0
css-selector-tokenizer	0.7.0	progress	1.1.8
css-value	0.0.1	progress	2.0.0
css-what	2.1.0	promise-finally	3.0.0
css	2.2.1	promise	1.3.0
cssauron	1.4.0	promise	7.3.1
cssesc	0.1.0	proxy-addr	1.1.5
cssnano	3.10.0	proxy-from-env	1.0.0
csso	2.3.2	prr	0.0.0
cuint	0.2.2	pruner	0.0.7
currently-unhandled	0.4.1	pseudomap	1.0.2
custom-event	1.0.0	public-encrypt	4.0.0
custom-event	1.0.1	punycode	1.3.2
d	1.0.0	punycode	1.4.1
dashdash	1.14.1	puppeteer	1.7.0
date-format	0.0.2	q	1.5.0
date-now	0.1.4	qjobs	1.1.5
debug	0.7.4	qs	6.3.2
debug	2.2.0	qs	6.4.0

Library	Version	Library	Version
debug	2.3.3	qs	6.5.0
debug	2.6.8	qs	6.5.1
debug	2.6.9	qs	6.5.2
debug	3.1.0	query-string	4.3.4
decamelize	1.2.0	querystring-es3	0.2.1
decompress-zip	0.3.0	querystring	0.2.0
deep-diff	1.0.1	querystringify	0.0.4
deep-eql	0.1.3	querystringify	1.0.0
deep-eql	3.0.1	randomatic	1.1.7
deep-equal	1.0.1	randombytes	2.0.5
deep-extend	0.4.2	range-parser	1.2.0
deep-is	0.1.3	raw-body	2.3.2
deepmerge	2.0.1	raw-loader	0.5.1
default-require-extensions	1.0.0	rc	1.2.1
defaults	1.0.3	rcedit	1.0.0
define-properties	1.1.2	read-pkg-up	1.0.1
defined	1.0.0	read-pkg-up	2.0.0
del	2.2.2	read-pkg	1.1.0
del	3.0.0	read-pkg	2.0.0
delayed-stream	1.0.0	readable-stream	1.0.34
delegates	1.0.0	readable-stream	1.1.14
denodeify	1.2.1	readable-stream	2.0.6
depd	1.1.1	readable-stream	2.2.9

Library	Version	Library	Version
des.js	1.0.0	readable-stream	2.3.3
destroy	1.0.4	readable-stream	2.3.6
detect-indent	4.0.0	readdirp	2.1.0
detect-indent	5.0.0	readline2	1.0.1
detect-node	2.0.3	recast	0.11.23
dev-null	0.1.1	reconnecting-websocket	4.0.0-rc5
devtron	1.4.0	redent	1.0.0
di	0.0.1	reduce-css-calc	1.3.0
diff	3.3.1	reduce-function-call	1.0.2
diff	3.5.0	reflect-metadata	0.1.10
diffie-hellman	5.0.2	regenerate	1.3.3
directory-encoder	0.7.2	regenerator-runtime	0.10.5
dns-equal	1.0.0	regenerator-runtime	0.11.1
dns-packet	1.2.2	regex-cache	0.4.3
dns-txt	2.0.2	regexpu-core	1.0.0
doctrine	1.5.0	registry-auth-token	3.3.2
dom-converter	0.1.4	registry-url	3.1.0
dom-serialize	2.2.1	regjsgen	0.2.0
dom-serializer	0.1.0	regjsparser	0.1.5
domain-browser	1.1.7	relateurl	0.2.7
domelementtype	1.1.3	remove-trailing-separator	1.0.2

Library	Version	Library	Version
domelementtype	1.3.0	renderkid	2.0.1
domhandler	2.1.0	repeat-element	1.1.2
domutils	1.1.6	repeat-string	0.2.2
domutils	1.5.1	repeat-string	1.6.1
dot-prop	4.2.0	repeating	2.0.1
dragula	3.7.2	request	2.79.0
duplexer3	0.1.4	request	2.81.0
ecc-jsbn	0.1.1	request	2.87.0
ee-first	1.1.1	require-directory	2.1.1
ejs	2.5.7	require-from-string	1.2.1
electron-chromedriver	1.7.1	require-main-filename	1.0.1
electron-download	3.3.0	require-uncached	1.0.3
electron-download	4.1.0	requires-port	1.0.0
electron-json-storage	4.0.2	resize-observer-polyfill	1.5.0
electron-log	2.2.14	resolve-from	1.0.1
electron-osx-sign	0.4.10	resolve-url	0.2.1
electron-packager	11.1.0	resolve	1.4.0
electron-reload	1.2.1	restore-cursor	1.0.1
electron-to-chromium	1.3.18	restore-cursor	2.0.0
electron	2.0.5	rgb2hex	0.1.8
elegant-spinner	1.0.1	right-align	0.1.3
elliptic	6.4.0	rimraf	2.2.8
ember-cli-normalize-	1.0.0	rimraf	2.6.1

Library	Version	Library	Version
entity-name			
ember-cli-string-utils	1.1.0	rimraf	2.6.2
emojis-list	2.1.0	ripemd160	2.0.1
encodeurl	1.0.1	rsvp	3.2.1
encoding	0.1.12	rsvp	3.6.2
end-of-stream	1.4.1	run-async	0.1.0
engine.io-client	1.8.3	run-async	2.3.0
engine.io-parser	1.3.2	rwlock	5.0.0
engine.io	1.8.3	rx-lite-aggregates	4.0.8
enhanced-resolve	3.3.0	rx-lite	3.1.2
enhanced-resolve	3.4.1	rx-lite	4.0.8
ensure-posix-path	1.0.2	rxjs	5.5.2
ent	2.2.0	safe-buffer	5.0.1
entities	1.1.1	safe-buffer	5.1.1
env-paths	1.0.0	safe-buffer	5.1.2
errno	0.1.4	sanitize-filename	1.6.1
error-ex	1.3.1	sass-graph	2.2.4
es5-ext	0.10.30	sass-lint	1.12.1
es6-iterator	2.0.1	sass-loader	6.0.6
es6-map	0.1.5	sax	0.5.8
es6-promise	4.2.4	sax	1.2.4
es6-promisify	5.0.0	schema-utils	0.3.0
es6-set	0.1.5	script-loader	0.7.0

Library	Version	Library	Version
es6-symbol	3.1.1	scss-tokenizer	0.2.3
es6-templates	0.2.3	select-hose	2.0.0
es6-weak-map	2.0.2	selfsigned	1.10.1
escape-html	1.0.3	semver-diff	2.1.0
escape-string-regexp	1.0.5	semver-dsl	1.0.1
escope	3.6.0	semver	4.3.6
eslint	2.13.1	semver	5.0.3
espree	3.5.2	semver	5.3.0
esprima	2.7.3	semver	5.4.1
esprima	3.1.3	send	0.15.4
esrecurse	4.2.0	serve-index	1.9.1
estraverse	4.2.0	serve-static	1.12.4
esutils	2.0.2	set-blocking	2.0.0
etag	1.8.0	set-immediate-shim	1.0.1
event-emitter	0.3.5	setimmediate	1.0.5
eventemitter3	1.2.0	setprototypeof	1.0.3
events	1.1.1	sha.js	2.4.11
eventsource	0.1.6	shallow-clone	0.1.2
evp_bytestokey	1.0.3	shebang-command	1.2.0
execa	0.7.0	shebang-regex	1.0.0
exit-hook	1.1.1	shelljs	0.6.1
expand-braces	0.1.2	signal-exit	3.0.2
expand-brackets	0.1.5	silent-error	1.1.0

Library	Version	Library	Version
expand-range	0.1.1	single-line-log	1.1.2
expand-range	1.8.2	slice-ansi	0.0.4
exports-loader	0.6.4	slice-ansi	1.0.0
express	4.15.4	sntp	1.0.9
extend	3.0.1	socket.io-adapter	0.5.0
external-editor	2.0.4	socket.io-client	1.7.3
extglob	0.3.2	socket.io-parser	2.3.1
extract-text-webpack-plugin	3.0.0	socket.io	1.7.3
extract-zip	1.6.0	sockjs-client	1.1.2
extract-zip	1.6.6	sockjs	0.3.18
extract-zip	1.6.7	sort-keys	1.1.2
extsprintf	1.0.2	source-list-map	0.1.8
extsprintf	1.3.0	source-list-map	2.0.0
fast-deep-equal	1.0.0	source-map-loader	0.2.1
fast-json-stable-stringify	2.0.0	source-map-resolve	0.3.1
fast-levenshtein	2.0.6	source-map-support	0.4.18
fastparse	1.1.1	source-map-support	0.5.4
faye-websocket	0.10.0	source-map-url	0.3.0
faye-websocket	0.11.1	source-map	0.1.43
fd-slicer	1.0.1	source-map	0.4.4
fibers	2.0.0	source-map	0.5.6
figures	1.7.0	source-map	0.5.7

Library	Version	Library	Version
figures	2.0.0	source-map	0.6.1
file-entry-cache	1.3.1	spdx-correct	1.0.2
file-loader	0.10.1	spdx-expression-parse	1.0.4
file-loader	0.11.2	spdx-license-ids	1.2.2
filename-regex	2.0.1	spdy-transport	2.0.20
fileset	2.0.3	spdy	3.4.7
fill-range	2.2.3	spectron	3.7.2
finalhandler	1.0.4	speedometer	0.1.4
finalhandler	1.0.6	split	1.0.1
find-up	1.1.2	sprintf-js	1.0.3
find-up	2.1.0	sqlstring	2.3.1
flat-cache	1.3.0	sshpk	1.13.0
flatten	1.0.2	sshpk	1.13.1
flix	6.0.0-dev.1	statuses	1.3.1
follow-redirects	1.4.1	stdout-stream	1.4.0
follow-redirects	1.5.9	stream-browserify	2.0.1
font-awesome	4.7.0	stream-http	2.7.2
for-in	0.1.8	strict-uri-encode	1.1.0
for-in	1.0.2	string-template	1.0.0
for-own	0.1.5	string-width	1.0.2
for-own	1.0.0	string-width	2.1.1
foreach	2.0.5	string_decoder	0.10.31
forever-agent	0.6.1	string_decoder	1.0.1

Library	Version	Library	Version
form-data	2.1.4	string_decoder	1.0.3
form-data	2.3.2	string_decoder	1.1.1
forwarded	0.1.0	stringstream	0.0.5
fresh	0.5.0	strip-ansi	3.0.1
front-matter	2.1.2	strip-ansi	4.0.0
fs-access	1.0.1	strip-bom	2.0.0
fs-extra	0.23.1	strip-bom	3.0.0
fs-extra	0.26.7	strip-eof	1.0.0
fs-extra	0.30.0	strip-indent	1.0.1
fs-extra	2.1.2	strip-json-comments	1.0.4
fs-extra	3.0.1	strip-json-comments	2.0.1
fs-extra	4.0.3	style-loader	0.13.2
fs-extra	5.0.0	style-loader	0.18.2
fs.realpath	1.0.0	stylus-loader	3.0.1
fsevents	1.1.2	stylus	0.54.5
fstream-ignore	1.0.5	sumchecker	1.3.1
fstream	1.0.11	sumchecker	2.0.2
function-bind	1.1.0	supports-color	2.0.0
function-bind	1.1.1	supports-color	3.2.3
gauge	2.7.4	supports-color	4.2.1
gaze	1.1.2	supports-color	4.4.0
generate-function	2.0.0	supports-color	4.5.0
generate-object-property	1.2.0	supports-color	5.0.1

Library	Version	Library	Version
get-caller-file	1.0.2	supports-color	5.3.0
get-func-name	2.0.0	supports-color	5.4.0
get-package-info	1.0.0	svgo	0.7.2
get-stdin	4.0.1	symbol-observable	1.2.0
get-stream	3.0.0	table	3.8.3
getpass	0.1.7	tapable	0.2.8
glob-base	0.3.0	tar-pack	3.4.0
glob-parent	2.0.0	tar-stream	1.5.5
glob	6.0.4	tar	2.2.1
glob	7.0.6	temp	0.8.3
glob	7.1.2	term-size	1.2.0
global-dirs	0.1.1	tether	1.4.3
globals	9.18.0	text-table	0.2.0
globby	5.0.0	thenify-all	1.6.0
globby	6.1.0	thenify	3.3.0
globule	1.2.0	throat	3.2.0
gonzales-pe-sl	4.2.3	throttleit	0.0.2
got	6.7.1	through2	0.2.3
graceful-fs	4.1.11	through2	2.0.3
growl	1.10.3	through	2.3.8
handle-thing	1.2.5	thunky	0.1.0
handlebars	1.3.0	ticky	1.0.1
handlebars	4.0.11	time-stamp	2.0.0

Library	Version	Library	Version
har-schema	1.0.5	timed-out	4.0.1
har-schema	2.0.0	timers-browserify	2.0.4
har-validator	2.0.6	tmp	0.0.28
har-validator	4.2.1	tmp	0.0.31
har-validator	5.0.3	to-array	0.1.4
has-ansi	2.0.0	to-arraybuffer	1.0.1
has-binary	0.1.7	to-fast-properties	1.0.3
has-cors	1.1.0	toposort	1.0.6
has-flag	1.0.0	touch	0.0.3
has-flag	2.0.0	touch	1.0.0
has-flag	3.0.0	tough-cookie	2.3.2
has-symbols	1.0.0	tough-cookie	2.3.4
has-unicode	2.0.1	traverse	0.3.9
has	1.0.1	trim-newlines	1.0.0
hash-base	2.0.2	trim-right	1.0.1
hash-base	3.0.4	true-case-path	1.0.2
hash.js	1.1.3	truncate-utf8-bytes	1.0.2
hawk	3.1.3	ts-node	5.0.1
he	1.1.1	tsickle	0.21.6
heimdalljs-logger	0.1.9	tslib	1.7.1
heimdalljs	0.2.5	tslint	5.4.3
highlight.js	9.12.0	tsutils	2.11.2
hmac-drbg	1.0.1	tty-browserify	0.0.0

Library	Version	Library	Version
hoek	2.16.3	tunnel-agent	0.4.3
home-path	1.0.6	tunnel-agent	0.6.0
hosted-git-info	2.5.0	tweetnacl	0.14.5
hpack.js	2.1.6	type-check	0.3.2
html-comment-regex	1.1.1	type-detect	0.1.1
html-entities	1.2.1	type-detect	1.0.0
html-loader	0.4.5	type-detect	4.0.8
html-minifier	3.5.3	type-is	1.6.15
html-webpack-plugin	2.30.1	typedarray	0.0.6
htmlparser2	3.3.0	typescript	2.2.2
http-deceiver	1.2.7	typescript	2.4.2
http-errors	1.6.2	typescript	2.5.0
http-parser-js	0.4.9	typings-core	2.3.3
http-proxy-agent	1.0.0	typings	2.1.1
http-proxy-middleware	0.17.4	uglify-js	2.3.6
http-proxy	1.16.2	uglify-js	2.8.29
http-signature	1.1.1	uglify-js	3.0.27
http-signature	1.2.0	uglify-to-browserify	1.0.2
https-browserify	0.0.1	uglifyjs-webpack-plugin	0.4.6
https-proxy-agent	1.0.0	uid-number	0.0.6
https-proxy-agent	2.2.1	ultron	1.0.2
humanize-duration	3.15.1	ultron	1.1.1

Library	Version	Library	Version
humanize-plus	1.8.2	unc-path-regex	0.1.2
iconv-lite	0.4.18	uniq	1.0.1
iconv-lite	0.4.19	uniqid	4.1.1
icss-replace-symbols	1.1.0	uniqs	2.0.0
icss-utils	2.1.0	unique-string	1.0.0
ieee754	1.1.8	universalify	0.1.1
ignore-loader	0.1.2	unpipe	1.0.0
ignore	3.3.7	unzip-response	2.0.1
image-size	0.5.5	update-notifier	2.3.0
image-size	0.6.2	upper-case	1.1.3
img-stats	0.5.2	urix	0.1.0
import-lazy	2.1.0	url-loader	0.5.9
imurmurhash	0.1.4	url-parse-lax	1.0.0
in-publish	2.0.0	url-parse	1.0.5
indent-string	2.1.0	url-parse	1.1.9
indexes-of	1.0.1	url	0.11.0
indexof	0.0.1	user-home	2.0.0
inflection	1.12.0	useragent	2.2.1
inflight	1.0.6	utf8-byte-length	1.0.4
inherits	2.0.1	util-deprecate	1.0.2
inherits	2.0.3	util	0.10.3
ini	1.3.4	utila	0.3.3
inquirer	0.12.0	utila	0.4.0

Library	Version	Library	Version
inquirer	3.3.0	utils-merge	1.0.0
internal-ip	1.2.0	utils-merge	1.0.1
interpret	1.0.3	uuid	2.0.3
invariant	2.2.2	uuid	3.0.1
invert-kv	1.0.0	uuid	3.1.0
ion-rangeslider	2.2.0	uuid	3.3.2
ip	1.1.5	validate-npm-package-license	3.0.1
ipaddr.js	1.4.0	vary	1.1.1
is-absolute-url	2.1.0	vary	1.1.2
is-absolute	0.2.6	vendors	1.0.1
is-arrayish	0.2.1	verror	1.10.0
is-binary-path	1.0.1	verror	1.3.6
is-buffer	1.1.5	vlq	0.2.3
is-buffer	1.1.6	vm-browserify	0.0.4
is-builtin-module	1.0.0	void-elements	2.0.1
is-directory	0.3.1	walk-sync	0.3.2
is-dotfile	1.0.3	watchpack	1.4.0
is-equal-shallow	0.1.3	wav-encoder	1.3.0
is-extendable	0.1.1	wbuf	1.7.2
is-extglob	1.0.0	wcwidth	1.0.1
is-extglob	2.1.1	wdio-chai-plugin	0.0.2
is-finite	1.0.2	wdio-dot-reporter	0.0.9

Library	Version	Library	Version
is-fullwidth-code-point	1.0.0	wdio-junit-reporter	0.4.4
is-fullwidth-code-point	2.0.0	wdio-mocha-framework	0.5.12
is-glob	2.0.1	wdio-spec-reporter	0.1.5
is-glob	3.1.0	wdio-sync	0.7.1
is-installed-globally	0.1.0	webdriver-manager	12.0.6
is-my-json-valid	2.17.1	webdriverio	4.13.1
is-npm	1.0.0	webpack-dev-middleware	1.12.0
is-number	0.1.1	webpack-dev-server	2.5.0
is-number	2.1.0	webpack-dev-server	2.5.1
is-number	3.0.0	webpack-merge	4.1.0
is-obj	1.0.1	webpack-sources	1.0.1
is-path-cwd	1.0.0	webpack	3.3.0
is-path-in-cwd	1.0.0	webpack	3.4.1
is-path-inside	1.0.0	websocket-driver	0.7.0
is-plain-obj	1.1.0	websocket-extensions	0.1.2
is-plain-object	2.0.4	wgxpath	1.0.0
is-posix-bracket	0.1.1	when	3.6.4
is-primitive	2.0.0	whet.extend	0.9.9
is-promise	1.0.1	which-module	1.0.0
is-promise	2.1.0	which-module	2.0.0
is-property	1.0.2	which	1.3.0

Library	Version	Library	Version
is-redirect	1.0.0	wide-align	1.1.2
is-relative	0.2.1	widest-line	2.0.0
is-resolvable	1.0.1	window-size	0.1.0
is-retry-allowed	1.1.0	wordwrap	0.0.2
is-stream	1.1.0	wordwrap	1.0.0
is-svg	2.1.0	wrap-ansi	2.1.0
is-typedarray	1.0.0	wrappy	1.0.2
is-unc-path	0.1.2	write-file-atomic	2.3.0
is-utf8	0.2.1	write	0.2.1
is-windows	0.2.0	ws	1.1.2
is-windows	1.0.1	ws	3.3.2
is-wsl	1.1.0	ws	5.2.2
isarray	0.0.1	wtf-8	1.0.0
isarray	1.0.0	xdg-basedir	3.0.0
isbinaryfile	3.0.2	xml-char-classes	1.0.0
isexe	2.0.0	xml2js	0.4.19
isobject	2.1.0	xml	1.0.1
isobject	3.0.1	xmlbuilder	10.0.0
isstream	0.1.2	xmlbuilder	8.2.2
istanbul-api	1.2.1	xmlbuilder	9.0.4
istanbul-instrumenter-loader	2.0.0	xmlcreate	1.0.2
istanbul-lib-coverage	1.1.1	xmldom	0.1.27

Library	Version	Library	Version
istanbul-lib-hook	1.1.0	xmlhttprequest-ssl	1.5.3
istanbul-lib-instrument	1.7.4	xtend	2.1.2
istanbul-lib-instrument	1.9.1	xtend	4.0.1
istanbul-lib-report	1.1.2	xxhashjs	0.2.1
istanbul-lib-source-maps	1.2.2	y18n	3.2.1
istanbul-reports	1.1.3	yallist	2.1.2
jasmine-core	2.6.4	yargs-parser	4.2.1
jasmine-spec-reporter	4.1.1	yargs-parser	5.0.0
jodid25519	1.0.2	yargs-parser	7.0.0
jquery	3.2.1	yargs-parser	9.0.2
js-base64	2.1.9	yargs	3.10.0
js-tokens	3.0.2	yargs	6.6.0
js-yaml	3.7.0	yargs	7.1.0
js2xmlparser	3.0.0	yargs	8.0.2
jsbn	0.1.1	yauzl	2.4.1
jschardet	1.5.1	yeast	0.1.2
jsesc	0.5.0	yn	2.0.0
jsesc	1.3.0	zip-object	0.1.0
json-loader	0.5.4	zip-stream	1.2.0
json-schema-traverse	0.3.1	zone.js	0.8.12
json-schema	0.2.3	zone.js	0.8.20
json-stable-stringify	1.0.1		

Third-Party Licenses

The third-party licenses are available at this address: https://s3.amazonaws.com/thefoundry/products/flix/6.0v1/flix_6.0_licenses.html

End User License Agreement (EULA)

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; AND (C) CLAUSE 18.2 REGARDING THE DATA WE MAY COLLECT AND HOW WE MAY USE IT.

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 A LOGIN-BASED 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 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 own internal 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) "**Node Locked License**"

If Licensee purchases a Node Locked License, Licensee will install and use only a single copy of the Software on only one computer at a time, which may be located anywhere 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 restricted to the Home Country; (e) use of the Software shall be limited to no more than one concurrent use at all times.

(c) "**Floating License**"

If Licensee purchases a Floating License, use of the Software may be at any site in the Home Country.

(d) "**Login-Based License**"

If Licensee purchases a Login-Based 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) "**Modo Subscription License**"

If Licensee purchases a License for Modo on a subscription basis then the provisions of clause 8 shall apply.

(f) "**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's use of the Software shall be limited to no more than one concurrent use at all times; (e) Licensee shall not purchase or use more than one Mari Individual Subscription License; and (f) the provisions of clause 8 shall apply.

(g) "**Rental Licence**"

If Licensee has purchased a Licence 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.

(h) "**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.

(i) "**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.

(j) "**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.

(k) "**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.

(l) "**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. Licensee will use the Software on only one computer at a time.

(m) "**Subscription License**"

Any reference to a Subscription License shall mean a Modo Subscription 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 are limited to one computer such that the number of computers on which Headless Rendering can be carried out is limited to the number of valid Licenses that have been purchased.

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 authorized 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; (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.

Unless Licensee has purchased an Individual License or a Login-Based License, if the Software is moved from one computer 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 its own internal business 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 any open source software libraries ("**OSS Libraries**") that are provided to Licensee with the Software, nothing in the Agreement shall affect Licensee's rights under the licenses on which the relevant Third Party Licensor has licensed the OSS Libraries, as stated in the Documentation. To the extent that Third Party Licensors have licensed OSS Libraries on the terms of v2.1 of the Lesser General Public License issued by the Free Software Foundation (see http://www.gnu.org/licenses/lgpl-2.1.html) (the "**LGPL**"), those OSS Libraries are licensed to Licensee on the terms of the LGPL and are referred to in this clause 4.2 as the LGPL Libraries. 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 LGPL Libraries (the "**LGPL Source**") in order that Licensee may modify the LGPL Libraries in accordance with the LGPL, together with certain object code of the Software necessary to enable Licensee to re-link any modified LGPL Library to the Software (the "**Object**"); and

4.3 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 Libraries (including the LGPL Libraries), the LGPL Source 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 Libraries (including the LGPL Libraries) or the LGPL Source 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 Libraries (including the LGPL Libraries), the LGPL Source, 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: (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; in each case together with any/all applicable taxes or other duties or levies.

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 you not less than two reminder emails 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 first reminder email will be sent not less than sixty (60) days prior to the relevant Subscription Renewal Date. The second 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 Instalments**. 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.

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 any defect or fault caused by unauthorised use of or any amendment made to the Software by any person other than Foundry. 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. 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 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, 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 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, 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 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, 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, 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 collect limited information from Licensee's computer(s) and transmit it to Foundry, including the ability to locally cache such information on Licensee's computer. Such information (the "**Information**") may include details of Licensee's license(s) to Foundry products, computer and network equipment, details of the operating system(s) in use on such computer equipment, the location of the Licensee's computer(s) and the profile and extent of Licensee's 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, and (d) 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 ("GDPR") it shall be processed in accordance with the GDPR and with Foundry's Privacy Notice (see https://thefoundry.s3.amazonaws.com/documents/Privacy_Notice.pdf), as may be updated by Foundry from time to time. Licensee undertakes to make all of users of the Software aware of the uses which Foundry will make of the Information and of the terms of Foundry's Privacy Policy.

19. U.S. GOVERNMENT LICENSE RIGHTS

All Software, including all components thereof, and Documentation qualify as "commercial items," as that term is defined at Federal Acquisition Regulation ("**FAR**") (48 C.F.R.) 2.101, consisting of "commercial computer software" and "commercial computer software documentation" as such terms are used in FAR 12.212. Consistent with FAR 12.212 and DoD FAR Supp. 227.7202-1 through 227.7202-4, and notwithstanding any other FAR or other contractual clause to the contrary in any agreement into which this Agreement may be incorporated, a government end user will acquire the Software and Documentation with only those rights set forth in this Agreement. Use of either the Software or Documentation or both constitutes agreement by the government that all Software and Documentation are "commercial computer software" and "commercial computer software documentation," and constitutes acceptance of the rights and restrictions herein. The Software is the subject of the following notices:

* Copyright © 2001 - 2018 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 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, and the International Traffic in Arms regulations (collectively, "**U.S. Export Laws**"), and the United States Foreign Corrupt Practices Act, 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 U.S. 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 unless authorized by the U.S. Government by regulation or specific license.

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.

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 this 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. We will do our best to resolve the issue but if you are still not happy with our response, you may seek to resolve it using the Online Dispute Resolution Platform at www.ec.europa.eu/consumers/odr/.

Last updated 24 May 2018.