



Getting Started

with Episode Podcast

This document guides you through the setup of Episode Podcast, followed by three straight-forward tutorials to get you quickly started using Episode Podcast. The first tutorial shows how to add an additional encoding task to a workflow. The second shows how Episode Podcast enables Podcast Producer to take any format into the workflow. The last tutorial shows how to add a watermark or an intro movie to your media. You will see how easy it is to add Episode Podcast encoding tasks to any Podcast Producer default workflow.

Setup Episode Podcast

Set up your Podcast Producer system. Episode Podcast requires a fully configured Leopard Server and Podcast Producer setup.

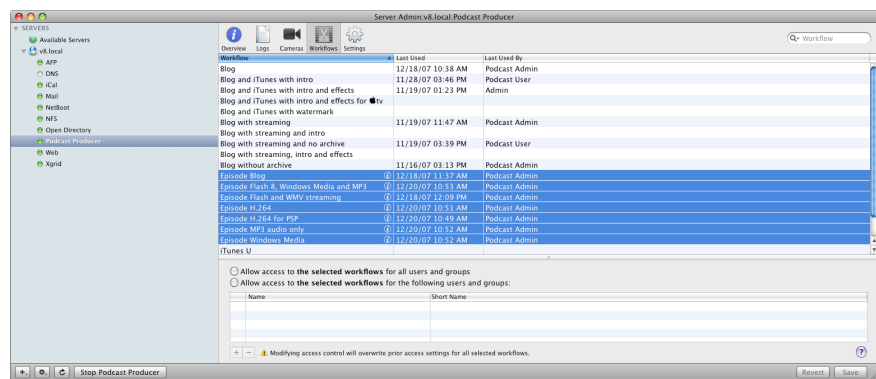
Episode Podcast is delivered as a disk image. The disk image contains the following items: A package installer for Episode Podcast, a folder with the Episode and Episode Podcast manuals, a folder with example workflows, and the client application Episode,.



1. Double-click the package installer to install Episode Podcast. Follow the instructions in the installer.

2. Drag Episode to the Applications folder.
3. Place the `licenses` file you have received in the `/Library/PodcastProducer/Resources/Tools/EpisodePodcast/etc/` folder.
4. Drag the workflows from Example Workflows to the `/Library/PodcastProducer/Workflows` folder.

Note how the example workflows are added to your available workflows visible in the Server Admin application.



You can now test your installation by running for example the Episode Windows Media workflow, which will create a WMV version of your input and place it in the output directory you have designated in Server Admin.

Tutorial 1: Create Windows Media output

This tutorial will show you how to add an encoding task for generating an additional Windows Media output of a different size in the example workflow *Episode Windows Media*. This will give you a workflow that encodes to Windows Media in two different sizes, suitable for distribution to, for example, the web and mobile phones.

Manage your encoding setting

Episode ships with hundreds of predefined settings templates created by encoding experts. You can choose to use one of the predefined settings or customize a setting to suit your specific needs. In this example we will use a predefined settings template.

1. Launch the Episode application.
2. Select a suitable Windows Media setting in the left pane in the Episode application: Use the setting `Compression Settings/Templates/By Format/Windows Media/Download/WMV9_Small`
3. Export the setting so it can be used in Episode Podcast.
 - Right/ctrl click on the setting template `WMV9_Small` and choose *Export* in the context menu.
 - Save the setting, using "Episode Podcast" format, to some suitable folder.
 - Switch to the Finder and move the setting to `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/Encodings`.
4. Quit Episode.

Manage the workflow

The arguments for Episode Podcast are listed in the Appendix of this document (or in the Episode Podcast Administrator's Guide). The property keys for Podcast Producer are listed in the Podcast Producer Admin Guide.

The new Xgrid command you need to add to the tasks has the following form:

```
EpisodePodcast --encoder=WMV9_Small.setting --input=<input movie> --output=<output movie>.wmv
```

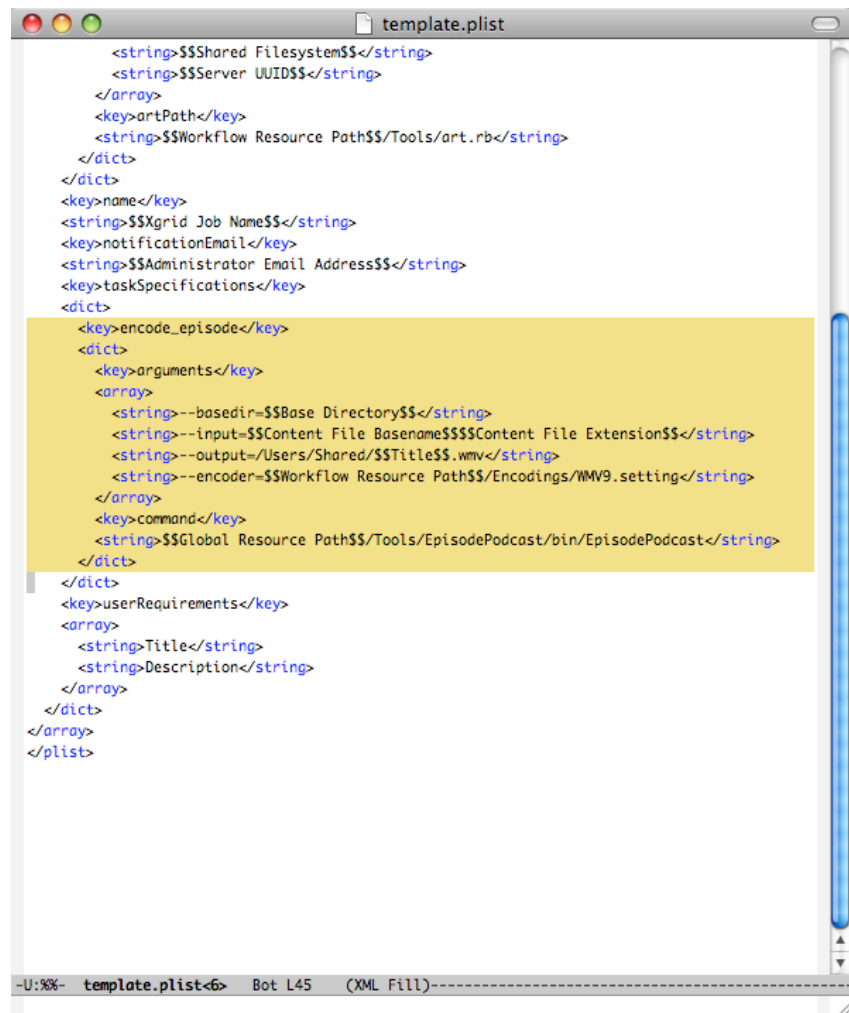
1. Use your favourite text editor to open `template.plist` in the example workflow *Windows Media Episode*.
 - `template.plist` is located in the folder `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/`.
2. Add a new encoding task.

To make this simpler, we will duplicate the task description for an existing entry:

- Locate the existing Episode Podcast encoding task for Windows Media; it starts with:

```
<key>taskSpecifications</key>
<dict>
    <key>encode_episode</key>
```

- Duplicate the Episode encoding task description (see image below).



- Modify the duplicated text as follows:

Rename the task name from encode_episode to encode_windowsmedia_small.

Rename the setting argument to WMV9_Small.setting.

```
<string>--encoder=$$Workflow Resource Path$$/Encodings/
WMV9_Small.setting</string>
```

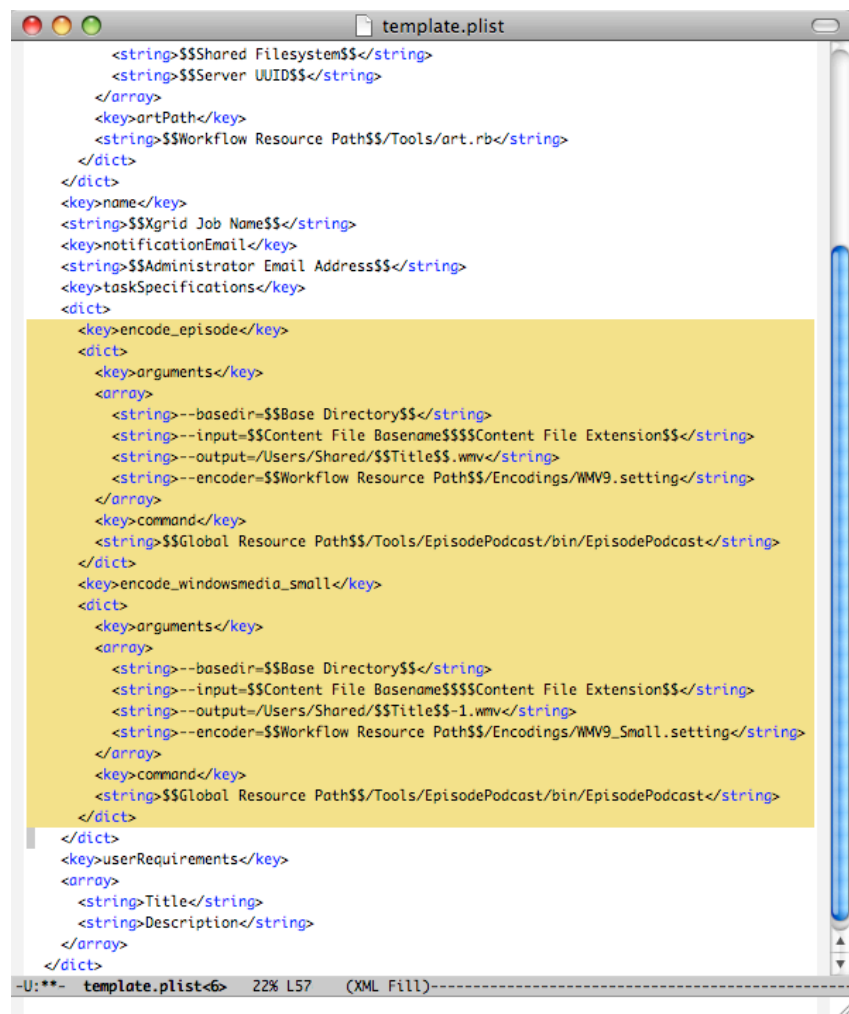
The second output file must have a name distinct from the first output file. The title is defined by the user in Podcast Capture, but you can add elements to it, like -1 as shown below. Both output files are stored in /Users/Shared and have the extension .wmv:

```
<string>--output=/Users/Shared/$$Title$$-1.wmv</string>
```

- Save your edited template.plist.

Verify the workflow

Your modified workflow should look like this:



```

template.plist
<string>$$Shared Filesystem$$</string>
<string>$$Server UUID$$</string>
</array>
<key>artPath</key>
<string>$$Workflow Resource Path$$/Tools/art.rb</string>
</dict>
</dict>
<key>name</key>
<string>$$Xgrid Job Name$$</string>
<key>notificationEmail</key>
<string>$$Administrator Email Address$$</string>
<key>taskSpecifications</key>
<dict>
<key>encode_episode</key>
<dict>
<key>arguments</key>
<array>
<string>--basedir=$$Base Directory$$</string>
<string>--input=$$Content File Basename$$$$Content File Extension$$</string>
<string>--output=/Users/Shared/$$Title$$ .wmv</string>
<string>--encoder=$$Workflow Resource Path$$/Encodings/WMV9.setting</string>
</array>
<key>command</key>
<string>$$Global Resource Path$$/Tools/EpisodePodcast/bin/EpisodePodcast</string>
</dict>
<key>encode_windowsmedia_small</key>
<dict>
<key>arguments</key>
<array>
<string>--basedir=$$Base Directory$$</string>
<string>--input=$$Content File Basename$$$$Content File Extension$$</string>
<string>--output=/Users/Shared/$$Title$$-1.wmv</string>
<string>--encoder=$$Workflow Resource Path$$/Encodings/WMV9_Small.setting</string>
</array>
<key>command</key>
<string>$$Global Resource Path$$/Tools/EpisodePodcast/bin/EpisodePodcast</string>
</dict>
</dict>
<key>userRequirements</key>
<array>
<string>Title</string>
<string>Description</string>
</array>
</dict>
-U:*** template.plist<6> 22% L57 (XML Fill)

```

Record a video

1. Record a video sequence with Podcast Capture.
2. Publish and choose the workflow Windows Media Episode.

View the podcast

View the encoded Windows Media podcast stored in `/Users/Shared/`. If using a Mac, install the free Flip4Mac Windows Media Component for QuickTime (www.microsoft.com/windows/windowsmedia/player/flip4mac.msp). This enables you to view your Windows Media content on your Mac.

Note: You may notice that the video is cropped on the sides. This is actually a parameter that was defined in the Episode setting you exported. This is one of the many parameters you can adjust in Episode.

Tutorial 2: Bring any format into a Podcast Producer workflow

This example demonstrates the capability to submit any leading format into a Podcast Producer workflow, made possible by Episode Podcast.

Manually submit a Windows Media file

1. Launch Podcast Capture and select file submission.
2. Select your previously generated Windows Media (.wmv) file stored in **/Users/Shared** as input file.
3. Publish as H.264.

Choose the “H264 Episode” workflow in Podcast Producer to transcode the Windows Media file to H.264. (It will produce a file with the extension .mp4.)

4. View the output file located in **/Users/Shared/**.

(You can confirm the format of the video track with QuickTime Player.)

Tutorial 3:

Add watermark and intro movie to your Episode workflow

In this exercise you will edit the example workflow *Windows Media Episode* to add a watermark and an intro movie to your Windows Media podcast.

Modify the setting

1. Copy your watermark and intro movie.

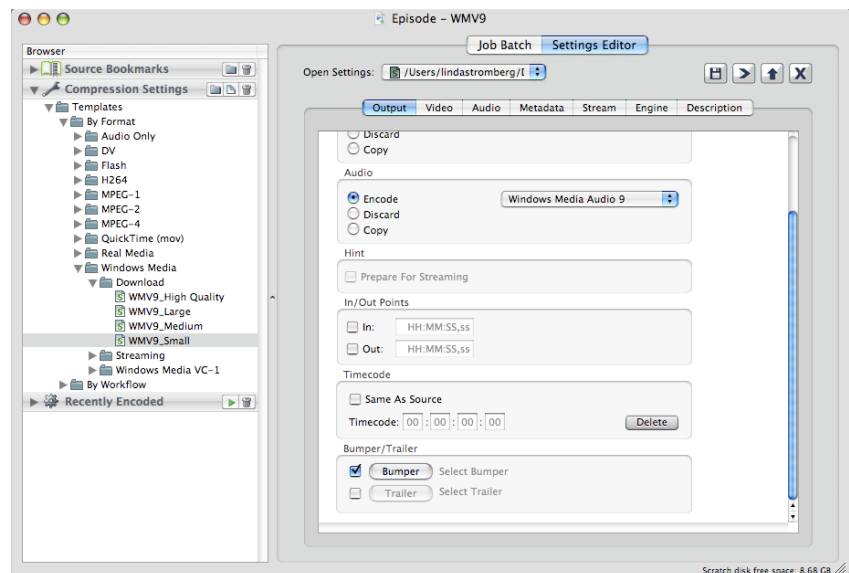
Navigate to the folder `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/`. Create a folder named `Images` and copy the watermark and intro movie to this folder.

2. Modify your setting

- Navigate to the folder `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/Encodings/`.
- Double click the setting `WMV9.setting` to edit it in the Episode application.

3. Add intro and watermark to the setting:

- In the Video tab, activate the watermark filter (without selecting a source).
- In the Output tab, select the Bumper option (without selecting a bumper source).



4. Export the setting to be used in Episode Podcast:

- Export the setting using the export button in the upper right corner.



- Save the setting in “Episode Podcast” format with the name WMV9-modified in a suitable folder.
- Move WMV-9-modified.setting to `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/Encodings`.
- Quit Episode.

Manage the workflow

The new task you need to add to the workflow has the following command line form:

```
EpisodePodcast --encoder=WMV9-modified.setting --
input=<input movie> --watermark=<watermark file> --
intro=<intro movie file> --output=<output movie>
```

1. Use your favourite text editor to open `template.plist` in the example workflow *Windows Media Episode*.

- `template.plist` is located in the folder `/Library/PodcastProducer/Workflows/Episode Windows Media.pwf/Contents/Resources/`.

2. Modify the encoding task.

- Locate the existing Episode Podcast encoding task for Windows Media:

```
<key>taskSpecifications</key>
<dict>
  <key>encode_episode</key>
```

- Add the watermark to the workflow, after the `--encoder` argument. Here we add the image `EpisodePodcastLogo.tif`. Episode accepts watermarks that are images, animations or even movies.

```
<string>--watermark=$$Workflow Resource Path$$/Images/
EpisodePodcastLogo.tif</string>
```

- Add the intro to the workflow. Here we use `UniversityIntro.mov` as an example. The intro can be in any format accepted by Episode.

```
<string>--intro=$$Workflow Resource Path$$/Images/
UniversityIntro.mov</string>
```

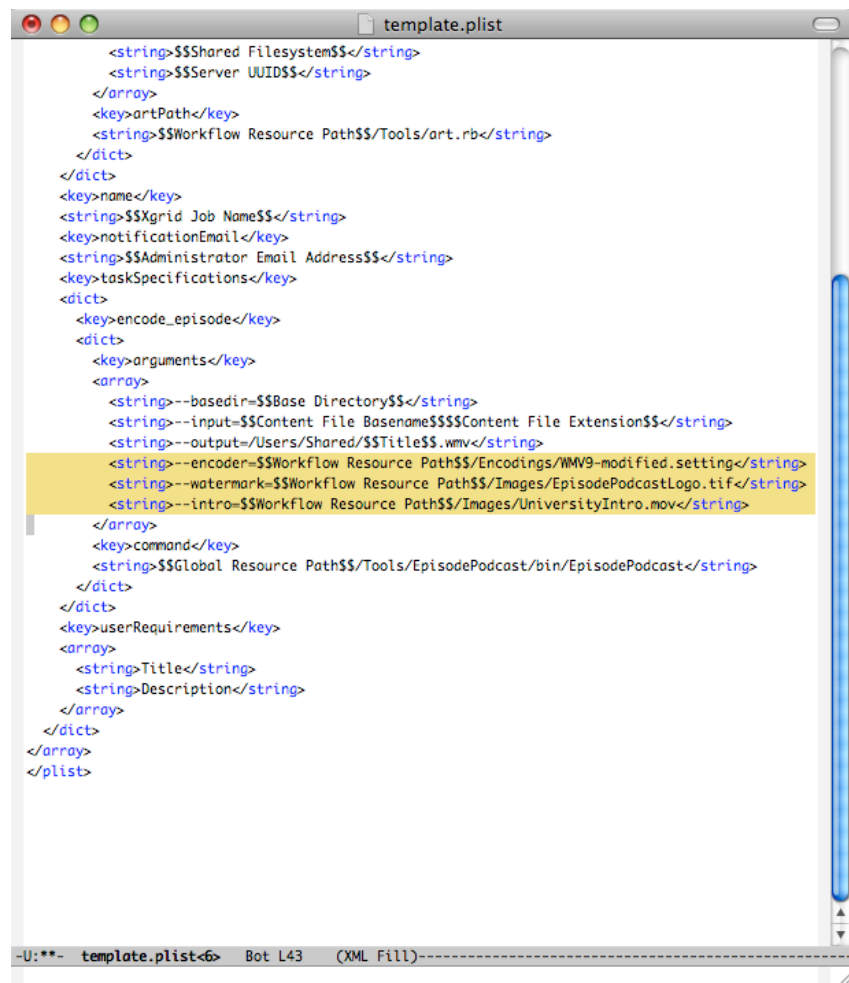
- Ensure that the name of the setting is the same as the setting you want to use.

```
<string>--encoder=${Workflow Resource Path}/Encodings/  
WMV9-modified.setting</string>
```

- Save template.plist.

Verify the modified workflow

Your workflow should look like this:



```
template.plist  
<string>${Shared Filesystem}</string>  
<string>${Server UUID}</string>  
</array>  
<key>artPath</key>  
<string>${Workflow Resource Path}/Tools/art.rb</string>  
</dict>  
</dict>  
<key>name</key>  
<string>${Xgrid Job Name}</string>  
<key>notificationEmail</key>  
<string>${Administrator Email Address}</string>  
<key>taskSpecifications</key>  
<dict>  
<key>encode_episode</key>  
<dict>  
<key>arguments</key>  
<array>  
<string>--basedir=${Base Directory}</string>  
<string>--input=${Content File Basename}${Content File Extension}</string>  
<string>--output=/Users/Shared/${Title}.wmv</string>  
<string>--encoder=${Workflow Resource Path}/Encodings/WMV9-modified.setting</string>  
<string>--watermark=${Workflow Resource Path}/Images/EpisodePodcastLogo.tif</string>  
<string>--intro=${Workflow Resource Path}/Images/UniversityIntro.mov</string>  
</array>  
<key>command</key>  
<string>${Global Resource Path}/Tools/EpisodePodcast/bin/EpisodePodcast</string>  
</dict>  
</dict>  
<key>userRequirements</key>  
<array>  
<string>Title</string>  
<string>Description</string>  
</array>  
</dict>  
</array>  
</plist>
```

Record a video

1. Record a video sequence with Podcast Capture.
2. Publish and choose the workflow Windows Media Episode.

View the podcast

View the Windows Media podcast in `/Users/Shared/` and verify that your watermark and intro movie have been added to the media.

Create a new workflow bundle in Podcast Producer

To create a new workflow, duplicate an existing workflow and follow the instructions below to make the required modifications (according to the Podcast Producer Admin Guide):

1 Duplicate a workflow bundle and store it in /Library/PodcastProducer/Workflows.

2 Add, delete, and modify tasks and custom properties as needed.

3 Configure the value of the new properties in Server Admin.

- Change the value of the **CFBundleName** property in the `<workflow_name>/Contents/Info.plist` file to the name of the new workflow bundle.

- Change the value of the **CFBundleIdentifier** property in the `<workflow_name>/Contents/Info.plist` file to the appropriate value.

- Change the **Name** and **Description** properties in the `<workflow_name>/Contents/Resources/<language>.lproj/InfoPlist.strings` file as appropriate.

4 Verify that you can see the new workflow in Server Admin.

For more information

For more information about Telestream and Episode Podcast visit

www.episodepodcast.com

For more information about Apple's Leopard Server and Podcast Producer visit

www.apple.com/server/macosx/

Getting started with Episode Podcast

Appendix: Episode Podcast Arguments

Episode Podcast takes the following arguments for creating tasks within the Podcast Producer workflows, defined in `template.plist`

-a <string1> <string2> | --replace <string1> <string2> <string1> is replaced by <string2> in the settings file. As the settings file syntax may change between versions this argument should be used with some care. Some potentially useful strings to replace are:

!SRCFILE! The absolute filename of the input file.

!DSTPATH! The absolute path to the output file.

!NAME! The basename of the input file. (Used when constructing the name of the output file.)

!SETTING! The basename of the settings file. (Used when constructing the name of the output file.)

-b <prefix> | --basedir <prefix> The <prefix> will be used as a path prefix for all relative path arguments. Example:

```
EpisodePodcast --basedir /tmp --input=file.mov
```

```
--encoder=/scratch/WMV.setting
```

will look for `/tmp/file.mov` and `/scratch/WMV.setting`.

-f <file> | --file <file> <file> is the `engine.conf` file. It is used by Episode Podcast for certain internal configuration parameters. The <file> path is not affected by the `--basedir` argument. You should normally never need to use this argument.

-h | --help Print out a list of all supported arguments.

-k <file> | --watermark <file> <file> is a watermark file. Watermark files can be BMP, GIF, JPEG, QuickTime, Targa or TIFF files. GIF animations and QuickTime videos can be used for animated watermarks. See the Episode manual for a fuller discussion of how to use watermarks. If a watermark has been specified in the settings file this argument is mandatory.

-l <low> <high> | --limit <low> <high> The internal buffers in the encoding process should be limited to avoid excessive memory use. <low> and <high> are the low- and high-water marks for the number of image frames that are allowed to be buffered between individual encoding stages. The default values are 4 and 8, respectively; only under exceptional circumstances should you change these values.

-n <file> | --input <file> <file> is the input file. This argument is mandatory.

-o <file> | --output <file> <file> is the output file. If no output file is specified the output will be placed in the Desktop directory with a name constructed from the basename of the input file and the basename of the settings file.

-p | --preflight Check that all input files are correct, but do not perform the actual transcoding. You should normally never need to use this argument.

-w <time> | --wait <time> Wait <time> seconds before exiting after transcoding is finished. You should normally never need to use this argument.

-x <file> | --encoder <file> <file> is the settings file. This argument is mandatory.

-y <file> | --intro <file> <file> is an intro video file. If an intro has been specified in the settings file this argument is mandatory.

-z <file> | --outro <file> <file> is an outro video file. If an outro has been specified in the settings file this argument is mandatory.