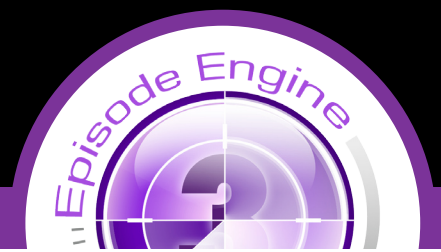
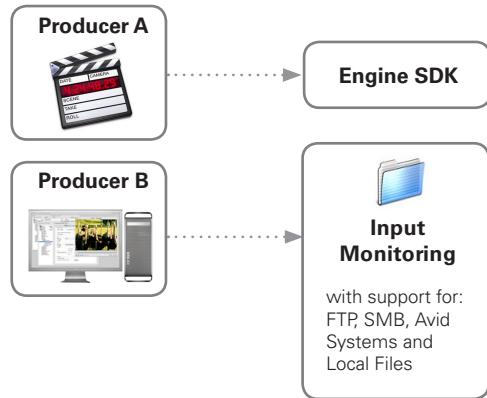


# Episode Engine

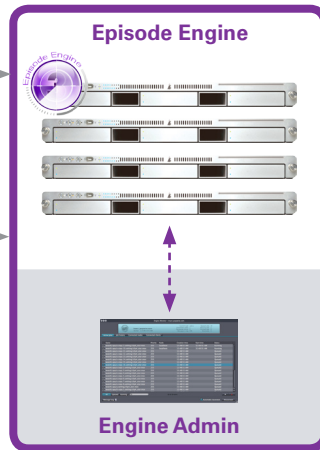
Integrated with your workflow



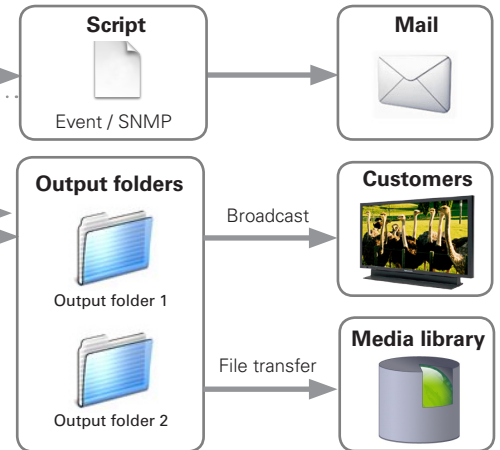
## Acquisition



## Transcoding



## Deployment



## How to maximize your Episode Engine value in your workflow

**Episode® Engine is a powerful and flexible tool adaptable to your workflow. With Episode Engine each step in your workflow - acquisition, transcoding and deployment - can be automated to meet your exact needs.**

### Acquisition

Episode Engine supports input monitoring (watch folders). With the new and easy-to-use Engine Admin interface, you can quickly create and administrate your input monitors. Simply configure what source directory to be monitored and as soon as an input file is placed in this directory, Episode Engine will retrieve it and start transcoding. Source material can be read from local files or remote file servers via for example FTP, Samba and NFS.

The input monitors can be given priorities so that high-priority media are transcoded before other. This enables separation of time-critical and non-time-critical transcoding as well as opening for premium services.

Another alternative acquisition method is the **Episode Engine SDK**. The SDK provides the tools you need to integrate Episode Engine into your workflows. It consists of several interfaces from command-line (for Mac OSX) to platform independent web services.

With the **Engine Admin** application you can easily adjust the order of execution by changing the priority of jobs or completely removing them from the execution queue. The Engine Admin can also communicate with multiple Episode Engine installations simultaneously. For example it allows administrators to manage clusters serving different geographical regions.

### Transcoding

The transcoding process not only allows you to go from one format to another, but also lets you add intro and outro sequences, visual watermarks and metadata to the output media. The input files for these enhancements are placed in watch folders just like your media files. Through a simple naming convention they can be designated as either global or specific for a given input media file, thus adaptable to special requirements with minimal effort. Global metadata can be supplied as part of the transcoding settings, but generally XML files are used for metadata input. The order for adding input files does not matter; Episode Engine waits until all required files are present.

You can interactively monitor the progress of transcoding with the Engine Admin. It shows the expected transcoding times, when files have been successfully transcoded and any problems that might have occurred.



# Episode Engine

Integrated with your workflow

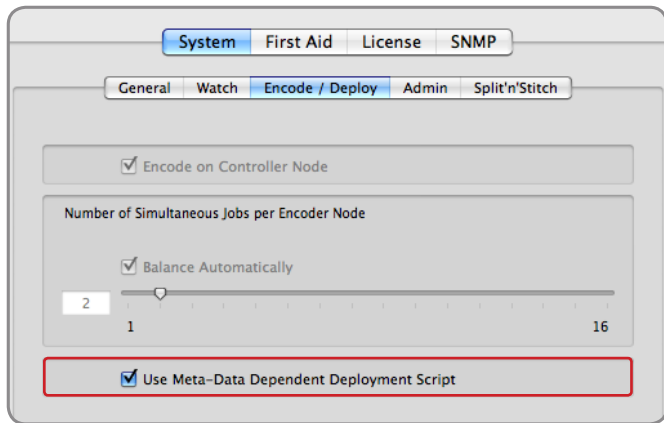


## Deployment

The output of transcoding jobs is placed in output folders in a structure corresponding to that of the input watch folders. The output folders can, just as input folders, be readable to authorized users only and be aliases of folders placed elsewhere in the file system.

Further automation of your workflow output deployment can be achieved with metadata driven event action scripts.

You can write these scripts yourself, using whatever scripting language you prefer, but if you enable the checkbox "Use Meta-Data Dependent Deployment Script" under the Episode Engine tab in System Preferences, Engine will scan the meta-data associated with a transcoding job for certain keywords which will then be used to perform certain deployment actions.



In this way you can create simple event scripts without any programming at all. The keywords can be supplied through the *Engine* tab in **Episode** as well as through an `.inmeta` file, thus allowing for different actions depending on the file. Note: The order of keywords is not important.

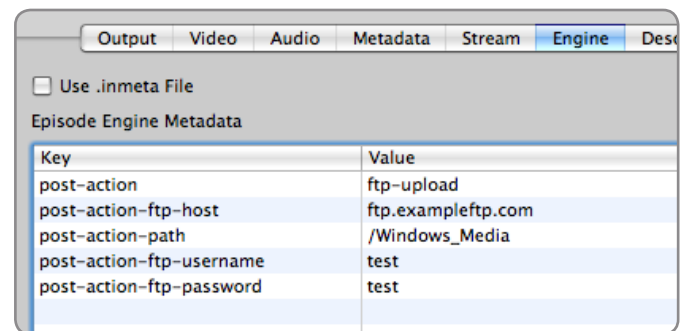
The *Key* post-action defines the action to be performed. The possible values are:

- copy
- ftp -upload

copy will copy the output file to a directory specified by the key `post-action-path`.

ftp-upload will upload the output file to an ftp server. This action also requires input values for:

- `post-action-ftp-host`: the host name of the server
- `post-action-path`: the directory on the server where the file is to be stored
- `post-action-ftp-username`: username for logging in on the server
- `post-action-ftp-password`: password for logging in on the server



Both commands will also take optional parameters: `post-action-transfer-meta` specifies whether any `.meta` file should be transferred along with the output file. Allowed values are yes and no, default is no.

`post-action-delete-files` specifies if the output file and any `.meta` file should be deleted from the output folder after having been copied/uploaded. Allowed values are yes and no, default is no.

You can of course extend the script with any additional commands you wish to implement.

## For more information - visit [Telestream.net](http://Telestream.net)

Specifications subject to change without notice. Copyright © 2007 Telestream, Inc. Telestream and Episode are registered trademarks of Telestream, Inc. All other trademarks are the property of their respective owners.

**Contact us at:**  
Stockholm, Sweden  
Nevada City, CA USA  
tel +1 530 470 1300  
[www.telestream.net](http://www.telestream.net)  
[episodesales@telestream.net](mailto:episodesales@telestream.net)



TELESTREAM