



# **Engine API Document**

**v.7.6.5**

*10th October 2018*

## Index

<b>API Overview</b>	<b>4</b>
<b>'info' Overview</b>	<b>5</b>
<b>'jobs' Overview</b>	<b>6</b>
<b>'workflows' Overview</b>	<b>7</b>
<b>'folders' Overview</b>	<b>8</b>
<b>Posting a Job</b>	<b>9</b>
<b>Getting Job Information From a Workflow</b>	<b>10</b>
<b>Job Status and Parameters</b>	<b>11</b>
<b>Workflow Parameters</b>	<b>13</b>
<b>Info Parameters</b>	<b>14</b>
<b>Folder Status and Parameters</b>	<b>15</b>
<b>Rechecking and Resetting Folders</b>	<b>17</b>
<b>Deleting a Job</b>	<b>18</b>
<b>Changing the Priority of a Job</b>	<b>19</b>
<b>Getting a Job Report</b>	<b>20</b>
<b>Importing and Exporting Workflows</b>	<b>21</b>
<b>Appendix A - Job Post XML Schema</b>	<b>22</b>
<b>Appendix B - Job Post Example XML</b>	<b>25</b>
<b>Appendix C - Job Post Response XML</b>	<b>26</b>
<b>Appendix D - Workflow Required XML Schema</b>	<b>27</b>
<b>Appendix E - Workflow Required Example XML</b>	<b>30</b>
<b>Appendix F - Basic Job Request Example Xml</b>	<b>31</b>
<b>Appendix G - Status Job Request Example Xml</b>	<b>32</b>
<b>Appendix H - Detail Job Request Example Xml</b>	<b>33</b>
<b>Appendix I - Basic Workflow Request Example Xml</b>	<b>34</b>
<b>Appendix J - Summary Workflow Request Example Xml</b>	<b>35</b>

<b>Appendix K - Detail Workflow Request Example Xml</b>	<b>36</b>
<b>Appendix L - Info Request Example Xml</b>	<b>38</b>
<b>Appendix M - Basic Folder Request Example Xml</b>	<b>39</b>
<b>Appendix N - Status Folder Request Example Xml</b>	<b>40</b>
<b>Appendix O - Path Folder Request Example Xml</b>	<b>41</b>
<b>Appendix P - Detail Folder Request Example Xml</b>	<b>42</b>
<b>Appendix Q - Recheck Folder Post and Response Example Xml</b>	<b>44</b>
<b>Appendix R - Reset Folder Post and Response Example Xml</b>	<b>45</b>
<b>Appendix S - Deleting Jobs and Response Example Xml</b>	<b>46</b>
<b>Appendix T - Changing Job Priority and Response Example Xml</b>	<b>47</b>
<b>Appendix U - Error Message Format and Example Xml</b>	<b>48</b>
<b>Appendix V - Report Job Request Example Xml</b>	<b>49</b>

## API Overview

The Engine API follows a RESTful architectural style. It is intended to integrate seamlessly into highly automated operations and to provide simple control and status information.

The primary interaction allows simple HTTP access to the system. The API behaviour is asynchronous, and is accessed by URIs using standard HTTP GET, POST, PUT and DELETE methods. There is no encapsulation or additional protocol.

The Eflow component of Engine runs as a HTTP server. It receives and handles all HTTP requests. Eflow runs as a native Windows Service on Windows and as a Daemon on OS X and Linux. It monitors port 44544.

All commands should be sent in XML format. Responses will also be in XML format.

Access to specific resources is based on a URL in the form:

*http://<hostname or IP>:<portNumber>/<requestURI>*

where <requestURI> can be one of the following:

- info
- jobs
- workflows
- folders

## 'info' Overview

This simply stores information relating to the Engine system.

The following commands can be run relating to the 'info' resource:

HTTP	URI	Description
GET	info	Returns information on the current system including the software version number and the number ESPs available

## ‘jobs’ Overview

A list of jobs is maintained by Eflow. Each job is assigned a uuid which is used to track and access information relating to it. This uuid can be defined by the user or if left blank will be generated by Eflow.

The following commands can be run relating to the ‘jobs’ resource:

HTTP	URI	Description
POST	jobs	Adds a job to the job list maintained by Eflow
GET	jobs	Returns a list of all jobs
GET	jobs/status	Returns a list of each job and the status of each job
GET	jobs/detail	Returns a list of each job and the status and parameters of each job
GET	jobs/<jobUuid>/status	Returns the status for the given <jobUuid>
GET	jobs/<jobUuid>/detail	Returns the status and parameters for the given <jobUuid>
GET	jobs/<jobUuid>	Same as <b>jobs/&lt;jobUuid&gt;/detail</b> . Returns the status and parameters for the given <jobUuid>
GET	jobs/<jobUuid>/report	Returns the xml report for the given <jobUuid> (provided that the job has been completed, a xml report has been generated and the report has not been moved from the location specified in the job parameters)
PUT	jobs/<jobUuid>	This can be used to edit the parameters for the given <jobUuid>
DELETE	jobs	Deletes all jobs from list
DELETE	jobs/list	Delete a list of jobs
DELETE	jobs/<jobUuid>	Deletes job with the given <jobUuid>

## ‘workflows’ Overview

Workflows are created and edited using the Engine Configurator component of Engine. Each workflow is assigned a uuid which is used to track and access information relating to it. Eflow can return a list of workflows and a summary of information regarding source, destination and signal processing parameters. Eflow can also return the uuids and tags required by the workflow which must be used when creating the xml posted to the ‘jobs’ resource.

The following commands can be run relating to the ‘workflows’ resource:

HTTP	URI	Description
GET	workflows	Returns a list of all workflows
GET	workflows/required	Returns a list of each workflows and the required parameters for each workflow
GET	workflows/summary	Returns a list of each workflows and a summary for each workflow
GET	workflows/detail	Returns a list of each workflows and a summary and the required parameters for each workflow
GET	workflows/<workflowUuid>/required	Returns the required parameters for the given <workflowUuid>
GET	workflows/<workflowUuid>/summary	Returns a summary for the given <workflowUuid>
GET	workflows/<workflowUuid>/detail	Returns a summary and the required parameters for the given <workflowUuid>
GET	workflows/<workflowUuid>	Same as <b>workflows/&lt;workflowUuid&gt;/detail</b> . Returns a summary and the required parameters for the given <workflowUuid>
GET	workflows/export	Returns all the workflows in an xml format that can be used to import them into another system (this should only be used in an extremely controlled environment and the xml should never be edited by an external program)
GET	workflows/<workflowUuid>/export	Returns the given <workflowUuid> in an xml format that can be used to import it into another system (this should only be used in an extremely controlled environment and the xml should never be edited by an external program)
PUT	workflows/import	Allows workflows to be imported from another Engine system (provided xml is in correct format)

## 'folders' Overview

Watch Folders are created and edited using the Engine Configurator component of Engine. A list of the watch folders is maintained by Eflow. Each folder is assigned a uuid which is used to track and access information relating to it, and well as status information.

The following commands can be run relating to the 'folders' resource:

HTTP	URI	Description
POST	folders	Posts an action to perform on folders
GET	folders	Returns a list of all folders
GET	folders/status	Returns a list of each folder and the status of each folder
GET	folders/detail	Returns a list of each folder and the status and parameters and directory paths of each folder
GET	folders/paths	Returns a list of each folder and the directory paths of each folder
GET	folders/active	Returns a list of all active folders
GET	folders/<folderUuid>/status	Returns the status for the given <folderUuid>
GET	folders/<folderUuid>/detail	Returns the status and parameters and directory paths for the given <folderUuid>
GET	folders/<folderUuid>/paths	Returns the directory paths for the given <folderUuid>
GET	folders/<folderUuid>	Same as <b>folders/&lt;folderUuid&gt;/detail</b> . Returns the status and parameters and directory paths for the given <folderUuid>



## Posting a Job

The following XML parameters must be defined in order to post a job.

Xml Tag	Attribute	Description
Job	uuid	Job Uuid. This can be passed in by user or left blank for Eflow to generate.
Job	title	Job Title. Option files to assign a title to the job.
Job	userFlag	Job User Flag. Optional flag to assign user defined information to the job.
Job	priority	Job Priority. If this is set to "true", then the job will be placed at the front of the job queue.
Job	version	Job Version. This must be set to "7" for this version.
Source	-	Source information for the job.
Destination	-	Destination information for the job.
Report	-	Report information for the job.
Workflow	uuid	Workflow uuid.

A detailed schema for the xml format can be found in **Appendix A - Job Post XML Schema**.

An example of a job post xml can be found in **Appendix B - Job Post Example XML**.

An explanation and example of the xml response can be found in **Appendix C - Job Post Response XML**

## Getting Job Information From a Workflow

The tags and information sent in the 'Source' and 'Destination' tags will vary depending on which workflow is specified.

The **GET workflows/<workflowUuid>/required** command returns the tags and information needed in order to create a valid job post for the given workflow.

It will specify the number of sources/destinations and their uuids, as well as filetypes and optional metadata tags.

A detailed schema for the xml format can be found in ***Appendix D - Workflow Required XML Schema.***

An example of a job post xml can be found in ***Appendix E - Workflow Required Example XML.***

## Job Status and Parameters

Information from the jobs is split into three sections:

- Basic (returned in list **GET jobs**)
- Status (returned when status is requested **GET jobs/status**)
- Detail (returned when detail is requested **GET jobs/detail**)

The following gives more detail:

Section	Xml Tag	Attribute	Description
Basic	Job	uuid	Gives the Job Uuid
Basic	Job	title	Gives the Job title
Basic	Job	watchfolderProfileName	The name of the watchfolder the file was posted into. Empty if not posted via watchfolder.
Basic	Job	watchfolderProfileId	The watchfolder Uuid of the watchfolder the file was posted into. {00000000-0000-0000-0000-000000000000} if not posted via watchfolder.
Status	Job	state	Gives a state code number: 0 - SUCCESS 1 - ERROR 2 - QUEUED 3 - PROCESSING 4 - LICENSE 5 - ABORTED 6 - ERROR 7 - GROWING 8 - REGISTERED 9 - FOLDER ACTIONS 10 - INSPECTING
Status	Job	stateDescription	Gives a text string associated with the give state code number. Presently these are: <ul style="list-style-type: none"> <li>• SUCCESS - Job completed successfully</li> <li>• ERROR - Job completed with an error. See detailDescription which describes the error</li> <li>• QUEUED - Job is in the queue waiting to be processed</li> <li>• PROCESSING - Job is currently being processed</li> <li>• LICENSE - Job experienced license error.</li> <li>• ABORTED - Job was aborted by user.</li> <li>• GROWING – File in watchfolder is still being copied.</li> <li>• REGISTERED - Job has been registered and stored in database.</li> <li>• INSPECTING – File is being inspected to make sure that copying has finished.</li> <li>• FOLDER ACTIONS - File in watch folder is being moved/renamed/copied.</li> </ul>
Status	Job	detail	Gives a code number associated with the current status of the job

Section	Xml Tag	Attribute	Description
Status	Job	detailDescription	Gives a text string describing the current status or error condition.
Status	Job	progress	This gives the progress of the file while it is in a given state as a value between 0 and 100.
Status	Job	workflow	Gives the Workflow Uuid
Status	Job	startTime	Gives the date and time that the job starting processing in Engine.
Status	Job	finishTime	Gives the date and time that the job finished processing in Engine.
Status	Job	postTime	Gives the date and time that the job was posted to Engine.
Status	Job	detectedTime	Gives the date and time that the job was detected in the watchfolder.
Status	Job	workflowName	Gives the Workflow name
Status	Job	ipAddress	Gives the IP Address that posted the job to Engine
Status	Job	hostName	Gives the Host Name that posted the job to Engine
Status	Job	userFlag	Gives the Job User Flag
Status	Job	priority	Gives the Job Priority.
Detail	Source	-	Returns source information as defined in post
Detail	Destination	-	Returns destination information as defined in post
Detail	Report	-	Returns report information as defined in post
Detail	AdvancedStatus	-	Returns advanced status information

An example of a basic GET request can be found in ***Appendix F - Basic Job Request Example Xml.***

An example of a status GET request can be found in ***Appendix G - Status Job Request Example Xml.***

An example of a detail GET request can be found in ***Appendix H - Detail Job Request Example Xml.***

## Workflow Parameters

Information from the workflows is split into three sections:

- Basic (returned in list **GET workflows**)
- Summary (returned when summary is requested **GET workflows/summary**)
- Required (returned when required is requested **GET workflows/required**)

The Required section is explained in section **Getting Job Information From a Workflow** and so will be skipped in this section.

The following gives more detail:

Section	Xml Tag	Attribute	Description
Basic	Workflow	uuid	Gives the Workflow Uuid
Basic	Workflow	title	Gives the Workflow title
Summary	Summary	-	Returns information regarding the workflow type, source, destination and any signal processing steps in the workflow

An example of a basic GET request can be found in **Appendix I - Basic Workflow Request Example Xml**.

An example of a summary GET request can be found in **Appendix J - Summary Workflow Request Example Xml**.

A request for detail GET request will return basic, summary and required information. An example can be found in **Appendix K - Detail Workflow Request Example Xml**

## Info Parameters

This is returned when **GET info**. It returns information regarding Engine and the system it is running on.

Xml Tag	Description
Application	Application name
Version	Application version
Pid	Pid assigned to the application by the system
ESPsAvailable	Number of ESPs that Eflow has detected
WatchfoldersActive	Number of active watchfolders
WatchfoldersAvailable	Total number of watchfolders that can be active
TimeSinceStart	Time since Application was last restarted
VolumeRootPath	Volume root that application is running on
VolumeReadOnly	If the Volume that application is running on is read-only, this is set to 'true'
VolumeName	Name of the Volume that the application is running on
VolumeFileSystemType	System type of the Volume that the application is running on
VolumeSizeMB	Size in MB of the Volume that the application is running on
VolumeAvailableSizeMB	Bytes Available in MB of the Volume that the application is running on
HostName	Host Name of the system that the application is running on
HostAddress	Host address of the system that the application is running on
OSType	OS type of the system that the application is running on
OSVersion	OS version of the system that the application is running on
CpuArchitecture	CPU architecture of the system that the application is running on

An example of an info GET request can be found in **Appendix L - Info Request Example Xml**.

## Folder Status and Parameters

Information from the folders is split into three sections:

- Basic (returned in list **GET folders**)
- Status (returned when status is requested **GET folders/status**)
- Paths (returned when status is requested **GET folders/paths**)
- Detail (returned when detail is requested **GET folders/detail**)

The following gives more detail:

Section	Xml Tag	Attribute	Description
Basic	Folder	uuid	Gives the Folder Uuid
Basic	Folder	title	Gives the Folder title
Basic	Folder	active	Gives the active state of the Folder
Basic	Folder	workflow	Gives the Workflow Uuid assigned to the Folder
Basic	Folder	workflowName	Gives the Workflow title assigned to the Folder
Basic	Folder	shortcut	Indicates if the Folder is a Shortcut Watchfolder
Status	Status	active	Gives the active state of the Folder
Status	Status	valid	Gives the valid state of the Folder
Status	Status	numFiles	Gives the number of files currently being watched by the Folder
Status	Status	state	Gives a state code number: 0 - IDLE 1 - ERROR 2 - SWEEPING 3 - RUNNING 4 - STOPPED
Status	Status	stateDescription	Gives a text string associated with the give state code number. Presently these are: <ul style="list-style-type: none"> <li>• IDLE - Folder has no error but is not sweeping</li> <li>• ERROR - Folder experienced an error and is no longer sweeping</li> <li>• SWEEPING - Folder is searching for files</li> <li>• RUNNING - Folder is running and waiting for next sweep</li> <li>• STOPPED - Folder has been stopped and will no longer sweeping for files</li> </ul>
Status	Status	lastSweepTime	Gives the time the last sweep of files occurred
Paths	Directory	path	Gives the directory path

Section	Xml Tag	Attribute	Description
Paths	Directory	folderType	Gives the folder type that the directory refers to. Presently these are: <ul style="list-style-type: none"> <li>• Source</li> <li>• Source_Move</li> <li>• Destination</li> <li>• Report</li> <li>• Error</li> <li>• Shortcut</li> <li>• Shortcut_Error</li> <li>• Shortcut_Move</li> </ul>
Paths	Directory	-	Gives information as to whether directory is “valid” or “invalid”
Detail	-	-	Returns Folder settings as xml tags

An example of a basic GET request can be found in **Appendix M - Basic Folder Request Example Xml.**

An example of a status GET request can be found in **Appendix N - Status Folder Request Example Xml.**

An example of a paths GET request can be found in **Appendix O - Paths Folder Request Example Xml.**

An example of a detail GET request can be found in **Appendix P - Detail Folder Request Example Xml.**



## Rechecking and Resetting Folders

Two actions can be posted to the folders

- Recheck
- Reset

A Recheck will signal Eflow to re-examine all folders, check validity and active states.

An example of a Recheck post and the response can be found in ***Appendix Q - Recheck Folder Post and Response Example Xml.***

A Reset will signal Eflow to stop all watchfolders, remove all records previously submitted from the folder (including those submitted to the job list) and clear all databases recording information regarding files being watched. The action can not be undone.

An example of a Recheck post and the response can be found in ***Appendix R - Reset Folder Post and Response Example Xml.***

## Deleting a Job

A **DELETE** can be called on a specific job, a list of jobs or all jobs.

Once a job has been deleted, all record of the job will be removed.

The action can not be undone.

An example of deleting jobs and the response can be found in ***Appendix S - Deleting Jobs and Response Example Xml.***

## Changing the Priority of a Job

The priority of a job can be changed by sending a **PUT** request to a given <jobUuid>

**PUT /jobs/<jobUuid>**

Once submitted, if priority is set to “true”, the job will move to the top of the job queue. If the priority is set to “false”, the job will be moved out of the priority position of the job queue.

An example of changing the job priority and the response can be found in **Appendix T - Changing Job Priority and Response Example Xml.**

## Getting a Job Report

Provided that the job has been completed, a xml report has been generated and the report has not been moved from the location specified in the job parameters, then the summary xml report can be requested for a given <jobUuid>

**GET /jobs/<jobUuid>/report**

An example of a report GET request can be found in ***Appendix V - Report Job Request Example Xml.***

## Importing and Exporting Workflows

The export command

**GET workflows/export** and **GET workflows/<workflowUuid>/export**

can be used to export workflows from one system and into another using

**PUT workflows/import**

The data should not be modified in any way. The response of the export command should be copied directly in to import command.

## Appendix A - Job Post XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!--Uuid-->
  <xs:simpleType name="Uuid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-fo-9]{32}(\{\|\O?[A-Fa-fo-9]{8}-([A-Fa-fo-9]{4}-){3}[A-Fa-fo-9]{12}(\}\|\}))?"/>
    </xs:restriction>
  </xs:simpleType>

  <!--OnOff-->
  <xs:simpleType name="onOff">
    <xs:restriction base="xs:string">
      <xs:enumeration value="on"/>
      <xs:enumeration value="off"/>
    </xs:restriction>
  </xs:simpleType>

  <!--TrueFalse-->
  <xs:simpleType name="trueFalse">
    <xs:restriction base="xs:string">
      <xs:enumeration value="true"/>
      <xs:enumeration value="false"/>
    </xs:restriction>
  </xs:simpleType>

  <!--Pattern for Report-->
  <xs:complexType name = "ReportParameter">
    <xs:sequence>
      <xs:element name="ReportPath" type="xs:string" maxOccurs="1"/>
      <xs:element name="ReportType" maxOccurs="1">
        <xs:complexType>
          <xs:attribute name="xml" type="onOff" use="required"/>
          <xs:attribute name="pdf" type="onOff" use="required"/>
          <xs:attribute name="csv" type="onOff" use="required"/>
        </xs:complexType>
      </xs:element>
      <xs:element name="ReportName" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" maxOccurs="1">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Base" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <!--Pattern for Source-->
  <xs:complexType name = "SourceParameter">
    <xs:sequence>
      <xs:element name="Id" maxOccurs="unbounded">
        <xs:complexType mixed="true">
          <xs:sequence>
            <xs:element name="File" maxOccurs="unbounded">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Name" maxOccurs="1">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Base" type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="Path" type="xs:string" maxOccurs="1"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="ReferenceTimecode" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:element name="Segment" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="SOM" type="xs:string" minOccurs="0" maxOccurs="1"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:element name="DUR" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="EOM" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="uuid" type="Uuid" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<!--Pattern for Destination-->
<xs:complexType name = "DestinationParameter">
<xs:sequence>
    <xs:element name="Id" maxOccurs="unbounded">
        <xs:complexType mixed="true">
            <xs:sequence>
                <xs:element name="File" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Name" maxOccurs="1">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="Base" type="xs:string"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Path" type="xs:string" maxOccurs="1"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Timecode" type="xs:string" minOccurs="0" maxOccurs="1"/>
                <xs:element name="StreamTagging" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Stream" minOccurs="0" maxOccurs="64">
                                <xs:complexType>
                                    <xs:attribute name="number" type="xs:string" use="required"/>
                                    <xs:attribute name="languageTag" type="xs:string" use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="ChannelTagging" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Channel" minOccurs="0" maxOccurs="64">
                                <xs:complexType>
                                    <xs:attribute name="number" type="xs:string" use="required"/>
                                    <xs:attribute name="channelTag" type="xs:string" use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Segment" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="SOM" type="xs:string" minOccurs="0" maxOccurs="1"/>
                            <xs:element name="DUR" type="xs:string" minOccurs="0" maxOccurs="1"/>
                            <xs:element name="EOM" type="xs:string" minOccurs="0" maxOccurs="1"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
            <xs:attribute name="uuid" type="Uuid" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>

<!--Main-->
<xs:element name="Jobs">
<xs:complexType>
<xs:sequence>
    <xs:element name="Job" minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
            <xs:sequence>

```

```
<xs:element name="Workflow">
  <xs:complexType>
    <xs:attribute name="uuid" type="Uuid" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Source" type="SourceParameter" minOccurs="1" maxOccurs="1"/>
<xs:element name="Destination" type="DestinationParameter" minOccurs="0" maxOccurs="1"/>
<xs:element name="Report" type="ReportParameter" minOccurs="0" maxOccurs="1"/>
</xs:sequence>
<xs:attribute name="uuid" type="Uuid"/>
<xs:attribute name="title" type="xs:string"/>
<xs:attribute name="userFlag" type="xs:string"/>
<xs:attribute name="priority" type="trueFalse"/>
<xs:attribute name="version" type="xs:string" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```



## Appendix B - Job Post Example XML

```
POST http://localhost:44544/jobs HTTP/1.1
Content-Type: text/xml
Host: localhost:44544
<Jobs>
  <Job uuid="{1ff36f34-538e-44a4-afb7-e2f45313307e}" title="JOB1" userFlag="User1" priority = "true" version = "7">
    <Workflow uuid="{1ff36f34-538e-44a4-afb7-e2f45313307e}"/>
    <Source>
      <Id uuid="{253a2d58-c82d-4ae0-bc9d-f5dd055a9ce2}">
        <File>
          <Name>
            <Base>Test.mxf</Base>
          </Name>
          <Path>/Users/User/SourceFiles/</Path>
        </File>
        <ReferenceTimecode>09:59:30:00</ReferenceTimecode>
      </Id>
      <Id uuid="{73f4b33b-988a-4f64-bfc8-70f15279cab9}">
        <File>
          <Name>
            <Base>Test.wav</Base>
          </Name>
          <Path>/Users/User/SourceFiles/</Path>
        </File>
        <Segment>
          <SOM>10:00:00:00</SOM>
        </Segment>
      </Id>
    </Source>
    <Destination>
      <Id uuid="{a1a025bc-c928-42a8-bf99-057ecad812c0}">
        <File>
          <Name>
            <Base>Test-completed.mxf</Base>
          </Name>
          <Path>/Users/User/DestinationFiles/</Path>
        </File>
        <Timecode>10:00:00:00</Timecode>
        <StreamTagging>
          <Stream number="1" languageTag="ENG"/>
          <Stream number="2" languageTag="SPA"/>
          <Stream number="3" languageTag="FRE"/>
          <Stream number="4" languageTag="JAP"/>
          <Stream number="5" languageTag="GER"/>
          <Stream number="6" languageTag="KOR"/>
        </StreamTagging>
      </Id>
      <Id uuid="{07cb1e59-7766-40a8-bb1c-d8a493cecb2f}">
        <File>
          <Name>
            <Base>Test-completed.wav</Base>
          </Name>
          <Path>/Users/User/DestinationFiles/</Path>
        </File>
        <Segment>
          <SOM>10:00:30:00</SOM>
          <DUR>00:00:10:00</DUR>
          <EOM>10:00:40:00</EOM>
        </Segment>
      </Id>
    </Destination>
    <Report>
      <ReportPath>/Users/User/ReportFiles/</ReportPath>
      <ReportType xml="on" pdf="on" csv="off" />
      <ReportName>
        <Name>
          <Base>Test-report</Base>
        </Name>
      </ReportName>
    </Report>
  </Job>
</Jobs>
```

## Appendix C - Job Post Response XML

Provided the POST has gone through successfully, the number of successfully submitted jobs will be returned within the 'Message' tag and the uuids of those jobs returned in the 'Job\_UUID' tag.

For Example:

```
<POST>
  <Code>OK</Code>
  <Job_UUID>{ec8cd202-98ac-4588-8649-839cbd1bbe3a}</Job_UUID>
  <Message>1 of 1 records evaluated were successfully submitted</Message>
  <Time>Thu Aug 18 11:29:31 2018</Time>
</POST>
```

If any errors have occurred during job submission, they will reported in 'ParseError', 'ParseErrorUUID' and 'InvalidPath' tags.

For Example:

```
<POST>
  <Code>OK</Code>
  <ParseError>Record 0 on line 3: Workflow ID not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 10: {1a3f2db5-4043-40df-a52c-22e4bd650b01} File Path not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 6: Name or Path not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 20: {e40f0657-c345-480d-a7f4-53cf02112f28} File Path not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 16: Name or Path not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 25: Report Path not found or is invalid.</ParseError>
  <ParseError>Record 0 on line 24: ReportName or ReportPath or ReportType not found or is invalid.</ParseError>
  <ParseError>Source Id {1a3f2db5-4043-40df-a52c-22e4bd650b01} not found or is invalid.</ParseError>
  <ParseError>Destination Id {e40f0657-c345-480d-a7f4-53cf02112f28} not found or is invalid.</ParseError>
  <ParseErrorUUID>{ec8cd202-98ac-4588-8649-839cbd1bbe3a}</ParseErrorUUID>
  <InvalidPath>/Volumes/test_clips/Testing/Files/</InvalidPath>
  <InvalidPath>/Volumes/test_clips/Testing/Output/</InvalidPath>
  <InvalidPath>/Volumes/test_clips/Testing/Report/</InvalidPath>
  <Message>0 of 1 records evaluated were successfully submitted</Message>
  <Time>Thu Aug 18 11:29:31 2018</Time>
</POST>
```

## Appendix D - Workflow Required XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <!--Uuid-->
  <xs:simpleType name="Uuid">
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Fa-fo-9]{32}(\{\|\O?[A-Fa-fo-9]{8}-([A-Fa-fo-9]{4}-){3}[A-Fa-fo-9]{12}(\}\|\}))?" />
    </xs:restriction>
  </xs:simpleType>

  <!--OPTIONAL-->
  <xs:simpleType name="optional">
    <xs:restriction base="xs:string">
      <xs:enumeration value="OPTIONAL" />
    </xs:restriction>
  </xs:simpleType>

  <!--SourceFileType-->
  <xs:simpleType name="SourceFileType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:enumeration value="MEDIA" />
      <xs:enumeration value="MXF" />
      <xs:enumeration value="MOV" />
      <xs:enumeration value="WAV" />
      <xs:enumeration value="M2V" />
    </xs:restriction>
  </xs:simpleType>

  <!--DestinationFileType-->
  <xs:simpleType name="DestinationFileType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:enumeration value="SAME_AS_SOURCE" />
      <xs:enumeration value="MXF" />
      <xs:enumeration value="MOV" />
      <xs:enumeration value="WAV" />
      <xs:enumeration value="VARIABLE" />
      <xs:enumeration value="AUDIO_MXF" />
      <xs:enumeration value="AUDIO_MOV" />
      <xs:enumeration value="VIDEO_MXF" />
      <xs:enumeration value="VIDEO_MOV" />
      <xs:enumeration value="M2V" />
    </xs:restriction>
  </xs:simpleType>

  <!--ReportFileType-->
  <xs:simpleType name="ReportFileType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:enumeration value="XML" />
      <xs:enumeration value="PDF" />
      <xs:enumeration value="CSV" />
    </xs:restriction>
  </xs:simpleType>

  <!--StreamStructureType-->
  <xs:simpleType name="StreamStructureType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:enumeration value="MONO_STREAM_STRUCTURE" />
      <xs:enumeration value="STEREO_STREAM_STRUCTURE" />
      <xs:enumeration value="SINGLE_STREAM_STRUCTURE" />
      <xs:enumeration value="SOURCE_STREAM_STRUCTURE" />
      <xs:enumeration value="ADVANCED_STREAM_STRUCTURE" />
    </xs:restriction>
  </xs:simpleType>

  <!--LanguageFormatType-->
  <xs:simpleType name="LanguageFormatType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1" />
      <xs:enumeration value="DMS1_LANGUAGE_FORMAT" />
      <xs:enumeration value="AS11_LANGUAGE_FORMAT" />
      <xs:enumeration value="STANDARD_LANGUAGE_FORMAT" />
      <xs:enumeration value="ISO_LANGUAGE_FORMAT" />
    </xs:restriction>
  </xs:simpleType>

```

```

<!--Pattern for Report-->
<xs:complexType name = "ReportRequired">
  <xs:sequence>
    <xs:element name="FileType" type="ReportFileType" minOccurs="0" maxOccurs="3"/>
  </xs:sequence>
</xs:complexType>

<!--Pattern for Source-->
<xs:complexType name = "SourceRequired">
  <xs:sequence>
    <xs:element name="Id" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType mixed="true">
        <xs:sequence>
          <xs:element name="FileType" type="SourceFileType" minOccurs="0" maxOccurs="1"/>
          <xs:element name="NumChannels" type="xs:positiveInteger" minOccurs="0" maxOccurs="1"/>
          <xs:element name="Channels" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Channel" type="xs:positiveInteger" minOccurs="0" maxOccurs="64"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="ReplaceChannels" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Channel" type="xs:positiveInteger" minOccurs="0" maxOccurs="64"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="ReferenceTimecode" type="optional" minOccurs="0" maxOccurs="1"/>
          <xs:element name="Segment" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="SOM" type="optional" minOccurs="0" maxOccurs="1"/>
                <xs:element name="DUR" type="optional" minOccurs="0" maxOccurs="1"/>
                <xs:element name="EOM" type="optional" minOccurs="0" maxOccurs="1"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
      <xs:attribute name="uuid" type="Uuid" use="required"/>
      <xs:attribute name="description" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:sequence>
</xs:complexType>

<!--Pattern for Destination-->
<xs:complexType name = "DestinationRequired">
  <xs:sequence>
    <xs:element name="Id" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType mixed="true">
        <xs:sequence>
          <xs:element name="FileType" type="DestinationFileType" minOccurs="0" maxOccurs="1"/>
          <xs:element name="NumChannels" type="xs:positiveInteger" minOccurs="0" maxOccurs="1"/>
          <xs:element name="Channels" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Channel" type="xs:positiveInteger" minOccurs="0" maxOccurs="64"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="ReplaceChannels" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Channel" type="xs:positiveInteger" minOccurs="0" maxOccurs="64"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="StreamStructure" type="StreamStructureType" minOccurs="0" maxOccurs="1"/>
          <xs:element name="StreamTagging" minOccurs="0">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Stream" minOccurs="0" maxOccurs="64">
                  <xs:complexType>
                    <xs:attribute name="number" type="xs:string" use="required"/>
                    <xs:attribute name="languageTag" type="xs:string" use="required"/>
                    <xs:attribute name="channels" type="xs:string" use="required"/>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

```

        <xs:element name="LanguageFormat" type="LanguageFormatType" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Timecode" type="optional" minOccurs="0" maxOccurs="1"/>
  <xs:element name="Segment" minOccurs="0">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SOM" type="optional" minOccurs="0" maxOccurs="1"/>
        <xs:element name="DUR" type="optional" minOccurs="0" maxOccurs="1"/>
        <xs:element name="EOM" type="optional" minOccurs="0" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:sequence>
<xs:attribute name="uuid" type="Uuid" use="required"/>
<xs:attribute name="description" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<!--Main-->
<xs:element name="Workflows">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Workflow" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Source" type="SourceRequired" minOccurs="1" maxOccurs="1"/>
            <xs:element name="Destination" type="DestinationRequired" minOccurs="0" maxOccurs="1"/>
            <xs:element name="Report" type="ReportRequired" minOccurs="0" maxOccurs="1"/>
          </xs:sequence>
          <xs:attribute name="uuid" type="Uuid" use="required"/>
          <xs:attribute name="title" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

## Appendix E - Workflow Required Example XML

Example response for **GET workflows/<workflowUuid>/required**

(where **<workflowUuid>** = {9965d6b4-d879-40bc-b270-e3dd80d03e54})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Workflow uuid="{9965d6b4-d879-40bc-b270-e3dd80d03e54}" title="test_wrapping">
  <Source>
    <Id uuid="{98492ebo-dc12-41c2-b819-9bb1b8947171}" description="Video">
      <FileType>M2V</FileType>
    </Id>
    <Id uuid="{0c66dd44-f078-4395-88bc-e626535bbc3f}" description="Audio_Ch_1-2">
      <FileType>WAV</FileType>
      <NumChannels>2</NumChannels>
      <Channels>
        <Channel>1</Channel>
        <Channel>2</Channel>
      </Channels>
    </Id>
    <Id uuid="{6345ec1c-236b-47a8-976a-7d8fba6ee7a1}" description="Audio_Ch_3-4">
      <FileType>WAV</FileType>
      <NumChannels>2</NumChannels>
      <Channels>
        <Channel>3</Channel>
        <Channel>4</Channel>
      </Channels>
    </Id>
  </Source>
  <Destination>
    <Id uuid="{0d28b1e2-2df2-4dce-9fec-f2595338093c}" description="Video+Audio_Ch_1-4">
      <FileType>MXF</FileType>
      <NumChannels>4</NumChannels>
      <Channels>
        <Channel>1</Channel>
        <Channel>2</Channel>
        <Channel>3</Channel>
        <Channel>4</Channel>
      </Channels>
      <StreamStructure>MONO_STREAM_STRUCTURE</StreamStructure>
      <StreamTagging>
        <Stream number="1" languageTag="optional"/>
        <Stream number="2" languageTag="optional"/>
        <Stream number="3" languageTag="optional"/>
        <Stream number="4" languageTag="optional"/>
      </StreamTagging>
      <Timecode>OPTIONAL</Timecode>
    </Id>
  </Destination>
  <Report>
    <FileType>XML</FileType>
    <FileType>PDF</FileType>
    <FileType>CSV</FileType>
  </Report>
</Workflow>
```

In the case where 'required' is called on all workflows (**GET workflows/required**), the response will be as above but encapsulated within a 'Workflows' tag.

## Appendix F - Basic Job Request Example Xml

### Example response for GET jobs

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Jobs>
  <Job uuid="{108b02b7-fc41-4095-8674-24fd28c70162}" watchfolderProfileName="eXAMINE WF" title="DeepThought_WF_17-
Aug-16_10-30-27-953" watchfolderProfileId="{5b9ac218-dd36-4417-8e36-5f4358152338}"/>
  <Job uuid="{d6f216e7-26bc-4a56-8f78-14f8c370dfb7}" watchfolderProfileName="eXAMINE WF" title="DeepThought_WF_17-
Aug-16_10-30-28-090" watchfolderProfileId="{5b9ac218-dd36-4417-8e36-5f4358152338}"/>
  <Job uuid="{18df9f37-7d98-40dd-9525-9fb545490005}" watchfolderProfileName="WF1" title="DeepThought_WF_17-
Aug-16_10-54-42-856" watchfolderProfileId="{3dea4880-14b7-43b1-8810-6297cb7ce6d2}"/>
  <Job uuid="{2a4bbc7a-2784-4852-b3aa-a9705af51272}" watchfolderProfileName="eXAMINE WF" title="DeepThought_WF_17-
Aug-16_11-05-44-341" watchfolderProfileId="{5b9ac218-dd36-4417-8e36-5f4358152338}"/>
  <Job uuid="{8ab93f53-8774-47cc-bab9-784b45baa070}" watchfolderProfileName="WF3" title="DeepThought_WF_17-
Aug-16_20-35-38-599" watchfolderProfileId="{ae73a06c-4cbo-4c63-943f-948a47584767}"/>
  <Job uuid="{494abaac-6b48-405c-ba72-beb9f3e452b3}" watchfolderProfileName="WF3" title="DeepThought_WF_17-
Aug-16_20-37-01-102" watchfolderProfileId="{ae73a06c-4cbo-4c63-943f-948a47584767}"/>
</Jobs>
```

## Appendix G - Status Job Request Example Xml

Example response for **GET jobs/<jobUuid>/status**

(where **<jobUuid>** = {70005df9-496f-4825-99ad-826b0257e766})

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Job
userFlag=""hostName=""state="o"ipAddress=""uuid="{70005df9-496f-4825-99ad-826b0257e766}"stateDescription="SUCCESS"detail
Description="OK"priority="true"workflow="{8317a083-
e9bf-4331-95b9-4a2938b856dd}"watchfolderProfile="{3dea4880-14b7-43b1-8810-6297cb7ce6d2}"title="DeepThought_WF_17-
Aug-16_09-33-32-550"startTime="Wed Aug 17 09:36:54 2016"detectedTime="Wed Aug 17 09:33:32
2016"workflowName="Conditional Loudness"detail="o"progress="100"postTime="Wed Aug 17 09:36:53 2016"finishTime="Wed Aug
17 09:43:36 2016"watchfolderProfileName="WF1"/>
```

In the case where 'status' is called on all jobs (**GET jobs/status**), the response will be as above but encapsulated within a 'Jobs' tag.



## Appendix H - Detail Job Request Example Xml

Example response for **GET jobs/<jobUuid>/detail** and **GET jobs/<jobUuid>**

(where **<jobUuid>** = {70005df9-496f-4825-99ad-826b0257e766})

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<Job startTime="Fri Mar 3 16:37:01 2017" userFlag="eCLIENT_c-p-macbook-pro.home" state="o" title="c-p-macbook-
pro.home_EC_3-Mar-17_16:37:00.997" priority="false" workflow="{aec3154a-5837-4963-ab9c-1b6e96e65b35}"
watchfolderProfileId="{00000000-0000-0000-0000-000000000000}" progress="100" hostName="broadcasthost"
uuid="{8af19e3d-f773-4aa4-83d5-3f17d9e3dad3}" workflowName="loudness_test" watchfolderProfileName=""
stateDescription="SUCCESS" detailDescription="OK" detail="o" postTime="Fri Mar 3 16:37:01 2017" finishTime="Fri Mar 3 16:37:23
2017" detectedTime="" ipAddress="::1">
  <Source>
    <Id uuid="{bc16860f-0681-4bf8-aa66-8db1f59e58de}">
      <File>
        <Name>
          <Base>Test.mxf</Base>
        </Name>
        <Path>/Users/User/SourceFiles/</Path>
      </File>
    </Id>
  </Source>
  <Destination>
    <Id uuid="{d49237c3-564b-4ff7-aoa8-d977e0765855}">
      <File>
        <Name>
          <Base>Test-completed.mxf</Base>
        </Name>
        <Path>/Users/User/DestinationFiles/</Path>
      </File>
    </Id>
    <Id uuid="{07cb1e59-7766-40a8-bb1c-d8a493cecb2f}">
      <File>
        <Name>
          <Base>Test-completed.mxf</Base>
        </Name>
        <Path>/Users/User/DestinationFiles/</Path>
      </File>
    </Id>
  </Destination>
  <Report>
    <ReportPath>/Users/User/DestinationFiles/</ReportPath>
    <ReportType xml="on" pdf="on" csv="on"/>
    <ReportName>
      <Name>
        <Base>Test-report</Base>
      </Name>
    </ReportName>
  </Report>
  <AdvancedStatus>
    <Step stepNumber="1">
      <Loudness>
        <Profile>
          <LoudnessProfileId>{5cdf39be-0729-47c2-9cb8-ab71a53855fe}</LoudnessProfileId>
          <LoudnessProfileName>Example Profile</LoudnessProfileName>
          <LoudnessStatus>ANALYSIS SUCCESSFUL.FILE FAILED.CORRECTION SUCCESSFUL(PROCESSOR USED).</
LoudnessStatus>
        </Profile>
        <Profile>
          <LoudnessProfileId>{766c9b5a-f739-4c42-b192-696ce49a1fd8}</LoudnessProfileId>
          <LoudnessProfileName>Example Profile2</LoudnessProfileName>
          <LoudnessStatus>ANALYSIS SUCCESSFUL.FILE PASSED.</LoudnessStatus>
        </Profile>
      </Loudness>
    </Step>
  </AdvancedStatus>
</Job>
```

In the case where 'detail' is called on all jobs (**GET jobs/detail**), the response will be as above but encapsulated within a 'Jobs' tag.

## Appendix I - Basic Workflow Request Example Xml

### Example response for **GET workflows**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Workflows>
  <Workflow uuid="{84e8cfb1-a384-4589-8c01-7f7dfcb3241f}" title="API_Test"/>
  <Workflow uuid="{271cf415-de93-4292-87c5-ecbafa06130}" title="Everything"/>
  <Workflow uuid="{296e0b4a-1a69-424a-bf4f-d13a01b7c74d}" title="Example Workflow"/>
</Workflows>
```

## Appendix J - Summary Workflow Request Example Xml

Example response for **GET workflows/<workflowUuid>/summary**

(where **<workflowUuid>** = {9965d6b4-d879-40bc-b270-e3dd80d03e54})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <Workflow uuid="{9965d6b4-d879-40bc-b270-e3dd80d03e54}" title="test_wrapping">
    <Summary>
      <Sequence uuid="{d033909a-570c-494d-9198-339095d76397}" title="test_wrapping">
        <SourceChannels>4</SourceChannels>
        <Source>
          <Id uuid="{98492eb0-dc12-41c2-b819-9bb1b8947171}" description="Video">
            <FileType>M2V</FileType>
          </Id>
          <Id uuid="{0c66dd44-f078-4395-88bc-e626535bbc3f}" description="Audio_Ch_1-2">
            <FileType>WAV</FileType>
            <NumChannels>2</NumChannels>
            <Channels>
              <Channel>1</Channel>
              <Channel>2</Channel>
            </Channels>
          </Id>
          <Id uuid="{6345ec1c-236b-47a8-976a-7d8fba6ee7a1}" description="Audio_Ch_3-4">
            <FileType>WAV</FileType>
            <NumChannels>2</NumChannels>
            <Channels>
              <Channel>3</Channel>
              <Channel>4</Channel>
            </Channels>
          </Id>
        </Source>
        <Destination>
          <Id uuid="{0d28b1e2-2df2-4dce-9fec-f2595338093c}" description="Video+Audio_Ch_1-4">
            <FileType>MXF</FileType>
            <NumChannels>4</NumChannels>
            <Channels>
              <Channel>1</Channel>
              <Channel>2</Channel>
              <Channel>3</Channel>
              <Channel>4</Channel>
            </Channels>
            <Metadata>
              <StreamStructure>MONO_STREAM_STRUCTURE</StreamStructure>
              <LanguageFormat>DMS1_LANGUAGE_FORMAT</LanguageFormat>
              <StreamTagging>
                <Stream number="1">
                  <Language>ENG</Language>
                </Stream>
                <Stream number="2">
                  <Language>FRE</Language>
                </Stream>
                <Stream number="3">
                  <Language>JAP</Language>
                </Stream>
                <Stream number="4">
                  <Language>SPA</Language>
                </Stream>
              </StreamTagging>
              <Timecode>
                <StartTimecode>00:00:00:00</StartTimecode>
                <TimecodeFromFile>>false</TimecodeFromFile>
              </Timecode>
            </Metadata>
          </Id>
        </Destination>
      </Sequence>
    </Summary>
  </Workflow>
```

In the case where ‘summary’ is called on all workflows (**GET workflows/summary**), the response will be as above but encapsulated within a ‘Workflows’ tag.

## Appendix K - Detail Workflow Request Example Xml

Example response for **GET workflows/<workflowUuid>/detail** and **GET workflows/<workflowUuid>**

(where **<workflowUuid>** = {9965d6b4-d879-40bc-b270-e3dd80d03e54})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Workflow uuid="{9965d6b4-d879-40bc-b270-e3dd80d03e54}" title="test_wrapping">
  <Summary>
    <Sequence uuid="{d033909a-570c-494d-9198-339095d76397}" title="test_wrapping">
      <SourceChannels>4</SourceChannels>
      <Source>
        <Id uuid="{98492eb0-dc12-41c2-b819-9bb1b8947171}" description="Video">
          <FileType>M2V</FileType>
        </Id>
        <Id uuid="{0c66dd44-f078-4395-88bc-e626535bbc3f}" description="Audio_Ch_1-2">
          <FileType>WAV</FileType>
          <NumChannels>2</NumChannels>
          <Channels>
            <Channel>1</Channel>
            <Channel>2</Channel>
          </Channels>
        </Id>
        <Id uuid="{6345ec1c-236b-47a8-976a-7d8fba6ee7a1}" description="Audio_Ch_3-4">
          <FileType>WAV</FileType>
          <NumChannels>2</NumChannels>
          <Channels>
            <Channel>3</Channel>
            <Channel>4</Channel>
          </Channels>
        </Id>
      </Source>
      <Destination>
        <Id uuid="{0d28b1e2-2df2-4dce-9fec-f2595338093c}" description="Video+Audio_Ch_1-4">
          <FileType>MXF</FileType>
          <NumChannels>4</NumChannels>
          <Channels>
            <Channel>1</Channel>
            <Channel>2</Channel>
            <Channel>3</Channel>
            <Channel>4</Channel>
          </Channels>
          <Metadata>
            <StreamStructure>MONO_STREAM_STRUCTURE</StreamStructure>
            <LanguageFormat>DMS1_LANGUAGE_FORMAT</LanguageFormat>
            <StreamTagging>
              <Stream number="1">
                <Language>ENG</Language>
              </Stream>
              <Stream number="2">
                <Language>FRE</Language>
              </Stream>
              <Stream number="3">
                <Language>JAP</Language>
              </Stream>
              <Stream number="4">
                <Language>SPA</Language>
              </Stream>
            </StreamTagging>
            <Timecode>
              <StartTimecode>00:00:00:00</StartTimecode>
              <TimecodeFromFile>>false</TimecodeFromFile>
            </Timecode>
          </Metadata>
        </Id>
      </Destination>
    </Sequence>
  </Summary>
  <Source>
    <Id uuid="{98492eb0-dc12-41c2-b819-9bb1b8947171}" description="Video">
      <FileType>M2V</FileType>
    </Id>
    <Id uuid="{0c66dd44-f078-4395-88bc-e626535bbc3f}" description="Audio_Ch_1-2">
      <FileType>WAV</FileType>
      <NumChannels>2</NumChannels>
      <Channels>
        <Channel>1</Channel>
      </Channels>
    </Id>
  </Source>
</Workflow>
```

```

    <Channel>2</Channel>
  </Channels>
</Id>
<Id uuid="{6345ec1c-236b-47a8-976a-7d8fba6ee7a1}" description="Audio_Ch_3-4">
  <FileType>WAV</FileType>
  <NumChannels>2</NumChannels>
  <Channels>
    <Channel>3</Channel>
    <Channel>4</Channel>
  </Channels>
</Id>
</Source>
<Destination>
  <Id uuid="{0d28b1e2-2df2-4dce-9fec-f2595338093c}" description="Video+Audio_Ch_1-4">
    <FileType>MXF</FileType>
    <NumChannels>4</NumChannels>
    <Channels>
      <Channel>1</Channel>
      <Channel>2</Channel>
      <Channel>3</Channel>
      <Channel>4</Channel>
    </Channels>
    <StreamStructure>MONO_STREAM_STRUCTURE</StreamStructure>
    <StreamTagging>
      <Stream number="1" languageTag="optional"/>
      <Stream number="2" languageTag="optional"/>
      <Stream number="3" languageTag="optional"/>
      <Stream number="4" languageTag="optional"/>
    </StreamTagging>
    <Timecode>OPTIONAL</Timecode>
  </Id>
</Destination>
<Report>
  <FileType>XML</FileType>
  <FileType>PDF</FileType>
  <FileType>CSV</FileType>
</Report>
</Workflow>

```

In the case where ‘detail’ is called on all workflows (**GET workflows/detail**), the response will be as above but encapsulated within a ‘Workflows’ tag.

## Appendix L - Info Request Example Xml

### Example response for GET info

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<info>
  <Application>Eflow</Application>
  <Version>v.7.0.0</Version>
  <Pid>6256</Pid>
  <ESPsAvailable>4</ESPsAvailable>
  <WatchfoldersActive>10</WatchfoldersActive>
  <WatchfoldersAvailable>32</WatchfoldersAvailable>
  <TimeSinceStart>00:01:54</TimeSinceStart>
  <VolumeRootPath>/</VolumeRootPath>
  <VolumeReadOnly>>false</VolumeReadOnly>
  <VolumeName>Macintosh HD</VolumeName>
  <VolumeFileSystemType>hfs</VolumeFileSystemType>
  <VolumeSizeMB>1999167</VolumeSizeMB>
  <VolumeAvailableSizeMB>642586</VolumeAvailableSizeMB>
  <HostName>My_Mac</HostName>
  <HostAddress>192.168.1.72</HostAddress>
  <OSType>macos</OSType>
  <OSVersion>10.12</OSVersion>
  <CpuArchitecture>x86_64</CpuArchitecture>
</info>
```

## Appendix M - Basic Folder Request Example Xml

### Example response for **GET folders**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Folders>
  <Folder active="false" workflowName="DolbyE" workflow="{c462697c-f554-45a5-b571-3f570458ec6d}" shortcut="o"
  uuid="{cfd1ccd2-154e-4598-979b-bd95444fbf72}" title="DolbyE Folder"/>
  <Folder active="true" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9bod-cded2ea3fdea}" shortcut="o"
  uuid="{of41ceff-c343-41a9-b9bd-9b91b05fd9b8}" title="Loudness Folder"/>
  <Folder active="false" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9bod-cded2ea3fdea}"
  shortcut="o" uuid="{833ad92e-82bd-4243-8b12-b2375b4fb6a5}" title="Two Channel Loudness Folder"/>
  <Folder active="false" workflowName="Pitch" workflow="{c462697c-f554-45a5-b571-3f570458ec6d}" shortcut="o"
  uuid="{ofb423e8-40cb-43b7-a1f8-c31a1fc46660}" title="Pitch Folder"/>
</Folders>
```

If **GET folders/active** is called then only the active folders will be returned

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Folders>
  <Folder active="true" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9bod-cded2ea3fdea}" shortcut="o"
  uuid="{of41ceff-c343-41a9-b9bd-9b91b05fd9b8}" title="Loudness Folder"/>
</Folders>
```

## Appendix N - Status Folder Request Example Xml

Example response for **GET folders/<folderUuid>/status**

(where **<folderUuid>** = {0f41ceff-c343-41a9-b9bd-9b91b05fd9b8})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Folder active="true" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9bod-cded2ea3fdea}" shortcut="0"
uuid="{0f41ceff-c343-41a9-b9bd-9b91b05fd9b8}" title="Loudness Folder"/>
  <Status active="true" valid="true" numFiles="0" stateDescription="RUNNING" state="0" lastSweepTime="10-10-2018_11:49:50"/>
</Folder>
```

In the case where 'status' is called on all folders (**GET folders/status**), the response will be as above but encapsulated within a 'Folders' tag.

If an 'active' tag is used (**GET folders/active/status**), the response will only be as **GET folders/status** but only the active folders will be listed



## Appendix O - Path Folder Request Example Xml

Example response for **GET folders/<folderUuid>/paths**

(where **<folderUuid>** = {0f41ceff-c343-41a9-b9bd-9b91b05fd9b8})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Folder active="true" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9b0d-cded2ea3fdea}" shortcut="o"
uuid="{0f41ceff-c343-41a9-b9bd-9b91b05fd9b8}" title="Loudness Folder"/>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Destination" folderType="Destination">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Error" folderType="Error">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Report" folderType="Report">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Source" folderType="Source">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Source_Move" folderType="Source_Move">valid</Directory>
</Folder>
```

In the case where 'paths' is called on all folders (**GET folders/paths**), the response will be as above but encapsulated within a 'Folders' tag.

If an 'paths' tag is used (**GET folders/active/paths**), the response will only be as **GET folders/paths** but only the active folders will be listed

## Appendix P - Detail Folder Request Example Xml

Example response for **GET folders/<folderUuid>/detail** and **GET folders/<folderUuid>**

(where **<folderUuid>** = {0f41ceff-c343-41a9-b9bd-9b91b05fd9b8})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Folder active="true" workflowName="Conditional Loudness" workflow="{0d44299a-169d-48ab-9bod-cded2ea3fdea}" shortcut="0"
uuid="{0f41ceff-c343-41a9-b9bd-9b91b05fd9b8}" title="Loudness Folder"/>
<Status active="true" valid="true" numFiles="0" stateDescription="RUNNING" state="0" lastSweepTime="10-10-2018_11:49:50"/>
  <CreateCSVReports>2</CreateCSVReports>
  <CreatePDFReports>2</CreatePDFReports>
  <CreateReports>2</CreateReports>
  <CreateXMLReports>2</CreateXMLReports>
  <DestinationDirectory></DestinationDirectory>
  <DestinationRename>0</DestinationRename>
  <DestinationRenameEnd>2</DestinationRenameEnd>
  <DestinationRenameStart>0</DestinationRenameStart>
  <DestinationRenameTimeStamp>0</DestinationRenameTimeStamp>
  <DestinationRenameUDC></DestinationRenameUDC>
  <ErrorMove>2</ErrorMove>
  <ErrorMoveDirectory>/Volumes/Macintosh HD/Users/Test/Error</ErrorMoveDirectory>
  <ErrorRename>0</ErrorRename>
  <ErrorRenameEnd>2</ErrorRenameEnd>
  <ErrorRenameStart>0</ErrorRenameStart>
  <ErrorRenameTimeStamp>0</ErrorRenameTimeStamp>
  <ErrorRenameUDC></ErrorRenameUDC>
  <FolderDestinationMap></FolderDestinationMap>
  <InspectionDelay>2</InspectionDelay>
  <InspectionDelayMins>2</InspectionDelayMins>
  <MultipleDestinationFlag>>false</MultipleDestinationFlag>
  <MultipleSourceFlag>>false</MultipleSourceFlag>
  <ReportAddTimeStamp>0</ReportAddTimeStamp>
  <ReportCustomIdentifier></ReportCustomIdentifier>
  <ReportDirectory>/Volumes/Macintosh HD/Users/Test/Report</ReportDirectory>
  <ReportUseFileID>2</ReportUseFileID>
  <ReportUseFilename>0</ReportUseFilename>
  <ShortcutDelete>0</ShortcutDelete>
  <ShortcutDeleteOnError>0</ShortcutDeleteOnError>
  <ShortcutDirectory></ShortcutDirectory>
  <ShortcutErrorMove>0</ShortcutErrorMove>
  <ShortcutErrorMoveDirectory></ShortcutErrorMoveDirectory>
  <ShortcutErrorRename>0</ShortcutErrorRename>
  <ShortcutErrorRenameEnd>2</ShortcutErrorRenameEnd>
  <ShortcutErrorRenameStart>0</ShortcutErrorRenameStart>
  <ShortcutErrorRenameTimeStamp>0</ShortcutErrorRenameTimeStamp>
  <ShortcutErrorUDC></ShortcutErrorUDC>
  <ShortcutMove>0</ShortcutMove>
  <ShortcutMoveDirectory></ShortcutMoveDirectory>
  <ShortcutMoveDirectoryOnError></ShortcutMoveDirectoryOnError>
  <ShortcutMoveOnError>0</ShortcutMoveOnError>
  <ShortcutRename>0</ShortcutRename>
  <ShortcutRenameEnd>2</ShortcutRenameEnd>
  <ShortcutRenameEndOnError>2</ShortcutRenameEndOnError>
  <ShortcutRenameOnError>0</ShortcutRenameOnError>
  <ShortcutRenameStart>0</ShortcutRenameStart>
  <ShortcutRenameStartOnError>0</ShortcutRenameStartOnError>
  <ShortcutRenameTimeStamp>0</ShortcutRenameTimeStamp>
  <ShortcutRenameTimeStampOnError>0</ShortcutRenameTimeStampOnError>
  <ShortcutUDC></ShortcutUDC>
  <ShortcutUDCOnError></ShortcutUDCOnError>
  <SourceDelete>0</SourceDelete>
  <SourceDirectory>/Volumes/Macintosh HD/Users/Test/Source</SourceDirectory>
  <SourceMove>2</SourceMove>
  <SourceMoveDirectory>/Volumes/Macintosh HD/Users/Test/Source_Move</SourceMoveDirectory>
  <SourceRename>0</SourceRename>
  <SourceRenameEnd>2</SourceRenameEnd>
  <SourceRenameStart>0</SourceRenameStart>
  <SourceRenameTimeStamp>0</SourceRenameTimeStamp>
  <SourceRenameUDC></SourceRenameUDC>
  <WorkflowName>Conditional Loudness</WorkflowName>
  <eXamineWorkflowOption>>false</eXamineWorkflowOption>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Destination" folderType="Destination">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Error" folderType="Error">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Report" folderType="Report">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Source" folderType="Source">valid</Directory>
  <Directory path="/Volumes/Macintosh HD/Users/Test/Source_Move" folderType="Source_Move">valid</Directory>
</Folder>
```

In the case where 'detail' is called on all folders (**GET folders/detail**), the response will be as above but encapsulated within a 'Folders' tag.

If an 'detail' tag is used (**GET folders/active/detail**), the response will only be as **GET folders/detail** but only the active folders will be listed.

## Appendix Q - Recheck Folder Post and Response Example Xml

The following can be **POST** to folders to action a Recheck

```
<Folder>  
  <Action>Recheck</Action>  
</Folder>
```

The response will be

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<POST>  
  <Code>OK</Code>  
  <Message>All watchfolders signalled for recheck</Message>  
  <Time>Thu Aug 18 15:36:39 2016</Time>  
</POST>
```

## Appendix R - Reset Folder Post and Response Example Xml

The following can be **POST** to folders to action a Reset

```
<Folder>  
  <Action>Reset</Action>  
</Folder>
```

The response will be

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<POST>  
  <Code>OK</Code>  
  <Message>All active watchfolders signalled for reset</Message>  
  <Time>Thu Aug 18 15:36:39 2016</Time>  
</POST>
```

## Appendix S - Deleting Jobs and Response Example Xml

### Single Job

To delete a single job

#### DELETE jobs/<jobUuid>

Example response where <jobUuid> = {613410cf-4263-41f8-a931-78e08d86320a}

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DELETE>
  <Code>OK</Code>
  <Message>Database was updated to remove job {613410cf-4263-41f8-a931-78e08d86320a}</Message>
  <Time>Thu Jul 23 11:13:07 2015</Time>
</DELETE>
```

### All Jobs

To delete all jobs

#### DELETE jobs

The response will be

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DELETE>
  <Code>OK</Code>
  <Message>Database was updated to remove all jobs</Message>
  <Time>Thu Jul 23 10:09:49 2015</Time>
</DELETE>
```

### List of Jobs

To delete a list of jobs

#### DELETE jobs/list

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Jobs>
  <Job uuid="{613410cf-4263-41f8-a931-78e08d86320a}"/>
  <Job uuid="{0d44299a-169d-48ab-9bod-cded2ea3fdea}"/>
</Jobs>
```

If successful the uuids of those jobs returned in the 'Job\_UUID' tag.

For Example:

```
<DELETE>
  <Code>OK</Code>
  <Job_UUID>{613410cf-4263-41f8-a931-78e08d86320a}</Job_UUID>
  <Time>Thu Aug 18 11:29:31 2018</Time>
</DELETE>
```

If any errors have occurred during job submission, they will reported in 'ParseError'.

For Example:

```
<DELETE>
  <Code>OK</Code>
  <ParseError>Error deleting multiple jobs from database: Database Error. Please check.</ParseError>
</DELETE>
```

## Appendix T - Changing Job Priority and Response Example Xml

To change the priority of a job

### PUT jobs/<jobUuid>

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Job>
  <Priority>true</Priority>
</Job>
```

Example response where <jobUuid> = {613410cf-4263-41f8-a931-78e08d86320a}

```
<PUT>
  <Code>OK</Code>
  <Message>Priority of job {613410cf-4263-41f8-a931-78e08d86320a} was changed to true</Message>
</PUT>
```

## Appendix U - Error Message Format and Example Xml

The response to an incorrect request will always be in the form:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Error>
  <Code> ... Error Code ... </Code>
  <Message> ... Message describing the error ... </Message>
  <Time> ... Current Date and Time ... </Time>
</Error>
```

For example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Error>
  <Code>Not Found</Code>
  <Message>The requested URL "/jobs/test" was not found on this server. Incorrect Job options supplied. Please review the request.</
Message>
  <Time>Wed Oct 8 16:17:15 2018</Time>
</Error>
```



## Appendix V - Report Job Request Example Xml

Example response for **GET jobs/<jobUuid>/report**

(where **<jobUuid>** = {9ecd24c5-5d37-492e-89ea-3bb13420eca5})

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Job uuid="{9ecd24c5-5d37-492e-89ea-3bb13420eca5}" watchfolderProfileName="" title="job1"
watchfolderProfile="{00000000-0000-0000-0000-000000000000}">
  <EmotionReport>
    <VersionInformation>
      <Version>7.6.0</Version>
    </VersionInformation>
    <DateTime>
      <Date>10-10-2018</Date>
      <Time>15-49-27</Time>
    </DateTime>
    <Workflow>
      <WorkflowProfileId>{3bc3986d-d49b-4518-805e-4dcf05e960bd}</WorkflowProfileId>
      <WorkflowProfileName>8 channel R128 Loudness</WorkflowProfileName>
    </Workflow>
    <Sequence>
      <SequenceProfileId>{1a26ca7f-3852-4e2c-8ee8-360447362b6b}</SequenceProfileId>
      <SequenceProfileName>8 channel R128 Loudness</SequenceProfileName>
    </Sequence>
    <SourceInformation>
      <File>
        <FilePath>/Users/User/Test_Media</FilePath>
        <FileName>8ch_PCM.mov</FileName>
        <FileSize>429072428</FileSize>
        <FileAudioInformation>
          <AudioDuration>32.3</AudioDuration>
          <SampleRate>48000</SampleRate>
          <NumChannels>8</NumChannels>
          <BitsPerSample>16</BitsPerSample>
          <AudioSampleGroups>154800</AudioSampleGroups>
          <NumStreams>7</NumStreams>
          <Compression>twos (bigEndian)</Compression>
          <StreamInformation>
            <Stream>
              <StreamNumber>1</StreamNumber>
              <NumChannelsInStream>1</NumChannelsInStream>
              <Channel>
                <ChannelNumber>1</ChannelNumber>
                <ChannelEncoding>PCM</ChannelEncoding>
              </Channel>
              <StreamLanguage>English</StreamLanguage>
            </Stream>
            <Stream>
              <StreamNumber>2</StreamNumber>
              <NumChannelsInStream>1</NumChannelsInStream>
              <Channel>
                <ChannelNumber>1</ChannelNumber>
                <ChannelEncoding>PCM</ChannelEncoding>
              </Channel>
              <StreamLanguage>English</StreamLanguage>
            </Stream>
            <Stream>
              <StreamNumber>3</StreamNumber>
              <NumChannelsInStream>1</NumChannelsInStream>
              <Channel>
                <ChannelNumber>1</ChannelNumber>
                <ChannelEncoding>PCM</ChannelEncoding>
              </Channel>
              <StreamLanguage>English</StreamLanguage>
            </Stream>
            <Stream>
              <StreamNumber>4</StreamNumber>
              <NumChannelsInStream>1</NumChannelsInStream>
              <Channel>
                <ChannelNumber>1</ChannelNumber>
                <ChannelEncoding>PCM</ChannelEncoding>
              </Channel>
              <StreamLanguage>English</StreamLanguage>
            </Stream>
            <Stream>
              <StreamNumber>5</StreamNumber>
              <NumChannelsInStream>1</NumChannelsInStream>
            </Stream>
          </StreamInformation>
        </FileAudioInformation>
      </File>
    </SourceInformation>
  </EmotionReport>
</Job>
```

```

    <Channel>
      <ChannelNumber>1</ChannelNumber>
      <ChannelEncoding>PCM</ChannelEncoding>
    </Channel>
  </Stream>
</Stream>
<Stream>
  <StreamNumber>6</StreamNumber>
  <NumChannelsInStream>1</NumChannelsInStream>
  <Channel>
    <ChannelNumber>1</ChannelNumber>
    <ChannelEncoding>PCM</ChannelEncoding>
  </Channel>
  <StreamLanguage>English</StreamLanguage>
</Stream>
</Stream>
<Stream>
  <StreamNumber>7</StreamNumber>
  <NumChannelsInStream>2</NumChannelsInStream>
  <Channel>
    <ChannelNumber>1</ChannelNumber>
    <ChannelEncoding>PCM</ChannelEncoding>
  </Channel>
  <Channel>
    <ChannelNumber>2</ChannelNumber>
    <ChannelEncoding>PCM</ChannelEncoding>
  </Channel>
  <StreamLanguage>English</StreamLanguage>
</Stream>
</StreamInformation>
<FileTimecodeInformation>
  <FrameRate>29.97</FrameRate>
  <DropFrame>>false</DropFrame>
  <StartTimecode>00:02:23:06</StartTimecode>
</FileTimecodeInformation>
</FileAudioInformation>
<ChannelInformation>
  <SourceChannels>
    <SourceChannel>1</SourceChannel>
    <SourceChannel>2</SourceChannel>
    <SourceChannel>3</SourceChannel>
    <SourceChannel>4</SourceChannel>
    <SourceChannel>5</SourceChannel>
    <SourceChannel>6</SourceChannel>
    <SourceChannel>7</SourceChannel>
    <SourceChannel>8</SourceChannel>
  </SourceChannels>
</ChannelInformation>
</File>
</SourceInformation>
<Step>
  <StepInformation>
    <StepNumber>1</StepNumber>
    <StepDescription>LOUDNESS</StepDescription>
    <StepStatus>OK</StepStatus>
  </StepInformation>
<StepSummary>
  <Profile>
    <LoudnessProfileId>{817c2d37-892e-4465-a857-07a0ef392e22}</LoudnessProfileId>
    <LoudnessProfileName>R128</LoudnessProfileName>
    <LoudnessDetailInformation>
      <LoudnessStatusInformation>
        <LoudnessStatus>ANALYSIS SUCCESSFUL.FILE FAILED.CORRECTION SUCCESSFUL.</LoudnessStatus>
      </LoudnessStatusInformation>
      <ChannelGroupInformation>
        <NumGroups>2</NumGroups>
        <ChannelGroupLayout>Stereo_FivePointOne</ChannelGroupLayout>
      </ChannelGroupInformation>
      <ChannelGroup>
        <ChannelGroupName>Stereo</ChannelGroupName>
        <AnalysisInformation>
          <Loudness>
            <LoudnessSpecification>EBU R128</LoudnessSpecification>
            <LoudnessUnits>LUFS</LoudnessUnits>
            <ProgramLoudnessThreshold>-23.0</ProgramLoudnessThreshold>
            <ProgramLoudness>-22.0</ProgramLoudness>
          </Loudness>
          <TruePeak>
            <TruePeakThreshold>-3.00</TruePeakThreshold>
            <MaximumTruePeak>-4.99</MaximumTruePeak>
            <GreatestTruePeakViolation>0.00</GreatestTruePeakViolation>
          </TruePeak>
        </AnalysisInformation>
      </ChannelGroup>
    </LoudnessDetailInformation>
  </Profile>
</StepSummary>

```

```

    <Attenuation>
      <dBAttenuationRequired>-1.0</dBAttenuationRequired>
    </Attenuation>
  </AnalysisInformation>
  <CorrectionInformation>
    <Correct>ProgramLoudness</Correct>
    <Correct>TruePeak</Correct>
  </CorrectionInformation>
  <ToneDetection>
    <ToneDetected>False</ToneDetected>
    <Tone>No Tone Detected</Tone>
  </ToneDetection>
</ChannelGroup>
<ChannelGroup>
  <ChannelGroupName>FivePointOne</ChannelGroupName>
  <AnalysisInformation>
    <Loudness>
      <LoudnessSpecification>EBU R128</LoudnessSpecification>
      <LoudnessUnits>LUFS</LoudnessUnits>
      <ProgramLoudnessThreshold>-23.0</ProgramLoudnessThreshold>
      <ProgramLoudness>-22.0</ProgramLoudness>
    </Loudness>
    <TruePeak>
      <TruePeakThreshold>-3.00</TruePeakThreshold>
      <MaximumTruePeak>-4.99</MaximumTruePeak>
      <GreatestTruePeakViolation>0.00</GreatestTruePeakViolation>
    </TruePeak>
    <Attenuation>
      <dBAttenuationRequired>-1.0</dBAttenuationRequired>
    </Attenuation>
  </AnalysisInformation>
  <CorrectionInformation>
    <Correct>ProgramLoudness</Correct>
    <Correct>TruePeak</Correct>
  </CorrectionInformation>
  <ToneDetection>
    <ToneDetected>False</ToneDetected>
    <Tone>No Tone Detected</Tone>
  </ToneDetection>
</ChannelGroup>
</LoudnessDetailInformation>
</Profile>
</StepSummary>
</Step>
<DestinationInformation>
  <File>
    <FilePath>/Users/User/Destination</FilePath>
    <FileName>8ch_PCM_Media_Ch_1-8.mov</FileName>
    <FileSize>429072428</FileSize>
    <ChannelInformation>
      <DestinationChannels>
        <DestinationChannel>1</DestinationChannel>
        <DestinationChannel>2</DestinationChannel>
        <DestinationChannel>3</DestinationChannel>
        <DestinationChannel>4</DestinationChannel>
        <DestinationChannel>5</DestinationChannel>
        <DestinationChannel>6</DestinationChannel>
        <DestinationChannel>7</DestinationChannel>
        <DestinationChannel>8</DestinationChannel>
      </DestinationChannels>
    </ChannelInformation>
  </File>
</DestinationInformation>
<TimingInformation>
  <WorkflowTime>8.1</WorkflowTime>
  <WorkflowSpeed>4</WorkflowSpeed>
</TimingInformation>
</EmotionReport>
</Job>

```