
AME Scripting Guide

Release 1.0

Jul 30, 2020

1	Introduction	1
2	Application Object	3
2.1	Attributes	3
2.2	Methods	3
3	Front End Object	7
3.1	Methods	7
4	Exporter Object	11
4.1	Attributes	11
4.2	Methods	12
5	Encoder Host Object	13
5.1	Methods	13
6	Batch Item Object	17
6.1	Attributes	17
6.2	Methods	18

CHAPTER 1

Introduction

This is a community-created and -maintained project documenting the Extendscript scripting API for Adobe Media Encoder.

The Application Object is the master object for controlling the application.

app

2.1 Attributes

2.1.1 app.launchTime

app.launchTime

The time in seconds AME took to launch.

Type

Number; read-only.

2.2 Methods

2.2.1 app.getFrontend()

More on the Front End Object

app.getFrontend()

Parameters

None.

Returns

Front End Object.

2.2.2 app.getExporter()

More on the Exporter Object

```
app.getExporter()
```

Parameters

None.

Returns

Exporter Object.

2.2.3 app.getEncoderHost()

More on the Encoder Host Object

```
app.getEncoderHost()
```

Parameters

None.

Returns

Encoder Host Object.

2.2.4 app.quit()

Quits the application.

```
app.quit()
```

Parameters

None.

Returns

None.

2.2.5 app.getBuildNumber()

Gets the application build number.

```
app.getBuildNumber()
```

Parameters

None.

Returns

String. Build number.

2.2.6 app.scheduleTask()

Schedule a task for a later time. (similar to `setTimeout()` in modern JS)

```
app.scheduleTask()
```

Parameters

<code>stringToExecute</code>	A string containing JavaScript to be executed.
<code>delay</code>	A number of milliseconds to wait before executing the JavaScript. A floating-point value.
<code>repeat</code>	When true, execute the script repeatedly, with the specified delay between each execution. When false the script is executed only once.

Returns

Integer, a unique identifier for this task, which can be used to cancel it with `app.cancelTask()`.

2.2.7 app.cancelTask()

Removes the specified task from the queue of tasks scheduled for delayed execution.

```
app.cancelTask(taskID)
```

Parameters

<code>taskID</code>	An integer that identifies the task, as returned by <code>app.scheduleTask()</code> .
---------------------	---

Returns

Nothing.

2.2.8 app.getWatchFolder()*

```
app.getWatchFolder() *untested
```

Parameters

unknown

Returns

unknown

2.2.9 app.wait()*

```
app.wait() *untested
```

Parameters

unknown

Returns

unknown

2.2.10 app.assertToConsole()*

`app.assertToConsole()` **untested*

Parameters

unknown

Returns

unknown

2.2.11 app.renderFrameSequence()*

`app.renderFrameSequence()` **untested*

Parameters

unknown

Returns

unknown

2.2.12 app.isBlackVideo()*

`app.isBlackVideo()` **untested*

Parameters

unknown

Returns

unknown

2.2.13 app.isSilentAudio()*

`app.isSilentAudio()` **untested*

Parameters

unknown

Returns

unknown

2.2.14 app.simulateUIEvents()*

`app.simulateUIEvents()` **untested*

Parameters

unknown

Returns

unknown

The Front End Object allows you to add items to the render queue (aka batch).

```
app.getFrontend()
```

3.1 Methods

3.1.1 FrontEnd.addDLToBatch()

Add Comp from After Effects or Sequences from Premiere Pro.

```
app.getFrontend().addDLToBatch(project, format, preset, guid, destination)
```

Parameters

project	Path to the project file. (e.g. .aep, .prpro)
format	The name of the format used in the preset EPR. (e.g. H.264, Quicktime, etc.)
preset	Path to the EPR preset file.
guid	The Dynamic Link GUID from the After Effects Comp or Premiere Pro Sequence
destination	Path to the destination file.

Returns

Batch Item Object

Note: Application will hang until process is complete or will throw an exception if arguments are incorrect. This method ignores work area. This method resets timecode to 00:00:00.

3.1.2 FrontEnd.addCompToBatch()

Add Comp from After Effects.

```
app.getFrontend().addCompToBatch(project, preset, destination)
```

Parameters

project	Path to the project file. (e.g. .aep, .prpro)
preset	Path to the EPR preset file.
destination	Path to the destination file.

Returns

Batch Item Object

Note: Method requires project to be structured so that 1 and only 1 comp is at the root of the project. Application will throw an exception if no comp is present in the root or more than 1 comps are present.

3.1.3 FrontEnd.addItemToBatch()

Add any footage item into the render queue.

```
app.getFrontend().addItemToBatch(file)
```

Parameters

file	Path to the media file.
------	-------------------------

Returns

Batch Item Object

3.1.4 FrontEnd.addFileToBatch()

Add any footage item into the render queue.

```
app.getFrontend().addFileToBatch(file, format, preset, destination)
```

Parameters

file	Path to the media file.
format	The name of the format used in the preset EPR. (e.g. H.264, Quicktime, etc.)
preset	Path to the EPR preset file.
destination	Path to the destination file.

Returns

Batch Item Object

3.1.5 FrontEnd.addFileSequenceToBatch()*

Add any footage item into the render queue.

```
app.getFrontend().addFileSequenceToBatch() *untested
```

Parameters

unknown

Returns

Batch Item Object

3.1.6 FrontEnd.addXMLToBatch()*

Add an XML project to the batch.

```
app.getFrontend().addXMLToBatch() *untested
```

Parameters

unknown

Returns

Batch Item Object

3.1.7 FrontEnd.addTeamProjectsItemToBatch()*

Add team project items to the render queue.

```
app.getFrontend().addTeamProjectsItemToBatch() *untested
```

Parameters

unknown

Returns

Batch Item Object

3.1.8 FrontEnd.stitchFiles()*

Stitch files for rendering.

```
app.getFrontend().stitchFiles() *untested
```

Parameters

unknown

Returns

unknown

3.1.9 FrontEnd.stopBatch()*

Stops batch from running.

```
app.getFrontend().stopBatch()
```

Parameters

None.

Returns

Result Boolean.

3.1.10 FrontEnd.getDLItemsAtRoot(projectPath)

Returns a list of renderable sequences contained at the root of the specified .prproj.

```
var listOfRenderableSequences = app.getFrontend().getDLItemsAtRoot('/Users/  
example/Desktop/example.prproj');
```

Parameters

Full path to the .prproj file to be examined.

Returns

An array of GUIDs of renderable sequences; Array will be empty if there are none.

If you set format to empty (var format = ""), the format will instead be dictated by the preset. If a preset is not provided, then AME will use the most-recently-used preset associated with the given format. If both are provided, and they disagree, an error will occur.

The Exporter Object allows you to access and modify the render queue.

```
app.getExporter()
```

4.1 Attributes

4.1.1 Exporter.encodeID

```
app.getExporter().encodeID
```

Gets the current Encode ID.

Type

String; read-only.

4.1.2 Exporter.result

```
app.getExporter().result
```

unknown

Type

Boolean; read-only.

4.1.3 Exporter.elapsedMilliseconds

```
app.getExporter().elapsedMilliseconds
```

Elapsed milliseconds since queue started.

Type

Number; read-only.

4.2 Methods

4.2.1 `Exporter.removeAllBatchItems()`

Removes all items from the render queue.

```
app.getExporter().removeAllBatchItems()
```

Parameters

None.

Returns

Result Boolean.

Note: Application will hang until process is complete.

4.2.2 `Exporter.trimExportForSR()*`

```
app.getExporter().trimExportForSR() *untested
```

Parameters

unknown

Returns

unknown

4.2.3 `Exporter.ExportItem()*`

```
app.getExporter().exportItem() *untested
```

Parameters

unknown

Returns

unknown

4.2.4 `Exporter.ExportGroup()*`

```
app.getExporter().exportGroup() *untested
```

Parameters

unknown

Returns

unknown

The Encoder Host Object allows you to control and query the render queue.

```
app.getEncoderHost ()
```

5.1 Methods

5.1.1 EncoderHost.runBatch()

Starts the render queue.

```
app.getEncoderHost ().runBatch ()
```

Parameters

None.

Returns

Result Boolean.

5.1.2 EncoderHost.pauseBatch()

Pauses the render queue.

```
app.getEncoderHost ().pauseBatch ()
```

Parameters

None.

Returns

Result Boolean.

5.1.3 EncoderHost.stopBatch()

Stops the render queue.

```
app.getEncoderHost().stopBatch()
```

Parameters

None.

Returns

Result Boolean.

5.1.4 EncoderHost.isBatchRunning()

Checks if the render queue is rendering.

```
app.getEncoderHost().isBatchRunning()
```

Parameters

None.

Returns

Result Boolean.

5.1.5 EncoderHost.getFormatList()

Checks if the render queue is rendering.

```
app.getEncoderHost().getFormatList()
```

Parameters

None.

Returns

Array of Format Names as Strings.

5.1.6 EncoderHost.createEncoderForFormat()

Creates an Encoder for a given Format

```
app.getEncoderHost().createEncoderForFormat(format) *untested
```

Parameters

format: String. format name (H.264, Quicktime, etc.)

Returns

Encoder Object. (encode from there with `encoder.encode(src, dst)`)

5.1.7 EncoderHost.createMediaComparator()*

`app.getEncoderHost().createMediaComparator()` **untested*

Parameters

unknown

Returns

unknown

5.1.8 EncoderHost.getSupportedImportFileTypes()*

`app.getEncoderHost().getSupportedImportFileTypes()` **untested*

Parameters

unknown

Returns

unknown

5.1.9 EncoderHost.getSourceInfo()*

`app.getEncoderHost().getSourceInfo()` **untested*

Parameters

unknown

Returns

unknown

5.1.10 EncoderHost.getCurrentBatchPreview()*

`app.getEncoderHost().getCurrentBatchPreview()` **untested*

Parameters

unknown

Returns

unknown

Batch Item Object

The Batch Item Object is returned after adding an item to the render queue, and then allows you to make further modifications and track the status of the render job.

More on adding items to the queue with Front End Object

6.1 Attributes

6.1.1 BatchItem.outputFiles

`batchItem.outputFiles`

Contains the output files from the render job once the job is complete.

Type

Array; read-only.

6.1.2 BatchItem.encodeProgress

`batchItem.encodeProgress`

The encode progress from 0 - 100 of render job.

Type

Number; read-only.

6.1.3 BatchItem.outputWidth*

`batchItem.outputWidth` **untested*

unknown

Type

Number;

6.1.4 BatchItem.outputHeight*

`batchItem.outputHeight` **untested*

unknown

Type

Number;

6.2 Methods

6.2.1 BatchItem.getEncodeProgress()

Gets the progress from 0 - 100 of the render job.

`batchItem.getEncodeProgress()`

Parameters

None.

Returns

Number.

6.2.2 BatchItem.getEncodeTime()*

`batchItem.getEncodeTime()` **untested*

Parameters

None.

Returns

Number.

6.2.3 BatchItem.getCurrentBatchPreview()*

`batchItem.getCurrentBatchPreview()` **untested*

Parameters

unknown

Returns

unknown

6.2.4 BatchItem.getPresetList()*

`batchItem.getPresetList()` **untested*

Parameters

unknown

Returns

unknown

6.2.5 BatchItem.getMissingAssets()*

`batchItem.getMissingAssets()` **untested*

Parameters

unknown

Returns

unknown

6.2.6 BatchItem.loadPreset()*

`batchItem.loadPreset()` **untested*

Parameters

unknown

Returns

unknown

6.2.7 BatchItem.loadFormat()*

`batchItem.loadFormat()` **untested*

Parameters

unknown

Returns

unknown

6.2.8 BatchItem.setWorkArea()*

`batchItem.setWorkArea()` **untested*

Parameters

unknown

Returns

unknown

6.2.9 BatchItem.setUsePreviewFiles()*

`batchItem.setUsePreviewFiles()` **untested*

Parameters

unknown

Returns

unknown

6.2.10 BatchItem.setUseMaximumRenderQuality()*

`batchItem.setUseMaximumRenderQuality()` **untested*

Parameters

unknown

Returns

unknown

6.2.11 BatchItem.setUseFrameBlending()*

`batchItem.setUseFrameBlending()` **untested*

Parameters

unknown

Returns

unknown

6.2.12 BatchItem.setIncludeSourceXMP()*

`batchItem.setIncludeSourceXMP()` **untested*

Parameters

unknown

Returns

unknown

6.2.13 BatchItem.setIncludeSourceCuePoints()*

`batchItem.setIncludeSourceCuePoints()` **untested*

Parameters

unknown

Returns

unknown

6.2.14 BatchItem.setCropState()*

`batchItem.setCropState()` **untested*

Parameters

unknown

Returns

unknown

6.2.15 BatchItem.setCropType()*

`batchItem.setCropType()` **untested*

Parameters

unknown

Returns

unknown

6.2.16 BatchItem.setCropOffsets()*

`batchItem.setCropOffsets()` **untested*

Parameters

unknown

Returns

unknown

6.2.17 BatchItem.setOutputFrameSize()*

`batchItem.setOutputFrameSize()` **untested*

Parameters

unknown

Returns

unknown

6.2.18 BatchItem.setXMPData()*

`batchItem.setXMPData()` **untested*

Parameters

unknown

Returns

unknown

6.2.19 BatchItem.setCuePointData()*

`batchItem.setCuePointData()` **untested*

Parameters

unknown

Returns

unknown

6.2.20 BatchItem.setScaleType()*

`batchItem.setScaleType()` **untested*

Parameters

unknown

Returns

unknown