

# **pt\_OpenSesame Guide**

# Table of Contents

|  |    |
|--|----|
| pt_OpenSesame.....   | 4  |
| Creating backwards compatible projects.....                  | 4  |
| Troubleshooting.....   | 4  |
| Trial limitations.....                                       | 4  |
| Backwards compatible project limitations.....                | 4  |
| Exporting text layers for spreadsheet editing.....           | 5  |
| Exporting and editing project values.....                    | 6  |
| Using the Edit List to select the editable values.....       | 6  |
| Edit List export options.....                                | 7  |
| Setting up the Render Queue.....                             | 7  |
| Exporting a project template.....                            | 7  |
| Export Project.....  | 8  |
| Import Project.....  | 8  |
| Preferences.....   | 8  |
| pt_OpenSesame Server.....                                    | 10 |
| Creating project templates.....                              | 10 |
| Running watch folder mode.....                               | 10 |
| Scanning the watch folder.....                               | 10 |
| Text control file (.tmp).....                                | 11 |
| Processing and rendering.....                                | 11 |
| Email Alerts.....  | 11 |
| Logs.....  | 11 |
| Using OpenSesame Server with a render farm.....              | 12 |
| Preferences.....   | 12 |
| Editing OpenSesame text files.....                           | 14 |
| Deleting settings you don't want to edit.....                | 14 |
| Common settings.....   | 14 |
| Header section.....  | 15 |
| Main section.....  | 16 |
| Footage item.....  | 16 |
| Comp item.....   | 16 |
| Layer item.....  | 17 |
| Text item.....   | 17 |
| Values item.....   | 18 |
| Scripts section.....   | 21 |
| Render section.....  | 21 |
| Render Settings.....   | 21 |
| Output Module settings.....                                  | 21 |
| Render / Save options.....                                   | 22 |
| Additional topics.....                                       | 23 |
| Text Encoding.....   | 23 |
| Special Characters.....                                      | 23 |
| CSV/TSV Export.....  | 23 |
| Excel Safe Export.....                                       | 24 |
| JSON Export.....   | 24 |
| Information about file paths.....                            | 24 |
| Text formatting controls.....                                | 25 |
| Automatically resize footage elements using expressions..... | 26 |
| Helper scripts.....  | 27 |
| Existing helper scripts.....                                 | 27 |
| Getting more helper scripts.....                             | 28 |
| Creating helper scripts.....                                 | 28 |



## pt\_OpenSesame

OpenSesame opens up the After Effects project format, letting you export parts or entire projects to easily editable text files. Create backwards compatible projects that can be opened in any version back to CS3. Or export specific parts for editing in a spreadsheet or text editor to quickly change things like text, text styles, footage paths, property values and render settings.

OpenSesame can also be used in conjunction with OpenSesame Server, a separate script that enables fully automated project versioning and rendering workflows. [aescrpts.com/pt\\_opensesame-server](https://aescrpts.com/pt_opensesame-server)

## Creating backwards compatible projects

### Step 1

Open the project you want to convert, run OpenSesame and click Export Project. A log file will be created listing any items that couldn't be exported (see 'Backwards compatible project limitations' below).

NOTE: If you don't have the version of After Effects required to open the project you will need someone that does to do this first step. Possibly you could ask the person you received the project from. They can use the trial version of OpenSesame for this as there are no trial limitations when exporting.

### Step 2

Launch your earlier version of After Effects (CS3 or later) and run OpenSesame. Click Import Project, then choose the previously exported text file.

### Tips

Where possible OpenSesame stores absolute and relative file paths for footage used in a project (relative to the exported project location). If importing on a different computer than was used for the export, footage should still be relinked if it's in the same location relative to the project file.

If the text styles weren't imported correctly and you have both versions of AE on the same computer, you can copy/paste the text styles (using the Text tool to select the text) from the original to the imported project.

## Troubleshooting

If the import stalls before completing there may be a problem when trying to import certain types of footage. Click [?] to access the preferences and check the 'Don't import footage' option, then try importing again.

If that doesn't help, locate the text file containing 'import failed on' in the name, open a support ticket at [aescrpts.com/contact](https://aescrpts.com/contact) and send a copy of that file along with the main exported text file.

## Trial limitations

The trial version of OpenSesame is limited so that effects aren't added to the imported projects. Remove this limitation by purchasing a license at [aescrpts.com/pt\\_opensesame](https://aescrpts.com/pt_opensesame)

## Backwards compatible project limitations

Some parts of a project can't be exported due to scripting limitations. The following features aren't currently supported:

- Paint, Rotobrush and Puppet Pin effect values.
- Text style support is CS4+ only and doesn't include features like kerning, vertical and paragraph text, or multiple text styles on one layer.
- Proxies.
- Custom data values. Effect properties with custom data values can't be exported, such as the Levels Histogram and the Hue/Saturation effect. In these cases you could use the Levels (Individual Controls) and Color Balance (HLS) effects instead.

## **Exporting text layers for spreadsheet editing**

You can quickly and easily export the contents of text layers in a project for editing in a text editor or spreadsheet. This can be very useful if you need to send the text to a client for approval, or to produce multiple versions of a project for different languages.

### **Step 1**

Run OpenSesame, check the 'Export Editable Values' option and click 'Show Edit List'. When exporting parts of a project for external editing, you use the Edit List to select the parts you want to edit.

### **Step 2**

To export all the text in the project for editing, simply click the 'All Text' button. To export some of the text either click 'All Text' first, then select the unwanted ones and use 'Remove Items' to remove them from the list, or go through the project selecting the specific layers and use the add 'Selected Text' button.

NOTE: Items that have been added to the Edit List are flagged in the layer comments. You can restore them to the Edit List later by simply clicking the 'Load List' button.

### **Step 3**

Use the options below the Edit List to choose to export just the text contents or also the text styles (Font, Font Size and Fill Color). Once that's done you can close the Edit List window.

### **Step 4**

Click OpenSesame's [ ? ] button to access the preferences and choose the format you want from the 'Export editable as' dropdown menu. You can export to either tab separated (.TSV) or comma separated (.CSV) text file formats, both of which can be opened directly into a spreadsheet like Excel for editing.

### **Step 5**

Click the 'Export Project' button. You'll be asked to resave your project file and a text values file will be created along with it, with the same name and "VALUES" at the end.

### **Step 6**

Open the text VALUES file in either a text editor or spreadsheet app like Excel and edit the text as required. Once completed, save the file in the same format. Save as many versions as you like, the naming of the text file is not important.

### **Step 7**

Run OpenSesame and click the Import Project button. Select the text file containing the edited text. OpenSesame will open the original After Effects project and make the changes specified in the text file.

## Exporting and editing project values

The later '[Editing OpenSesame text files](#)' section provides deeper instructions on the variety of ways you can edit project values in a text editor, spreadsheet app or using some automated method for submissions to OpenSesame Server.

It is recommended that you use the 'Export Editable Values' option to edit values externally, as the original project will be preserved without any of the limitations that exist when exporting and reimporting the entire project, and it will import a lot faster.

Having said that, you have access to far more settings (mask vertices, keyframe interpolation, etc) when editing an entire project that has been exported to a text file, so there may be cases where you'll want to edit it in this way. You'll have to balance this with the limitation that certain settings can't be preserved.

These instructions describe how to export and edit specific parts of a project, but much of the information is relevant to the settings found in an entire project export too.

## Using the Edit List to select the editable values

Open the After Effects project you want to create a template from, run the OpenSesame script and check "Export Editable Values", then click "Show Edit List" to open the Edit List window.

The Edit List window shows all the elements in your project that you've chosen to make editable. You can add footage, compositions, layers, text and most other property values.

### Add Selected Items

Select footage items or compositions in the Project panel, or layers or properties in a composition and click "Add Selected Items" to add them to the Edit List.

### Add Selected Text

Select the text layers you want to make editable and click "Add Selected Text" to add them to the Edit list. This is the same as selecting a text layer's Source Text property and using "Add Selected Items" but is a bit easier.

### Add All Text

This option scans through the project and adds all text layers to the Edit List.

### Add All Footage

This option scans through the project and adds all footage items to the Edit List.

### Add Script

Use this to add OpenSesame compatible helper scripts to your export. See the '[Helper scripts](#)' section and '[Editing OpenSesame text files > Scripts section](#)' for more information.

Before adding any helper scripts, I'd recommend adding any other project items first and sorting the list by type. Helper scripts will be listed in the text file and applied in the order they are added (although you can always edit this order in the text file manually). Helper scripts aren't saved with the template project, so if you return to it later and use 'Load List' to restore the edit list, you'll have to add the scripts again.

### Remove Items

Select items in the Edit List and use this button to remove them.

### Load List

When an item is added to the Edit List a flag is added to the comments in the project. If you reopen a

template project at a later date, you can restore the edit list by simply clicking this button. OpenSesame will scan through the project looking for those comments and add any flagged items to the edit list.

### **Sort**

Items appear in the edit list in the order that they're added. Use the Sort pulldown to sort them by type or group them by composition. Use the Refresh option if you've made any changes to the project and want to see those changes reflected in the edit list.

### **Edit List export options**

The controls at the bottom of the Edit List window let you configure which settings will be exported for each of the item types.

#### **Text**

You can either choose to only export the actual text contents of text layers, or you can also export the text styles, which will include Font, Font Size and Fill Color.

#### **Layers**

For layers you can export just the on/off switches for video (and audio if applicable), the layer times (startTime, inPoint and outPoint) or both.

#### **Properties**

For properties you can export just the property values, or you can also export any expressions that might be applied to them.

#### **Footage**

For footage you can choose to just export the absolute footage paths or you can also export relative paths. Relative paths can only be generated if the footage is on the same volume as the exported project, and the paths are relative to that location (stored in the [Header section](#)'s FootageBase setting).

### **Setting up the Render Queue**

You can't add Render Queue items to the Edit List. Instead, OpenSesame will automatically export the information for any existing queued render items. When exporting for use with OpenSesame Server you should ensure you've set up at least one queued render item before exporting. You'll get a warning on export if this hasn't been done.

You can edit render queue items in the text file, duplicating them, changing the render times, render settings, output module templates (even adding additional output modules) and file paths.

### **Exporting a project template**

Click [ ? ] to open the preferences and choose the format you want to use for the text values file. Select one of the options from the "Export editable as" dropdown menu. The options are:

#### **Tab separated values (.tsv)**

Each entry is saved to a separate line of a text file, with each of the settings separated by tabs. This file can be edited in a text editor or in a spreadsheet where each setting will be shown in a different cell.

#### **Comma separated values (.csv)**

Each entry is saved to a separate line of a text file, with each of the settings separated by commas. This file

can be edited in a text editor or in a spreadsheet where each setting will be shown in a different cell.

### **JavaScript Object Notation (.json)**

This text file is saved using the JSON format. While it can be edited in a text editor, it is primarily intended to be read by some system you might design that can process, revise and export copies of the file to create a completely automated workflow from end to end.

### **Export Project**

Once you've added the items you want editable to the Edit List, click "Export Project". You'll be asked to resave your project file and a text values file will be created along with it, with the same name and "VALUES" at the end.

If you are using relative paths to specify footage then ideally you should export to your Watch Folder location. The FootageBase path at the beginning of the text file will be the path you export to and any relative footage paths will be relative to that location.

### **Import Project**

For OpenSesame, once you've edited the text file externally, use the "Import Project" button and select the edited text file. OpenSesame will open the original After Effects project and make any changes specified in the text file.

You can also use this for testing in advance of deploying with [pt\\_OpenSesame Server](#).

## **Preferences**

### **Export project as**

Set the format used when exporting an entire project (i.e. when 'Export editable values' is disabled). Choose either the tab (.TSV) or comma (.CSV) separated file formats. It doesn't matter which option you choose unless you intend to edit the project externally and find it easier to work with one format.

### **Export editable as**

Set the format for the exported editable values text file. Choose either tab separated (.TSV), comma separated (.CSV) or JavaScript Object Notation (.JSON) format text files. See 'Exporting a project template' on page 6 of this guide for a description of the different formats.

### **Preserve commas / tabs**

Tabs and commas that are used in text, comp names, layer names or footage will be encoded so they don't cause problems in the tab or comma separated text file (see the Special Characters section). When this 'Preserve' option is enabled, only the character that is being used as the separator will be encoded.

### **Excel Safe export**

Use this option to additionally encode quotes and apostrophes. This may be necessary to preserve those characters when editing in Excel (see the Special Characters section for more information).



## **Text encoding**

Set the text encoding format used to read/write text values files. See '[Additional topics > Text Encoding](#)' for more information.

## **Favour absolute/relative paths**

OpenSesame text files contain both absolute and relative paths for any footage. Use this preference to select which kind OpenSesame Server will use first when attempting to locate the footage. See '[Additional topics > Information about file paths](#)' for more information.

## **Don't import footage**

If the import stalls before completing there may be a problem when trying to import certain types of footage. Enable this option if that happens, then try importing again.

## **Check for updates**

When 'Check for updates on launch' is enabled, OpenSesame finds the latest version number of the script posted on [aescrpts.com](#) and compares it to your current copy, alerting you if there is a newer version. Alternatively you can disable this option and use the 'Check for updates now' button instead.

## **Allow helper scripts**

See the '[Helper scripts](#)' section for more information on this. Scripts run by After Effects have a wide range of abilities, including reading, writing and deleting files on your computer, so as a security precaution the ability to run helper scripts is disabled by default and you'll need to enable this pref if you want to use this feature. You also have the option of where these scripts can be located:

### **Installed in OpenSesame script location only**

Helper scripts will only run if they're installed in the same location as the currently running copy of OpenSesame. This will typically be the After Effects... Scripts folder which usually require admin access.

You will only need to include the name of the helper script in the text values file and OpenSesame will only run it if it is found in the same folder as the currently running copy of OpenSesame.

### **Installed in any location**

Helper scripts can be run from any location. If you just include the name of the script in the text values file, it will first look for it in the same location as the text values file, then in the OpenSesame script location (unless running from the Scripts > Startup folder in which case it will look in the main Scripts folder). If you include the full path it will look for it in that location.

## pt\_OpenSesame Server

OpenSesame Server provides a fully automated API based solution for project customization, versioning and rendering to generate content on demand, using either the JSON API for web apps or regular text files that can be edited in a spreadsheet.

Set up any number of project templates and change many things including text, footage, property values, layer switches and render queue items. Changes are made by submitting a text file to a watch folder, whether that's across a network, from a mobile device, database or web app.

OpenSesame Server creates detailed log files, has built-in error handling, can automatically restart and continue rendering or send alerts in case of any failures. You can abort renders remotely and easily prioritize tasks.

Have multiple computers running OpenSesame Server to process and render submissions simultaneously, or have one computer modify the projects and save them for processing on a render farm. It's compatible with the After Effects Render Engine, so you can easily set up a render farm to generate high volumes of content on demand.

## Creating project templates

OpenSesame Server works in conjunction with a separate script called OpenSesame which you use to prepare the project templates. [aescrpts.com/pt\\_opensesame](http://aescrpts.com/pt_opensesame)

With OpenSesame it is possible to export an entire AE project to a text file, but that is primarily intended to create backwards compatible projects by importing then (again using OpenSesame) into earlier versions of After Effects. These text files are much larger, harder to edit and slower to import, so it's more likely you'll want to export just the editable values. Both types of text file can be imported using OpenSesame Server, but these instructions will assume you are using the "Export Editable Values" option.

## Running watch folder mode

Using OpenSesame Server is as simple as launching the script and clicking Start Watch Folder. You'll be asked to choose the location of your watch folder, after which it will continually scan that folder and one level deep into any sub-folders, in alpha-numerical order, looking for a valid text values files.

## Scanning the watch folder

As it scans one folder deep into any subfolders, you have the option of creating separate folders for each of your project templates, perhaps containing any required source footage inside them.

The first line of the text file contains the name of the After Effects project it is associated with. This AE project also needs to be in the watch folder in the same folder that you submit the text file to. Typically you would put all your AE projects inside the watch folder and leave them there. They won't be altered in any way and won't be imported until instructed to do so by a valid text values file.

NOTE: When running OpenSesame Server on a different computer to the one used to create your template projects, you should manually open each project in After Effects to check there aren't any missing fonts, effects or footage that are used in the project. Relink any missing footage and re-save the project. Install any missing fonts or effects, restart After Effects and check the projects again. Otherwise, these issues will cause After Effects to display a warning dialog when OpenSesame Server attempts to open the project. These warning dialogs have to be cleared manually and would therefore interrupt the automation process.

## Text control file (.tmp)

When OpenSesame Server finds a valid text values file it immediately creates a new text file in the same folder with the full file name and extension, stripped of any periods and with the extension .tmp. As it's possible to have multiple machines running OpenSesame Server and scanning the same watch folder, this prevents these other machines from attempting to import the same text values file. They will simply move on to the next available text values file they find.

The .tmp text file just contains the name of the computer that is processing that text values file, so you can query this when running OpenSesame Server on multiple computers to find which one is processing a particular file.

## Processing and rendering

OpenSesame Server locates the name of the associated AE project in the text values file, opens that AE project and makes any changes specified in the text file. It then moves the processed text file to the log folder if you have this option selected in the preferences.

Depending on the text file's render options settings, it will then save a copy of the revised project to a new location and/or render it. See the Render Options section for more information on this.

If your text values file doesn't contain at least one set of Render Settings and OutputModule information, OpenSesame Server will instead render any existing queued Render Queue items in the original AE project, but this will mean you won't be able to change the name and location for the rendered video.

When rendering, any existing renders with the same name will be overwritten without warning so you should ensure the render paths have unique names if you don't want this to happen.

To abort a render in progress, for example if you have a higher priority render you want to push through, simply delete the .tmp file that was created. OpenSesame Server will abort the render and resume scanning the watch folder for more text control files.

Once the render is complete it removes the .tmp file that was created, then starts scanning the watch folder again alphanumerically from the start.

## Email Alerts

In the preferences you can set up email alerts to be notified when a text values file has been processed. You can choose whether to receive an email alert for each import or only in cases where there was an error.

To set this up, check '[Send Email Alerts](#)' in the preferences, then click Configure to enter your mail server and authentication settings. Use the option to send a test email to ensure it has been configured correctly.

## Logs

Detailed logs are created for each processed text file in the OpenSesame Logs folder. The default log folder location is inside your chosen watch folder but there is a preference that lets you choose an alternate location for it.

A new subfolder will be created for each text file that is processed, named based on the date and time it started being processed, along with the name of the text file. i.e. 2014-06-17 09-08-33 test1 VALUES.

The log file starts with the name of the computer that is processing it, followed by the path and name of the text file being processed. If there are any errors during the import and processing of changes they will be written to the log.

If OpenSesame Server is set to render the revised project, the rest of the log file is made up of the detailed render logs that After Effects creates during a render. This includes the output paths, render settings and output modules used for the render.

The log file is continuously updated during a render, logging each frame as it is rendered, so it can be queried if you want to monitor the render progress.

## Using OpenSesame Server with a render farm

Rather than having OpenSesame Server do the rendering, you can set it up so it only applies the changes to the project, before saving a copy of the project that can be rendered on a render farm.

In the Options section of the text values file you will need to set it to not render the project and to save the project either to the logs folder or a path and name you specify.

Also, if you're going to use AE's built-in Render Engine / Watch Folder feature, you will need to set the RenderControl option so OSS generates the necessary render control file (.rcf).

For more information on setting this up, check out the '[Render section](#) > [Render / Save options](#)'.

## Preferences

### Favour absolute/relative paths

OpenSesame text files contain both absolute and relative paths for any footage. Use this preference to select which kind OpenSesame Server will use first when attempting to locate the footage. See '[Additional topics](#) > [Information about file paths](#)' for more information.

### Text encoding

Set the text encoding format used to read/write text values files. See '[Additional topics](#) > [Text Encoding](#)' for more information.

### Watch Folder location

You can choose whether OpenSesame Server will ask you to choose the watch folder each time you click "Start Watch Folder" or it can remember the previously used folder and continue to use that. Click the Watch Folder button to specify a new default watch folder and use the pulldown option to set whether it asks you to select the watch folder each time.

### Start watch mode automatically on script launch

With this option checked, as soon as you launch OpenSesame Server it will start running in watch mode. If you install OpenSesame Server in After Effects' Scripts > Startup folder you can have it set to launch and start in watch mode simply by launching After Effects. If you set After Effects to automatically launch when your computer restarts, you can get the watch mode to resume running in the event of a power failure.

### Render Folder location

You can specify a folder that any renders are saved to. Use the pulldown option to choose whether all footage will be rendered to that folder regardless of path specified in the text file, or just have it set to render to that folder if the output path in the text file is invalid.

## Log folder location

By default, log files are stored in an "OpenSesame Logs" subfolder of the watch folder, but alternatively you can choose a different location for the logs folder.

## Move processed text files to log folder

You can choose whether the text files are moved to the logs folder after importing them. OpenSesame Server does remember the file names and creation date of any text files it processes, so it isn't strictly necessary to have the text files moved but it is recommended. If you have multiple computers watching the same folder then you will need to leave this option checked.

## Allow helper scripts

See the '[Helper scripts](#)' section for more information on this. Scripts run by After Effects have a wide range of abilities, including reading, writing and deleting files on your computer, so as a security precaution the ability to run helper scripts is disabled by default and you'll need to enable this pref if you want to use this feature. You also have the option of where these scripts can be located:

### Installed in OpenSesame script location only

Helper scripts will only run if they're installed in the same location as the currently running copy of OpenSesame Server. This will typically be either the After Effects... Scripts or Startup Scripts folder which usually require admin access. Using this option may be safest if you don't have fully locked down access to the OSS watch folder.

You will only need to include the name of the helper script in the text values file and OSS will only run it if it is found in the same folder as the currently running copy of OpenSesame Server.

### Installed in any location

Helper scripts can be run from any location. If you just include the name of the script in the text values file, it will first look for it in the same location as the text values file, then in the OpenSesame Server script location (unless running from the Scripts > Startup folder in which case it will look in the main Scripts folder). If you include the full path it will look for it in that location.

While unlikely, I feel I should point out the worst case scenario, that someone with access to your OSS watch folder uploads a malicious script and a valid text values file that calls that script. It is up to you what level of caution you feel is necessary when using this feature.

## Send Email Alerts

You can choose to have OpenSesame Server send emails, either for every import or only those that had errors. You will need to configure your email settings first and use the "Send Test Email" option to ensure it is configured correctly.

## Check for updates

When 'Check for updates on launch' is enabled, OpenSesame Server finds the latest version number of the script posted on [aescripts.com](#) and compares it to your current copy, alerting you if there is a newer version. Alternatively you can disable this option and use the 'Check for updates now' button instead.

## Editing OpenSesame text files

OpenSesame can export tab separated (.tsv), comma separated (.csv) or JavaScript Object Notation (.json) format text files. These can be edited in a standard text editor or a spreadsheet app like Excel or OpenOffice can be used to import .tsv or .csv files, displaying each entry in a separate cell.

These files open directly into Excel, but for OpenOffice you may need to create a new spreadsheet and choose Insert > Sheet From File. In both apps you may need to specify if it was tab or comma separated. When you have finished editing and export the file from the spreadsheet app, if there's a text delimiter option (often set as " ") it should be set to have no text delimiter.

You should be careful to retain the existing formatting otherwise OpenSesame could fail when importing the file. The order of the various settings isn't important as OpenSesame Server simply scans each line looking for the setting description, followed by a colon, followed by the value. You should match this convention and ensure there is at least one comma or tab separator (depending on chosen file type) between each setting.

## Deleting settings you don't want to edit

For each item in the Edit List, the exported text file contains a range of settings that can be edited for that item type. You can delete many of the entries in the text values file that control settings that you don't want to edit. This way you can simplify the text file further to make it easier to edit.

In the tab and comma separated text file formats you can freely delete any of the settings after "EDITABLE->" which you don't require.

## Common settings

This is a list of settings that are common to many of the item types in the exported text file.

### Common settings that must not be deleted

The following settings are common to many of the item types in the text values file. They are required in order to locate the item in the After Effects project and must not be deleted.

#### Title

The first part of each line of the .tsv/.csv file or the Title variable for each object in the JSON file are used to specify what editable value type is being referred to.

On the first line of a .tsv/.csv file or the Header object of a JSON file the title "pt\_OpenSesame" is used by to verify that it is a valid text values file.

#### CompID

Most entries contain a CompID setting. This is the internal number that After Effects uses to identify a composition and is what OpenSesame uses to locate it in a project.

The only time you may want to change this value is in the Render Settings as this specifies which composition will be rendered. To find the CompID value for a particular composition, either add the comp or something inside it to the list of editable values before exporting your template.

#### Layer

The layer setting specifies the layer number of the Layer, Text or Value that is to be edited. In general you would not want to edit this but there are times when you might.

For example, if you have a number of similar layers in a composition and want to specify that one of those layers should be turned on (depending on some external criteria) you would turn them all off when exporting the template, include one of the layers in the Edit List when exporting, then edit the Layer

number setting in the text values file to choose which layer is to be edited. In this case, you would also set Video:1 to turn it on.

### **Common settings the are optional and can be deleted**

The following settings are only included to make it easier to understand which project items are being referred to. Changing them won't cause those items to be renamed in the project and they aren't required to locate the items, so they can be deleted from the text values file if you want to minimize the amount of text that it contains.

#### **Name**

The name of the item referred to in the entry, for example the name of the layer.

#### **Comp**

The name of the composition containing the item referred to in the entry.

## **Header section**

The header section is the first line of the .csv or .tsv file or the Header object in the JSON file. It contains the information that identifies this as an OpenSesame text values file, contains the name of the AE project it is linked to, and sets a base footage path for any footage being specified using relative footage paths.

This is a list of the settings contained in the Header section.

#### **Title**

This simply says "pt\_OpenSesame" and tells the script that this is a valid text values file.

#### **Version**

The version of OpenSesame that was used to export the template. Generally it is best if you use the same version OpenSesame (and OpenSesame Server) to export and re-import the text file, as it's possible the OpenSesame text file format may change over time due to bug fixes or additional features. You'll receive a warning if importing a text file created in an earlier version.

#### **AEProject**

The name or file path of the AE project which this text values file will be making changes to. On export, just the name of the AE Project is specified in this setting, and in this case the project must be in the same location as the text values files. However, if you'd prefer to have the AE project located somewhere else, you can do this by including the full file path and name for the project in the AEProject setting.

#### **FootageBase**

The base footage path used to find footage that has been specified using a relative path. This setting will only be added if there are footage items in the Edit List and you've selected relative footage paths in the Edit List export options.

By default this will be the path that the project template was exported to and any relative paths will be relative to this location. Where possible, OpenSesame will export absolute and relative file paths for any footage.

When OpenSesame is locating footage it will try an additional FootageBase path, the location where the text file was imported from, if this is different from the specified FootageBase path. So even if you have moved the project and footage since the template was exported, as long as the footage remains relative to the template's current location the footage should still be found.

Because of this, you can actually use the FootageBase to specify a second base path location, for example if some footage is stored on a different drive or network. This means you can use relative paths for this footage too and it can still be found by OpenSesame.

## Encoding

You can override the encoding setting that OpenSesame uses during import by adding an "Encoding:" setting in the header section. This way you can submit text values files with different encoding formats and OpenSesame should import them correctly, regardless of the 'Text encoding' import preference.

## Main section

Subsequent lines in the .csv or .tsv files, or the Main object array in the JSON file, contain information for all the Footage, Comps, Layers and Properties that were marked to be editable. Which of these settings are exported will depend on the selections in the Edit List export options.

## Footage item

This is a list of settings specific to the Footage item type

### FootageID

As with CompID this value refers to the internal number that After Effects uses to identify a footage item. You should retain this setting and not change it. It can be useful to know what this value is though, as you can change the FootageID setting on a Layer entry to make OpenSesame change the footage being used for that layer.

### ImageSeq

This is a value of 0 or 1 which specifies whether the footage is an image sequence. If the footage is a still or video it should be set to 0. If it is an image sequence it should be set to 1. You will only need to change this if you're changing the path to choose some different footage and that footage is of a different type to the original.

### Path

This specifies the absolute path where the footage item is located. Change this or the Relative Path to another footage location to change what footage is being used in the project. If you keep both Path and RelativePath settings then OpenSesame will first try whichever of the two is set as favoured by the "Favour absolute/relative path" preference. It will use the first valid footage path it finds.

Typically you would decide which path type you prefer to use and delete the other one from the text file.

### RelativePath

This specifies a path to where the footage is located, relative to the location where the project file was exported.

## Comp item

Currently there are no editable settings for a composition. It simply contains the name of the composition and the CompID. Knowing the CompID can be useful if you want to be able to change which comp is being rendered in the Render Settings.

It is purely included for CompID reference and can be removed from the text file once that is known.



## Layer item

This is a list of settings specific to the Layer item type.

### FootageID

For layers that have a footage source this refers to the internal After Effects number for that footage item that is already in the project. Change this to the ID of a different footage item to have OpenSesame change the footage used for this layer. It can be removed if you don't need to change the footage source. So if you have footage that you want to change, you can either change the file path of the footage item (on the Footage line change Path or RelativePath), or you can have multiple footage items in the project already and just change which footage item is being used for a particular layer.

### Video

This controls whether a layer is on or off, the equivalent of turning the eyeball icon on/off on a composition layer. The only layers that don't have this setting are audio only layers. Set to 0 to turn the layer off, 1 to turn it on.

### Audio

If the layer has audio, this setting will control whether the audio is on or off. Set to 0 to turn the audio for this layer off, set to 1 to turn it on.

### startTime

This controls the startTime for the layer. This is the time into the composition in seconds of this layer's original untrimmed start time, which may not be the same as the layer's inPoint if it has been trimmed. Changing this will offset the layer and any keyframes it contains in time, with the in/out points moving with it.

### InPoint

This is the inPoint in seconds for the layer relative to the start of the composition. Changing this will trim the inPoint of the layer while not changing the timings of any keyframes or the outPoint.

### OutPoint

This is the outPoint in seconds for the layer relative to the start of the composition. Changing this will trim the outPoint of the layer while not changing the timings of any keyframes or the inPoint.

## Text item

This is a list of settings specific to the text item type.

NOTE: Due to a limitation with AE scripting, OpenSesame cannot edit a text layer and retain multiple fonts, sizes or colors contained in a single text layer. Editing any text layer using OpenSesame will cause it to have a single style for all characters. If you need to edit a text layer containing multiple styles you should break it up into a separate text layer for each style before exporting the template.

### Font

The font used for the text layer. If you want to change the font you should use the exact name for that font as used by AE scripting. To find the proper font names, create a text layer for each font you want to use and export a template with these text layers included in the edit list.

### FontSize

The font size for the text layer. This is the same as the font sizes shown in the After Effects Character panel.

**FillColor**

The Fill Color for the text layer.

**Values**

This setting contains the text contents for the text layer.

**Times**

If the text layer's Source Text property has keyframes this setting will include the time in seconds for each of those keyframes.

**Values item**

This is a list of settings specific to the property values item type.

**PropName**

The name of the property. This is included to make it easier for you to know which property is being referred to. It is not an essential item and can be deleted if you wish. Changing the PropName setting will have no effect on the template. For effect properties, the PropName will contain both the effect and property name in the format effectName|propertyName.

**Property**

This describes the path to the property from the layer that contains it. It is an essential setting used to locate the property and must not be deleted. In most cases you won't want to edit this.

The property path is made up of a series of match names, the internal naming system that After Effects uses to describe each property and the property group that it is in. For mask, shape or effect properties, part of the path will be a number which refers to the index for that particular item in its group. Each part of the path is separated by a vertical bar symbol.

**Values**

This setting contains the values for the specified property.

**Times**

When exporting a keyframed value, the additional Times setting will contain the times in seconds for each keyframe.

**Examples of property values****Static values**

Non keyframed values are stored under "Values" as either single values for one-dimensional properties or multiple values for two/three dimensional values. The values of a multi-dimensional property are separated by semi-colons in tsv/csv formats or stored as an array in the JSON format.

**Examples**

A static one dimensional property like Opacity property:

**TSV/CSV**

Values: 100

**JSON**

"Values": 100

A static three dimensional property like Position on a 3D layer:

**TSV/CSV**

Values: 320;240;0

**JSON**

"Values": [ 320, 240, 0 ]

**Keyframed values**

For properties with keyframes, these are saved as sets of values with each keyframed value separated by a vertical bar symbol in tsv/csv formats or as an array of keyframe values in JSON format. The times (in seconds) for each keyframe are stored under "Times", with the values separated by semi-colons in tsv/csv formats and as an array in the JSON format.

**Examples**

An Opacity property with two keyframes, fading in over the first second of the composition:

**TSV/CSV**

Values: 0;100

Times: 0; 1

A Position property on a 2D layer with two keyframes moving from top left to bottom right:

**TSV/CSV**

Values: 0;0|640;480

Times: 0; 1

**JSON**

"Values": [ [ 0, 0 ], [ 640, 480 ] ]

"Times": [ 0, 1 ]

NOTE: When OpenSesame imports keyframed values and there were already keyframes in the template project, the keyframes will be replaced but any existing keyframe interpolation settings will be preserved.

If you don't want to change the keyframe times then the "Times" setting can be removed from the text file. Values will be applied in order at the times of the existing keyframes. If the number of values doesn't match the number of keyframes, additional keyframes will be removed, additional values will be ignored.

**Color values**

RGB color values are stored as three decimal values between 0.0 and 1.0, separated by semi-colons. It isn't possible to specify high dynamic range values greater than 1.0 as this isn't supported by scripting in After Effects.

**Example**

The Color property of the Fill effect, set to red:

**TSV/CSV**

Values: 1;0;0

**JSON**

"Values": [ 1, 0, 0 ]

## Text values

On a text item, the text contents of that layer are stored under Values. The text may have been encoded so it can be safely stored in the text file, depending on the export format and what characters it contains. See the Special Characters section for information on how text may be encoded.

If there are no keyframes the text will be a single string for the text contents. If there are keyframes they will be stored in the same format as other keyframed values, with both Values and Times settings.

IMPORTANT NOTE FOR TSV/CSV EXPORT: The text values setting is unique in OpenSesame in that the "Values:" heading and the actual text are separated by a comma/tab when exporting editable values as TSV or CSV. This is designed to make it easier to edit the text when using a spreadsheet, as the actual text contents will be displayed in a separate cell.

### Examples

The text value for a text layer without keyframes:

#### TSV/CSV

Values:some text

#### JSON

"Values": "some%20text",

The text values for a text layer with keyframes:

#### TSV/CSV

Values:some|keyframed|text

Times:0|2|4

#### JSON

"Values": [ "some", "keyframed", "text" ],

"Times": [ 0, 2, 4 ]

## Checkbox values

Checkbox properties are used in various effects such as the Expression Controls > Checkbox Control effect. Checkbox values are stored as either 0 for unchecked and 1 for checked.

### Example

The Checkbox property of the checkbox Control effect set as checked:

#### TSV/CSV

Values: 1

#### JSON

"Values": 1

## Scripts section

These are the generic settings used for all helper scripts. A helper script may also add additional settings specific to that script.

### PostRender

[Helper scripts](#) can run either 'pre render', after the changes in the text file have been applied but before any render items are set up and the project is rendered and/or saved, or 'post render', after any renders have completed. By default, all helper scripts are added as pre render scripts so this is set to "false". If a helper script is designed to always be used 'post render' it can override the default, or you can manually edit it to `PostRender:true` after export.

NOTE: Post render scripts will only run if the '[Render / Save options](#)' in the text file are set to render, not when the project is only being saved.

### ScriptPath

The filename or full file path for the helper script. If the helper script is in the same location as the running copy of OpenSesame during export, just the filename will be exported, otherwise the full file path will be exported.

If only the name is specified, if the helper security pref allows it will first attempt to locate the script in the folder containing the submitted text file, followed by the location of the running copy of OpenSesame. If you're running OpenSesame Server from the `Scripts > Startup` folder it will instead look for the helper script in the main `Scripts` folder.

## Render section

### Render Settings

If you delete all the "Render" lines in the values text file it will leave the project's existing render queue items untouched. For OpenSesame Server this will cause it to render any existing queued items to their original file paths.

If you retain the "Render" line you must at least have the "CompID:" value which identifies the numeric ID of the composition that will be rendered. If you delete the "Start:" and "Duration:" values it will default to rendering the work area range specified in the original AE project. The "RenderSettings:" will use the default render settings template specified in After Effects on the render machine, or you can change it to the name of another template. That template must already be set up on the render machine otherwise the default template will be used.

### Output Module settings

If you retain the "Render" lines you must ensure there is at least one subsequent "OutputModule" section, and for that there must at least be a filename specified under "Path:". If you only specify a filename, or if the file path is invalid:

- OpenSesame will set the folder path to your Documents folder.
- OpenSesame Server will render that item to the render folder specified in the OpenSesame Server preferences.

You can set the "OMName:" to another output module template to change the file format and size of the render. This template must already have been set up in After Effects on that computer. If the template isn't found it will revert to the default template used by AE on that machine.

You can add as many "Output Module" lines as you like after each "Render" line, each with their own file path and output module template.

Note: When OpenSesame has to use default Render Settings and Output Module templates, these are the templates that are set to be applied by default on that particular machine, which is not necessarily the same as the template settings saved in the template project. The only time it is guaranteed to use the exact templates saved in the original AE project are when there are no "Render" lines in the text values file, but this also removes the ability to specify a new file name and path for the render. In other words, if you rely on using the default Render Settings or Output Module templates you should ensure they have been set correctly as the default templates in After Effects on the render machine. For Render Settings the default template is "Best Settings" which is most likely what you will want anyway.

## **Render / Save options**

If render items have been exported there will be an "Options" section at the end of the text file that lets you specify whether the project should be rendered and/or saved to a different location once the changes have been applied.

### **Render Project**

By default this is set to "RenderProject:true" and the project will be rendered after changes are applied. When using one of the SaveProject options to hand off the project to a render farm you would set this to "RenderProject:false" instead.

### **Save Project**

Change the "SaveProject" string to one of the following options:

#### **Off**

Project will not be saved. This is the default setting and assumes you will be rendering locally.

#### **To\_Path**

Save the changed project to a specified path. You could use this to hand off the changed project to a render farm instead of rendering locally. NOTE: If you select this option you must also specify a path and filename for the project in the "Path:" section.

#### **To\_Log\_Folder**

Save the changed project to the log folder alongside the moved text file. Useful for troubleshooting, as you can review the project state after the changes were made but before it started rendering. You can even use this option when creating a render farm with the AE Render Engine by choosing the log folder as the render engine watch folder.

#### **RenderControl**

If you've chosen to save the project, you can use the RenderControl setting to make OpenSesame create an After Effects render control file (RCF) which is needed if you're going to use AE's built-in Watch Folder mode for rendering the altered projects. See the After Effects help regarding 'Automated rendering and network rendering' for more information on AE's watch folder and render engine.

To have these RCF files created (in the same location as the saved project) you should set RenderControl to a value higher than zero. This value will set the maximum number of computers that will render the

project. Only image sequence renders are suitable for multi-machine rendering, so for anything else you should set RenderControl:1

## Path

If you use "SaveProject:To\_Path" then you must also specify a path and filename for the saved After Effects project in the "Path:" section.

## Additional topics

### Text Encoding

Specify the text encoding format used to read/write text values files using the '[Text encoding](#)' preference in the Help Window. The default encoding is UTF-8, a unicode format that can contain characters in all languages, and this is still the recommended format to use.

However, Excel generally has poor support for UTF-8, so if you're finding that foreign language characters are getting lost when editing the text files in Excel you can try one of the other encoding formats. MS\_ANSI is Microsoft's legacy format for western languages so this can work better for Excel.

Be aware that OpenSesame can have problems reading text files if the encoding format for the text file doesn't match the "Text Encoding" pref, so you should be consistent with your encoding format for both export and subsequent imports.

The only recommended encoding formats are UTF-8, or MS-ANSI for use with Excel. The other options have been included in case they are helpful, but they haven't been tested.

NOTE: You can override the encoding setting that OpenSesame uses during import by including an "Encoding:" setting in the header section of the text values file. This way you can submit text values files with different encoding formats and OpenSesame should import them correctly, regardless of the Text Encoding import preference. See the '[Editing OpenSesame text files > Header section](#)' for more information on this.

### Special Characters

#### CSV/TSV Export

Certain characters are converted in order to be safely stored in the text file. When making changes to the text values file you should ensure you follow these conventions.

The following characters are converted:

\n (carriage return) is converted to %n%

\r (line feed) is converted to %r%

Commas are converted to %c%

Tabs are converted to %t%

Note: Depending on whether you export as a comma or tab separated text file, there's a preference to preserve the character that isn't being used as the separator. So if you're going to be editing text and would prefer to retain commas as they are, you can choose to export as tab separated. With the '[Preserve commas / tabs](#)' preference enabled this will prevent commas from needing to be converted.

If you want to set up a spreadsheet so someone can enter text while using regular line returns, you could place the actual text entry cells in a lower part of the spreadsheet under the "End" line then link to those cells from the intended destination in the Text settings line.

For example, use this in the Text Values cell to link it to a lower cell, substituting any line returns with the required %r%:

```
=SUBSTITUTE(K21,"  
","%r%")
```

You can nest these functions to substitute multiple different characters such as this example which will replace both line returns and commas:

```
=SUBSTITUTE(SUBSTITUTE(K21,"","%c%"),"  
","%r%")
```

By moving the data entry to a lower part of the spreadsheet there are many ways you can help make it easier for someone to change the values, such as setting up in-cell dropdown lists in cases where you want to offer a limited set of choices.

## Excel Safe Export

Excel in particular can fail to preserve certain other characters, so there's a preference to also convert these during export. If you're not using Excel you can leave this option turned off.

The extra characters that will be converted are:

Apostrophes are converted to %a%.

Standard quotation marks are converted to %q%

Left quotes are converted to %lq%

Right quotes are converted to %rq%

If you're finding that non-English alphabetical characters aren't being preserved you could try changing the 'Text encoding' preference to use a Microsoft legacy encoding option such as MS-ANSI for export and subsequent imports.

## JSON Export

When expressions are exported they are always encoded using the standard Javascript encodeURIComponent method and should be encoded the same way if modified and resubmitted.

While a text layer's Values setting isn't encoded on export you should consider encoding them using the encodeURIComponent method for your JSON submissions. OpenSesame Server will always attempt the decodeURI method on text Values, and you may run into problems if the text includes certain characters and isn't encoded when submitted.

## Information about file paths

OpenSesame Server writes file paths using a platform independent format called universal resource identifier (URI) notation.

If in any doubt about how a file path should be constructed, the simplest method is to export a template project containing a footage or render queue item at that location, then see how the file path looks in the exported text file.

Paths for files located inside your user folder begin with ~/ while paths outside the user folder or on a different volume begin with just a forward slash. A forward slash is also used to separate each directory in the path.

To access footage on a remote volume, you can use a uniform naming convention (UNC) path name of the form //servername/sharename. On Windows this gives you an alternative to using the drive letter assignment, which is how footage paths are named on export from OpenSesame.



## Relative file paths

If a footage item is on the same volume that the template project was exported to, OpenSesame Server will include a RelativePath to that location, relative to the location where the template project was exported to (which is stored in the FootageBase setting in the text files header).

When editing relative paths you should not include a forward slash at the end of the FootageBase path or the beginning of the RelativePath.

If the footage item is located on the same volume but outside of the FootageBase location you will see one or more ../ at the beginning of the RelativePath. This represents the parent folder of the FootageBase location.

So for example, if you had both a Watch Folder and a Footage folder in your Documents folder you might see something like:

FootageBase: ~/Documents/Watch Folder

RelativePath: ../Footage/myfootageitem.jpg

In this case, the relative path is starting with the parent of the Watch Folder (i.e. Documents).

## Text formatting controls

When using OpenSesame to change the contents of text layers there are a number of ways you can control how to accommodate text of varying lengths to constrain the area it takes up in your design.

### Point or Paragraph Text

Decide whether to use point or paragraph text. Point text is created when you click in the composition with the text tool or choose Layer > New > Text. Point text won't word wrap automatically and requires you to include line returns for the text to flow into another line.

Paragraph text is created when you click and drag in the composition with the text tool, dragging to select a box in which the text will be constrained. Text will automatically word wrap once it reaches the edge of the defined box, and if you try adding more text than it can contain it will simply not display the additional text.

### Text justification

Use the justification options at the top of AE's Paragraph panel to choose whether text is Left Aligned, Right Aligned or Centered. This will ensure text remains correctly aligned to your design template however many characters are used.

### Scale

There are a number of ways you can change the size of the text to accommodate varying numbers of characters.

You could change the FontSize setting on the Text item in the text values file depending on the number of characters being applied to the text layer. You could also do this by adding the text layer's Scale property to the Edit List and changing this, giving you the option to change the scale width independently of the scale height. You could even add an expression to the Scale property that would automatically change the scale based on the number of characters. But these options all require some testing to find the right settings to use depending on the number of words/characters being applied.

### Auto Scale

OpenSesame Server has an additional option for more accurately altering the size of a (point text only) text layer depending on its contents. When setting up your AE project template, if you rename a text layer in a specific way it will automatically resize the text layer to fit if it goes over a certain size.

Select a text layer in the Timeline panel (where the layers in the composition are listed) then press Enter to enable renaming of the layer. Note: Renaming the text layer is different from changing the actual contents of the text layer as seen in the composition. Rename the text layer using one of the following formats:

WIDTH 500  
FIT 500

Rename the text layer to either WIDTH or FIT, followed by a space, then the width in pixels you want to constrain the text to. WIDTH causes only the width of the text layer to be reduced, while FIT will uniformly scale the width and height to constrain the text inside the chosen pixel width.

In both cases, the scale will never exceed 100% so text with a small number of characters won't be scaled up to fit. Text will only be scaled down if it would otherwise exceed the specified width.

NOTE: When setting this up, ensure the text layer hasn't been animated or rotated. You may want to precompose the static text then animate it in a subsequent comp. Additionally, don't include the text layer's Scale property in your Edit List. If you do, this setting may be applied after the auto scale has rescaled the text layer, causing the setting to be overridden.

## Automatically resize footage elements using expressions

If you're using OpenSesame to change footage elements (such as photos) that may be different sizes, you can use expressions on the Scale properties of those layers when setting up your project so they are automatically resized depending on their original dimensions.

NOTE: This technique won't work on text layers. See the 'Text formatting controls' section for ways to control text formatting.

Here's an example of a Scale expression that will fit a still or video element to the size of the composition, scaling it uniformly to the maximum scale needed to fill the composition:

```
x = (thisComp.width/width)*100;  
y = (thisComp.height/height)*100;  
s = Math.max(x,y);  
[s,s];
```

Here's an example where you can specify the final dimensions. It will non-uniformly scale the element to that exact size:

```
finalSize = [500,500]  
x = (finalSize[0]/width)*100;  
y = (finalSize[1]/height)*100;  
[x,y];
```

There are many other ways of using expressions to help your project react to changes without having to specify everything in the text file. One simple example is to link multiple properties of the same type and value to a single 'master' property, so you only need to add that one 'master' property to the Edit List and can change all instances by just editing that one value in the text file. For example, you could have multiple copies of the Color Balance (HLS) effect linked through expressions so you could change the color scheme of the entire project by just editing a single set of Hue and Saturation values.

## Helper scripts

Use helper scripts to extend OpenSesame's functionality in all kinds of ways, or to make changes that would otherwise be difficult to accomplish just by altering the text values file. You can run as many helper scripts as you like, and choose whether they run before or after any renders are completed.

This feature is primarily intended for OpenSesame Server's fully automated project versioning and rendering workflow, and is included in OpenSesame so you can test helper scripts with your project templates before deploying to OpenSesame Server, but regular OpenSesame users may find it useful too.

### How to use

While setting up your editable values for export, click the 'Add Script' button in the Edit List, then locate the script you want to include as part of the import process.

By default, helper scripts are set to be applied 'pre-render', after the changes in the text file have been applied, but before the render items are set up and the project is rendered and/or saved. To have them applied 'post-render' you'll need to set `PostRender:true` in the text file's Scripts section. See the '[Editing OpenSesame text files > Scripts section](#)' for more information.

NOTE: Post render scripts will only run if the '[Render / Save options](#)' in the text file are set to render, not when the project is only being saved.

### Security

As a security precaution, the ability to run helper scripts is disabled by default. See the '[Preferences > Allow helper scripts](#)' section for information on enabling this.

Scripts run by After Effects have a wide range of abilities, including reading, writing and deleting files on your computer, so you should only use scripts obtained from trusted sources.

## Existing helper scripts

The following helper scripts are currently available. `OSHelper_ArrangeLayers` is included with your copy of OpenSesame. The rest are available for purchase on [aescripts.com](http://aescripts.com).

### OSHelper\_ArrangeLayers

This helper script is included with your copy of OpenSesame. A common issue among OpenSesame Server users is that, after using OSS to replace footage, you may want to rearrange the layers and alter the comp duration if the footage duration has changed. This isn't so easy to do just by specifying changes in the text file as you may not know what the new footage duration is, but it's a perfect example of what helper scripts can be used for.

NOTE: If you're changing the duration of the comp / work area and want these changes reflected in the render duration, you should remove the `Start:` and `Duration:` settings from any Render Settings lines in the text values file. Removing these settings will cause the render durations to adopt the altered comp work area settings instead.

### pt\_ImportSubtitles

[aescripts.com/pt\\_importsubtitles](http://aescripts.com/pt_importsubtitles)

The latest version of `pt_ImportSubtitles` can either be used manually as a standalone UI script or as a headless OpenSesame helper script. Use with OpenSesame Server to fully automate importing subtitles

into your project. You can specify the subtitle file to import along with most other settings you'd normally access through the pt\_ImportSubtitles user interface.

## **TM\_HebrewText**

[aescrpts.com/typemonkey-hebrew-text-modifier](https://aescrpts.com/typemonkey-hebrew-text-modifier)

The latest version of the Type Monkey Hebrew Text mod can also be used as an OpenSesame helper script. If you're making text changes in Hebrew you can apply them with the character stored in logical order and have TM\_HebrewText convert them to the display order needed to appear correctly in After Effects.

## **TM\_ArabicText**

[aescrpts.com/typemonkey-arabic-text-modifier](https://aescrpts.com/typemonkey-arabic-text-modifier)

The latest version of the Type Monkey Arabic Text mod can also be used as an OpenSesame helper script. If you're making text changes in Arabic you can apply them with the character stored in logical order and have TM\_ArabicText convert them to the display order needed to appear correctly in After Effects.

## **Getting more helper scripts**

Scripts need to be specially designed to work as OS Helper scripts. If there are other scripts on sites like aescrpts.com that offer functionality you would like to use in OpenSesame Server's automated environment, you should contact the author to see if they'd be prepared to add OS Helper Script support.

While I'm including helper script support in dual usage versions of the above scripts (at no additional cost if you've previously purchased them), due to the niche nature of the OpenSesame Server user base it's more likely that other script authors will make them separate scripts, sold at a premium price.

Please open a support ticket at aescrpts.com if there is a helper script you would like. If you have an OpenSesame Server SLA ( Service Level Agreement) then I may be able to help as part of that, depending on the complexity of what you require. Otherwise myself or other script authors may be available for hire to provide a solution.

## **Creating helper scripts**

### **OSHelperScript\_Run()**

The one thing you need to do to create a working helper script is to make sure everything is inside one or more functions, and have a function called OSHelperScript\_Run() as the starting point for your helper script. On import, after OpenSesame evaluates the script file it will call this function.

So the most basic example would be:

```
// CODE START
function OSHelperScript_Run() {
    alert("hello world");
}
// CODE END
```

But don't do that! Helper scripts are designed to be used with OpenSesame Server's automated workflow, so putting up an alert would require user intervention and bring everything to a halt!

If all you want to do is to create simple helper scripts that rely on no external data other than what they can access through the After Effects project itself, that's really all you need to know to get started.

But there's also much more to it if you feel like reading on...

## **OSHelperScript\_Setup()**

You can optionally include a function called `OSHelperScript_Setup()` if you want settings specific to your helper script to be included in the initial OpenSesame text values file export. During the export process, the helper script is evaluated and this function will be called if it exists, allowing you to define additional settings as part of the `OSHelperScript_Data` object.

An example of this would be:

```
// CODE START
function OSHelperScript_Setup() {
    OSHelperScript_Data.DoThisCoolThing = true;
}
// CODE END
```

Which would result in the exported text file containing a line like:

```
Script, PostRender:false, ScriptPath:myHelperScript.jsx, DoThisCoolThing:true
```

## **OSHelperScript\_Data**

Helper scripts have access to the `OSHelperScript_Data` object for sending and receiving information from OpenSesame / Server both while the helper script is running during an import, but also during export to supply OpenSesame with additional script specific settings you want included in the exported text file.

### **OSHelperScript\_Data - import and export**

#### **OSHelperScript\_Data.JSON**

Read only, true or false. Most useful during the `OSHelperScript_Setup()` process on export, you can find out if the text values file is being exported in JSON format.

Use this information to decide if your own script specific settings can include arrays and special characters like commas and tabs, or if you need to ensure they are strings suitable for exporting to TSV and CSV files.

This is also accessible during the `OSHelperScript_Run()` process on import to find if the imported text values file is in JSON format. It's probably not such a good idea to rely on this though as you can easily parse strings and check for instances of arrays.

The included `OSHelper_ArrangeLayers.jsx` script shows an example of how to allow for either types of values on import without needing to rely on `OSHelperScript_Data.JSON`.

#### **OSHelperScript\_Data.PostRender**

During an OpenSesame import you can read the `OSHelperScript.PostRender` boolean to determine if the helper script is running either pre or post render.

During export, all helper scripts exported to the text values file are set as pre-render scripts. If your helper script is designed to always be used as a post render script you can override this by including in the setup function:

```
OSHelperScript_Data.PostRender = false;
```

This is the only time when this setting should be included and would be done simply to save the user from having to make this change manually.

### **OSHelperScript\_Data - import only**

If you used the setup function to add custom settings during import, these will be available to your helper script when run during an import, providing the user didn't delete those settings from the text values file before submission. You should always check they aren't "undefined" so you can handle those situations correctly.

### **OSHelperScript\_Data.RelativePathsArray**

This is a copy of the array of relative file paths used by OpenSesame. If there's only one item in the array it will be the location of the imported text file. If the text file specifies an additional "FootageBase" path in the Header section then this will also be included.

This could be useful if your script specific settings ask for a file location, but you're happy for the user to just supply a file name and let you find it by adding that to the relative path location.

### **OSHelperScript\_Data.Message**

Use this to send a string back to OpenSesame which will be included in the import logs. Typically this would be to report whether the helper script completed successfully. If the string contains the word error it will be treated as such, and could set off an 'email alert on error notification' if that is enabled in the OpenSesame Server prefs.

### **Custom script specific settings**

As I've mentioned, you can add any number of your own script specific settings during the setup process by adding them to the OSHelperScript\_Data object. You should avoid using any of the object variables listed above except for their intended uses, otherwise at best they will be ignored, or at worst could lead to unpredictable behaviour.

As I touched on talking about OSHelperScript\_Data.JSON, great care needs to be taken both when supplying default values for custom script settings and when interpreting the results.

Anything stored in a TSV or CSV file must be in the form of a string (or number / boolean which will be converted to a string anyway) and must not contain tabs, commas or line returns. You may want to adopt the same OpenSesame practices explained in the '[Additional topics > Special Characters](#)' section.

A pretty common technique used in OpenSesame for storing multiple values is to arrange them into a string of values separated by semi-colons and/or vertical bars, then use the Javascript split() method to break them into arrays which you then loop through using parseInt() on each element.

### **Dual use scripts**

pt\_ImportSubtitles is an example of a script which can run both as a typical AE script with a user interface when launched through After Effects or as a headless helper script when run by OpenSesame.

The basic structure I'm using to set up a script this way is as follows:

```

function OSHelperScript_Setup() {
    // function called during OpenSesame export
    // set up custom settings here;
}
function OSHelperScript_Run() {
    // function called during OpenSesame import
    if (OSHelperScript_Data !== undefined && OSHelperScript_Data !== null) {
        pt_ImportSubtitles(OSHelperScript_Data, helperMode = true);
    }
}
function pt_ImportSubtitles(thisObj, helperMode) {
    // the main script function
    if (helperMode) doStuffWithoutUI;
    else buildUI;
}
if (typeof OSHelperScript_Data === "undefined" || OSHelperScript_Data === null) {
    pt_ImportSubtitles(this, helperMode = false);
}

```

The `OSHelperScript_Data` object is always nullified by OpenSesame when not directly being used, so you can use this to determine whether the script is being evaluated by OpenSesame or launched directly by the user.