



CompsFromSpreadsheet is an After Effects script that will allow you to create limitless copies of your original comp, filling in text and replacing layers based on data you supply from a spreadsheet.

For a quick intro to CompsFromSpreadsheet watch this [quick intro](#)

## Installing CompsFromSpreadsheet

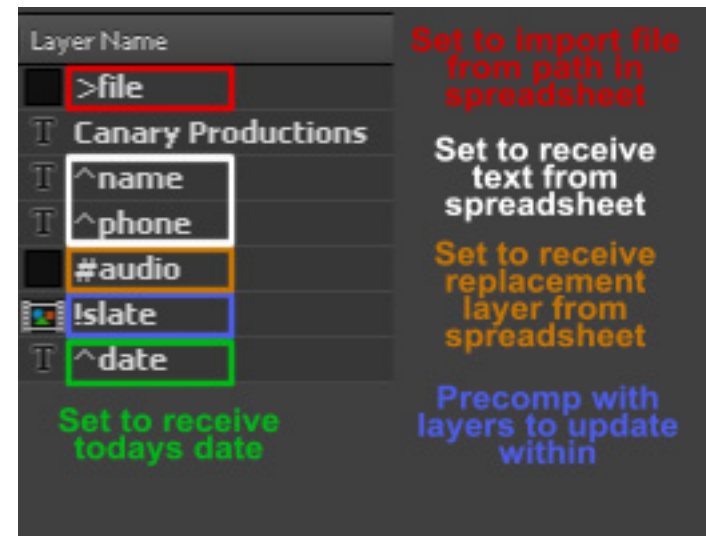
The compsFromSpreadsheet script is meant to be run as a dockable panel. This means it needs to be saved in “Adobe After Effects CS(your version here)/Support Files/Scripts/ScriptUI Panels/” folder.

1. Save the “compsFromSpreadsheet.jsxbin” file into the ScriptUI Panels folder
2. Restart After Effects.
3. Click on the After Effects “Window” menu.
4. Way down at the bottom of the “Window” menu click on “compsFromSpreadsheet”, the compsFrom-Spreadsheet interface will load onto a new panel.
5. A window will pop up allowing you to enter a license, or run a fully functional demo.
6. In After Effects General Preferences, make sure that “Allow Scripts to Write Files and Access Network” is enabled.

## Creating Template Comps

In order to use CompsFromSpreadsheet you need to create a template composition to receive data from your spreadsheet.

- Text layers that begin with ^ (shift 6) will be able to receive data from the spreadsheet as the source text for that layer. If you need a new line or comma in a text layer DO NOT put these characters in your spreadsheet. Instead use <br> for a newline and <c> for a comma and <qq> for “ and the script will replace these symbols with their intended values at runtime.
- If you have a text layer named ^date it will receive today's date (great if you're generating slates) default format is MM/DD/YYYY numerically. You can switch to a different date format by clicking on the settings window and going to the Misc tab.
- If you have a non-text layer in your comp that begins with #, it will be replaced by the project item named in your spreadsheet.
- If you have a non-text layer in your comp that begins with >, it will be replaced by a file. If the file is an image sequence use \* instead of >. The script will import a file based on the path in your spreadsheet, then that imported file will replace the layer. If you have the same file referenced more than once in your spreadsheet, the script will only import it once and use that item for every other time the file is referenced. Instead of putting the path in your spreadsheet, you can define a [default import directory](#). If you have a precomp layer in your comp that begins with !, any layer in the precomp setup for compsFromSpreadsheet will be updated from the spreadsheet.



You can designate a [layer to control the length of the composition](#) by putting a ~ at the end of the name. So if you have a layer that is named >audio which will receive imported audio files you can add a ~ to the end making the layer name >audio~ this will cause this layer to dictate the length of the comp. So when the script replaces that layer with the imported audio file, it will adjust the compositions length so that the comp ends on the last frame of the imported audio file.

If the composition is lengthened, then all the layers are lengthened to the new comp time. If you have a layer that you do not want lengthened, just lock the layer before running the script.

## Setting up your spreadsheet

SAMPLE SPREADSHEET					
	A	B	C	D	
1	COMP	name	File	Iscl	audio layer
2	spot1	Oakland	~/desktop/b.jpg	464654	audio1.wav
3	spot2	Florida	~/desktop/c.jpg	645645	audio2.wav

4	T	^name
5		>file
6	T	^date
7	T	^isci
8		#audio

This column will tell the script which comp to use as a base for this line

Replace layer 5 with jpeg from the desktop

Today's Date

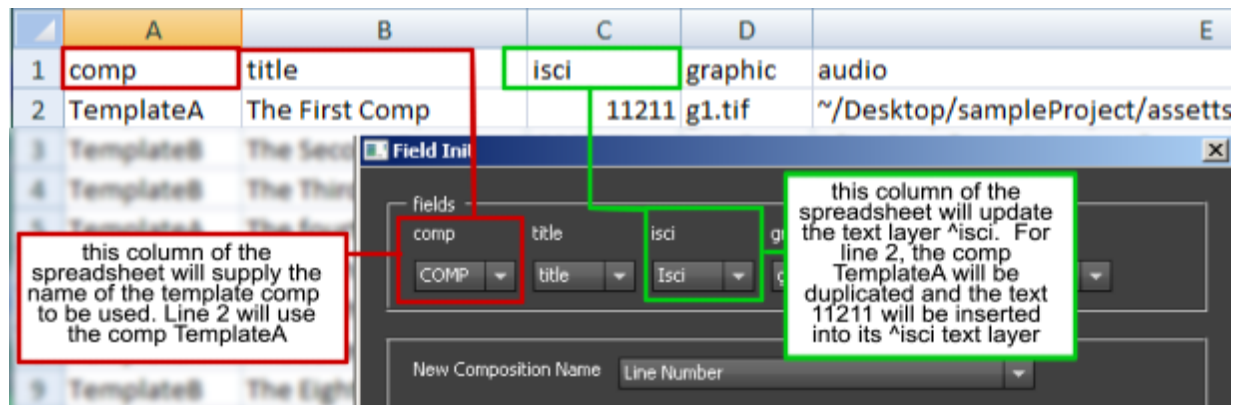
Replace layer 8 with audio1.wav

The spreadsheet is the way that you create copies of your template comp and update text and layers.

The first line of your spreadsheet must contain the names for each column. If you have more than one composition in your project, you will need a column named 'COMP' that will hold the name of the comp to copy for that line.

When you run the script it will take each of these column names and allow you to connect that column with target layers in your template comp(s). If you name the columns the same name as the layers they are going to update, these items will automatically be selected.

## Synching Column Data to your Comp



When the script reads the first line of your spreadsheet, it needs you to tell it where the data from each column should go in your comp. The Field Init window opens. Each column name from the first line of the spreadsheet appears with a dropdown box underneath.

The dropdown box links the data from that column in the spreadsheet to elements in your template comps. The dropdown box contains the names of all the layers that were set up earlier to receive data (remember, the text layers that started with ^ will receive the data from this column, layers with a # will be replaced with the project item named in this column, and layers with > will be replaced by files imported from the location specified in the column.)

If a column name is the same as an element name in one of your comps, that will be automatically selected in the dropdown i.e. If you have a column in your spreadsheet

“phone” and a text layer in one of your template comps named “^phone”, that item will be automatically selected in the

dropdown box. If however, your text layer is “^phone number” it will not be selected in the dropdown box.

If you have more than one comp to be used you will need to have a column named *COMP* in your spreadsheet with the name of the comp to be used for that line. This is not necessary if you only have one composition in your project

## Converting Your Spreadsheet

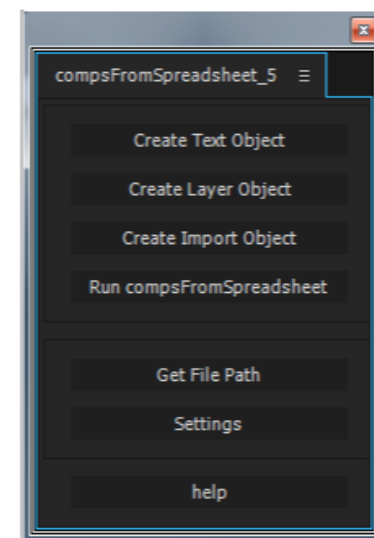
Before After Effects can read your spreadsheet, it needs to be converted into a tab delimited text file (most spreadsheet programs do this easily). This is a universal text format that will allow After Effects to read in your data and organize it by rows and columns.

When you run the script be sure to point it to this newly created text file and not your spreadsheet as After Effects cannot read any spreadsheet format (maybe I should call the script `compsFromTabDelimitedTextFile`).

## The Interface

The first three buttons of the `compsFromSpreadsheet` interface allow you to quickly make target layers for the script in the comp you are viewing.

While in a comp clicking the “*Create Text Object*” button will bring up a window asking for the text object name. When you click ok the script will make a new text layer set up to receive data from the column name specified. So if you entered “isci”, a new text layer named “^isci” will be created in the comp you are viewing; this will receive text from the “isci” column of your spreadsheet.



While in a comp clicking the “*Create layer Object*” button will bring up a window asking for the layer object name. When you click ok the script will make a solid layer set up to be replaced by the project item specified from the column in your spreadsheet. So if you entered “graphic”, a new solid layer named “#graphic” will be created in the comp you are viewing; When you run the script, this layer will be replaced with the project item named in the “graphic” column of your spreadsheet.

While in a comp clicking the “*Create Import Object*” button will bring up a window asking for the import object name. When you click ok the script will make a solid layer set up to be replaced by the project item imported from the path specified in the column in your spreadsheet. So if you entered “audio”, a new solid layer named “>audio” will be created in the comp you are viewing; When you run the script, the script will import the file located at the path referenced in the “audio” column of your spreadsheet. It will then replace the solid with this newly imported file. To get the proper path for a file click the “Get File Path” button and navigate to the file you want to reference. A window will pop up with the path. You can copy this path and use it in your spreadsheet.

Instead of entering the full path for files to import, you can set a [default import directory](#) where all your files are, then you only need to put the filename in the spreadsheet.

You do not need to create target layers using the interface. Any text layer in a comp that starts with a ^ will be set up as a text target. Any layer in a comp that starts with a # will be a layer replace object. Any layer that starts with a > will be set up to be replaced by an imported file. Clicking “Run compsFromSpreadsheet” will run the script. It will go through all your comps looking for target layers. Then it will ask you to select your grid file. This is your spreadsheet saved as a tab-delimited text file. The script will go through this file line by line, making copies of the template comp referenced in the ‘COMP’ column and replace any target layer in that comp with data from the column of the spreadsheet it is connected to.

# Running the Script

When you click the “run compsFromSpreadsheet” button a window will pop up asking you to select your grid file. This is the tab-delimited text file that you created from your spreadsheet. Do not select your actual spreadsheet as the script will not be able to understand it.

Once you select the tab-delimited text file, the script will read the first line of it which needs to contain the names of all the columns. The field init window will pop up to allow you to sync the data in the text file to target layers in your comp.

## Naming Comps

The next step you need to tell the script how to name the newly created sequence, and where you want the rendered sequences to go. The default naming convention is line number, so the name of the very first Comp is “2” ( because line 1 holds the column names). and the next comp created is called “3” and so on until it reaches the last line of the spreadsheet.

Alternately you can have the script keep the name of the original template comp and append a number to the end. so the first time the script creates a comp from “TemplateA” it is named “TemplateA\_1”, next line that uses that comp will name the new comp “spotA\_2”

Lastly, you can choose to get the name from one of your spreadsheet columns. So if you choose “FROM SPREADSHEET | name” from the dropdown, the comp will get its name from the ‘name’ column of the spreadsheet.

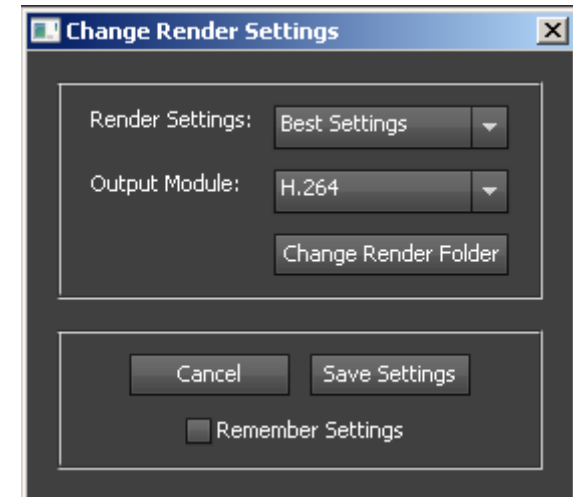


# Render Settings

By default, every comp created by the script will be placed in the render queue using the default settings for your render queue. It will default to saving the renders in the same directory as your project. If you don't want your new comps put in the render queue you can turn this off in the settings.

If you click the “Render Settings” button (on the “field init” window) you can select which render setting and output module to use for the script and where to put the rendered files. If you check the “Remember Settings” box these settings will be used every time you run compsFromSpreadsheet; if not checked, these settings will be used in the current run.

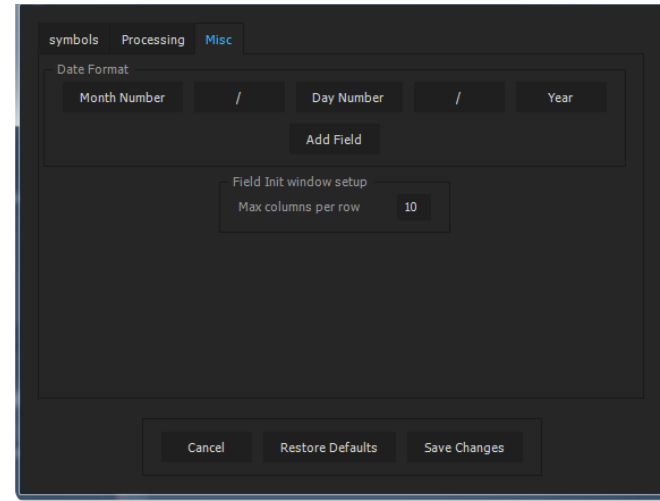
You can also set the output module and render setting for each item [via the spreadsheet](#)



## Settings

The settings button on the main UI panel allows you to change the default symbols the script uses to target layers in your Comps. You can also set the number of columns that appear in the field init window. And you can also change render settings from here, or uncheck the “add new comps to render queue” to bypass this altogether and just create comps.

The date format section (under the Misc tab) allows you to change the default date format. The default format is MM/DD/YYYY, you can change the order or have month names used instead of numerical



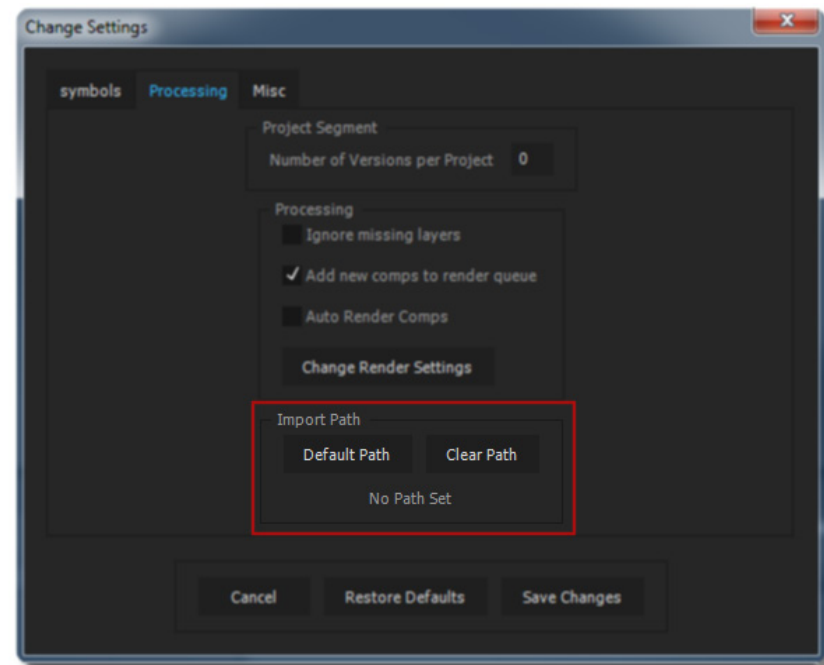
## Project Segment

If you are creating a lot of versions with your spreadsheet (200+), your system may slow considerably. The segment feature will break your run up into multiple projects. Setting the number to a value greater than zero will create a new project every time the number of versions reaches the number set. With the auto render button, the script will cycle through each project and render all comps.

When you run compsFromSpreadsheet in segment mode the system will ask where you want the segmented projects to be saved. The script will create a folder in this location named “Cfs\_segFiles” to hold all the After Effects projects to be created. If you set the “Number of versions per project” setting to 10, and your spreadsheet has 100 lines of data, the script will create 10 after effects projects, the first project will be named 1.aep, and hold the first 10 comps created from the spreadsheet.

# Default Import Directory

You can now choose a single directory from which to have CompsFromSpreadsheet import from. This way instead of having to put a long pathname in your spreadsheet you can set up a default location where all your files are, then you only need to put the filename in your spreadsheet.

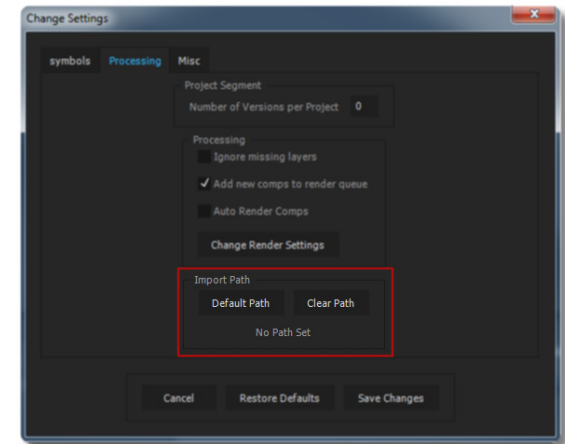


Old Import Data	
B	C
image 1	text
/c:/!Projects/zzmike/cfs/Rev1/archive/final/rev2/2.mov	Limited Time Offer
/c:/!Projects/zzmike/cfs/Rev1/archive/final/rev2/1.mov	Call Now
/c:/!Projects/zzmike/cfs/Rev1/archive/final/rev2/3.mov	Don't Wait

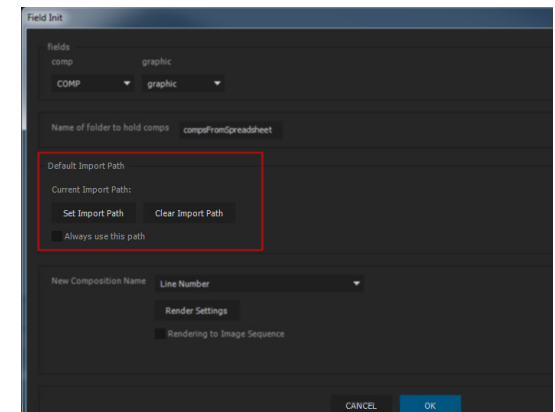
New Import Data	
B	C
image 1	text
2.mov	Limited Time Offer
1.mov	Call Now
3.mov	Don't Wait

In Previous versions of CompsFromSpreadsheet, when you wanted the script to import a file into your project, you had to put the full path to every file imported. Now if all your files are in the same folder you can tell CompsFromSpreadsheet to import all files from that folder.

There are two ways to set the import path for CompsFromSpreadsheet. If you want to set a default directory to be used every time you run CompsFromSpreadsheet, set the default directory from settings. Settings in CompsFromSpreadsheet are now tabbed, go to the *Processing* tab and click on the *default path* button and navigate to the folder you want to always import from. Click ok, now you only need to put the filename in your spreadsheet and the script will find it.



To set an import directory per run, there is a Default Import Path section in the Field Init Window. Click the *Set Import Path* button and navigate to the folder where your files are.

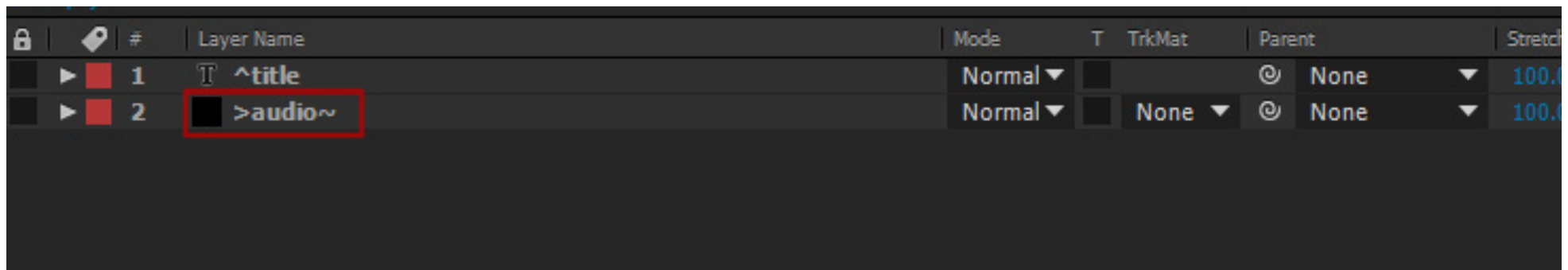


# Composition length can be adjusted based on layers length

The length of your composition can now be increased or decreased to match the length of a layer. Any layer that is set to receive either a layer from the project or an imported layer can be designated to adjust the length of the main comp, if the comp's length is increased all the layers in the comp will be lengthened accordingly.

To designate a layer as a length layer all you need to do is add a ~ after the name; so if you had a layer set to import audio files named `>audio` in your comp just add a trailing ~ to make it `>audio~` and when the script imports the audio file the length of the comp will be lengthened or shorted so that it ends exactly when the imported audio file ends.

This only works in the main sequence, if you put a length layer in a precomp the main comp's length won't be adjusted. If you have more than one length layer in your comp, only the last one will matter.



# Output Module and Render Settings can be set from the spreadsheet

You can now set the output module and render settings individually from the spreadsheet. To set the output module from the spreadsheet you need to have a column *outputModule* in your spreadsheet to hold the name of the output module to use for each render.

To set render settings you need to have a column *renderSettings* in your spreadsheet to hold the name of the render setting to use for each render.

The image shows a spreadsheet with columns A through E. Column A contains 'comp', B contains 'title', C contains 'outputModule', and D contains 'renderSettings'. Rows 2 and 3 show data for 'Comp 1' with titles 'movie 1' and 'movie 2'. The 'outputModule' column has values 'Lossless' and 'Lossless with Alpha'. The 'renderSettings' column has values 'Best Settings' and 'Best Settings'. Blue lines connect the spreadsheet cells to the 'Render Queue' window. Red lines connect the spreadsheet cells to the 'Current Render' window. The 'Render Queue' window shows two renders: 'Comp 1' (render 1) and 'Comp 1' (render 2). The 'Current Render' window shows the settings for the selected render, including 'Render Settings' and 'Output Module'.

	A	B	C	D	E
1	comp	title	outputModule	renderSettings	
2	Comp 1	movie 1	Lossless	Best Settings	
3	Comp 1	movie 2	Lossless with Alpha	Best Settings	
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					

Render Queue

Comp 1

Current Render

Render	#	Comp Name	Status	Started	Render Time
1	2	Comp 1	Queued	-	-
2	3	Comp 1	Queued	-	-

Render Settings: Best Settings

Output Module: Lossless

Log: Errors Only

Output To: 2.avi

Render Settings: Best Settings

Output Module: Lossless with Alpha

Log: Errors Only

Output To: 3.avi