



Making MOGRTs

Creating Motion Graphics templates
in Adobe After Effects

Jarle Leirpoll

Contents

2	Contents
4	What is a Motion Graphics template?
5	Why use After Effects to create MOGRTs
5	The Adobe MOGRT ecosystem
6	MOGRT limitations
6	Special cases where After Effects needs to be installed
8	A brief history of MOGRTs
10	The Essential Graphics Panel
10	Let's create a simple MOGRT
12	Share the MOGRT
14	Using the MOGRT in Premiere Pro
16	MOGRT settings in Effect Controls panel
17	Sort by Recent or by Title
17	Add more folder paths
18	List view in Essential Graphics Panel
19	Different ways to add properties to the Essential Graphics panel
19	Create a keyboard shortcut for Add Property to Essential Graphics
20	Supported types of property controls
22	Add groups and comments in the Essential Graphics panel
24	Using expressions to make user-friendly MOGRTs
24	What can expressions do?
24	Some features in After Effects that use expressions
26	How to add an expression to a property
26	Enable or disable all expressions on selected layers
27	Let After Effects write the code for you
29	Copy and paste expressions from documents and websites
32	Expression language fundamentals
32	Case sensitivity
32	Expressions ignore spaces and line breaks
32	Add comments to your code
33	Use existing values as a starting point for expressions
34	Punctuation in expressions
34	What's a variable?
35	What's an array?
36	What's a function?
36	Useful keyboard shortcuts for expressions
37	Be careful when using expressions with Master Properties
38	Expression Controls—your best friends
39	The Slider Control
41	Combine Separate Dimensions and Slider Controls
41	Color Control
44	Checkbox Control
45	Use a Controls layer
46	Most useful expressions for MOGRT design
46	The linear and ease expressions
49	if/else expressions
53	sourceRectAtTime
57	Use toComp to fix parenting problems
58	Saving your expressions for easy access
59	Bounce & Overshoot and Inertial Bounce
59	Custom ease expressions
60	parseFloat
61	Rounding numbers
62	Split
65	Change the timing of keyframes with expressions
66	Expression error banner

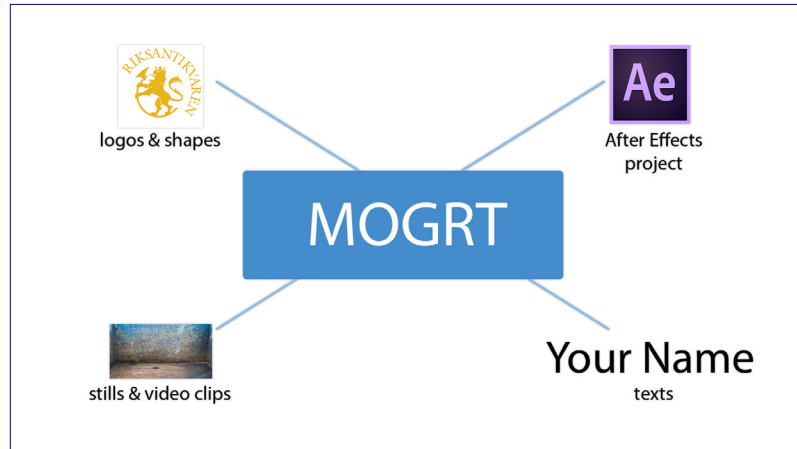
67	Making MOGRTs user-friendly
67	Decide if it's okay if After Effects must be installed
69	Use fonts that are available to all Adobe Creative Cloud users
69	Use sliders and rounding to give the user some options
70	Control several properties with one slider or checkbox
70	Give the user a Duration slider in earlier versions
70	Allow for multiple lines of text
72	Allow for different text designs
76	Nick Howell's timing guide preview
77	Responsive Design — Time
79	Why still use a duration slider?
79	Foolproofing the MOGRT
82	Data-driven MOGRTs
86	Using the data-driven MOGRT in Premiere Pro
88	MOGRT design tips
88	Use Adobe Color
89	The Adobe Capture app
90	Use Collapse Transformations
90	Make the MOGRT resolution independent
91	Center anchor point in new shape layers
91	Place anchor points where you need them
92	Use one custom animation in many layers
93	Get a visual representation of the animation
94	Reuse code with index
95	Use Master Properties
98	Use standard procedures, work faster
98	Add audio to your MOGRTs
100	Making MOGRTs for your customers and colleagues
100	Their design manual is the bible
101	Choice of approved fonts
101	Make MOGRTs with built-in safety nets
101	Beware of openers
102	Get access to their library
103	Optimizing MOGRT performance
103	Keep the layer count down
106	Reduce the render time with pre-renders
107	Exporting and distributing MOGRTs
107	Use nesting to avoid problems when you export MOGRTs
107	Use nesting to fix problems with unrelated properties
108	Universalizing the MOGRT
109	Exporting MOGRTs for use on your own system
109	Sending MOGRT files to customers and colleagues
110	Selling MOGRTs
111	Making changes in existing MOGRTs
112	What happens when you install a MOGRT?
116	Replace the MOGRT with a new version
117	How to solve font syncing issues
117	That's it!
118	Appendix: Online Resources for MOGRT creators
118	Tutorials on Essential Graphics panel
118	Tutorials on expressions
118	After Effects Expressions forums
119	JavaScript resources
119	Useful scripts and extensions for MOGRT creators
121	About the author
121	Thanks
122	Want to sell your MOGRTs?
122	Become a contributor to Adobe Stock.

What is a Motion Graphics template?

A Motion Graphics template (MOGRT) is a self-contained, animated composition that can be used in Adobe After Effects, Adobe Premiere Pro, and Adobe Premiere Rush. MOGRTs can include any combination of text, graphics, still images such as logos, and even audio and video files. Each instance of a MOGRT can be individually customized by the user.

The authoring of MOGRTs is where the full power of this workflow is revealed. Not only can Mograph creators determine which parameters are editable by the user, but they can also establish a workflow for the user by organizing those parameters. The result is a flexible and powerful creative tool, combining full design freedom, efficiency, and visual consistency across apps, devices, and users.

Although it's just a single file, a MOGRT includes an After Effects project, plus any text, still images, vector images, and video and audio files the designer used when creating it.



MOGRTs are self-contained files, so the editor can never miss a logo or use the wrong version of it. Fonts are not included in the MOGRT, but Premiere Pro and Premiere Rush can automatically install any missing fonts.

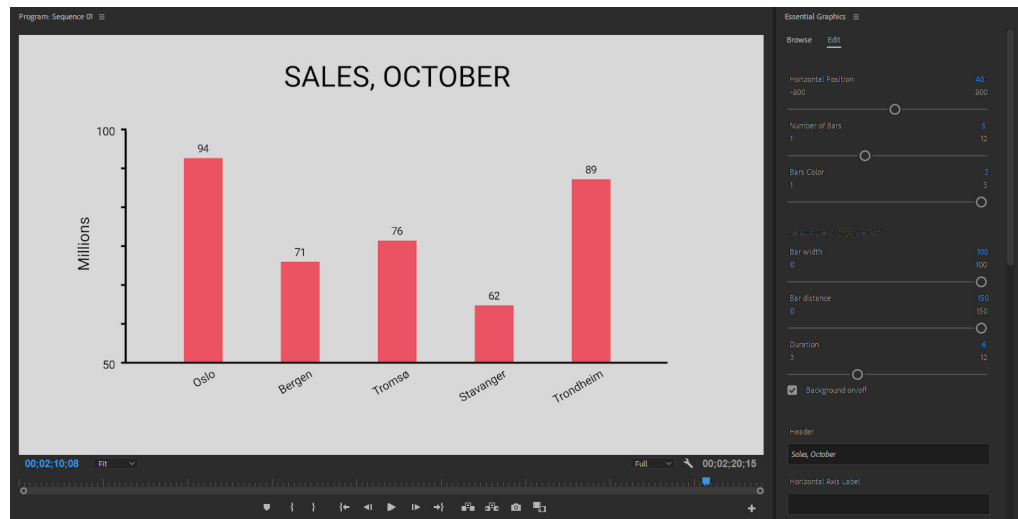
You can control settings in MOGRT clips from the Essential Graphics panel in Premiere Pro. Think of this panel as an easy-to-use remote control for your templates. The following figure shows a very simple MOGRT where the user can only change Name, Title, and Duration in the Essential Graphics panel.

Control parameters for brand consistency. The limited flexibility in this template for the Norwegian Public Roads Administration was a deliberate attempt to make sure that editors couldn't use other colors, fonts, sizes and so on.



The next figure shows a template that's a lot more flexible, allowing the user to change almost 40 settings in the Essential Graphics panel. As you can see from these examples, the designer has a lot of control, while using the template in Premiere Pro is still super easy.

This template lets the user control many settings, like the number of bars, bar width, distance between bars, and so forth. Still, I decided to give the user only three colors to choose from, and the template also has a fixed font size.



Graphic clips made with MOGRTs created in Premiere Pro can be tweaked and changed indefinitely by the editor in Premiere Pro.

Why use After Effects to create MOGRTs

MOGRTs can be created in both Premiere Pro and After Effects, so why should you make them in After Effects? The obvious answer is that since After Effects has much better motion graphics tools, MOGRTs created in After Effects can be more advanced than the ones created in Premiere Pro. That's great, but there's another difference that's equally important for MOGRT creators and users—what the editor can change when using the MOGRT.

Say your company's design manual allows only three colors and two fonts to be used. If you build a MOGRT in Premiere Pro, there's no way to restrict the colors and fonts. Yes, you can build a template that has the right colors and fonts by default, but if the editor wants to change them, the designer can't prevent this from happening.

This means that MOGRTs created in After Effects give the designer much more control over the result. If you build the MOGRT in After Effects, you can restrict font choices, color choices, and almost everything else. This enables the designer to maintain a common look on all motion graphics in the company's video stories.

The Adobe MOGRT ecosystem

Although MOGRTs can also be created in Premiere Pro, this book will only cover how to create MOGRTs in After Effects. Here's an overview of the whole workflow. We'll use a lower third as an example.

1. The designer builds a nice lower third in After Effects, including animation, logo, text and so on. Using the Essential Graphics panel in After Effects, the designer can choose which properties can be changed in Premiere Pro.
2. The designer exports the lower third as a Motion Graphics template, aka a MOGRT, which contains everything except the font. It's actually just a ZIP archive with a different name (.mogrt).
This can be saved to a local folder, distributed to a workgroup in a library, or uploaded to Adobe Stock or other sites that sell MOGRTs.
3. If the MOGRT is saved to a library, everyone with access to that library will immediately have the MOGRT available in the Essential Graphics panel in Premiere Pro. If the user buys it from Adobe Stock, it will also be available immediately in the Essential Graphics panel.

If the MOGRT is sent to the Premiere Pro editor as a file or an email attachment, there's one more step, but it's very simple. The editor installs it by clicking the + button in the Essential Graphics panel.

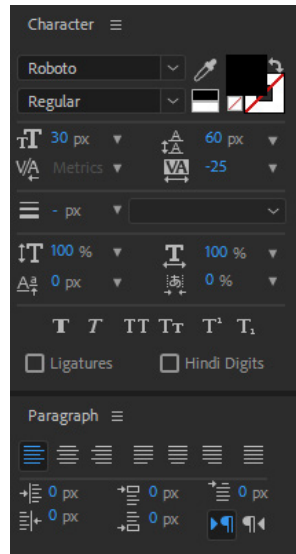
To use the template, the editor simply drags it from the Essential Graphics panel to the timeline and then uses the same panel to change text and parameters. If the font used in the template is not installed on the editor's system, a dialog box will pop up, allowing the editor to automatically install the missing font.

MOGRT limitations

There are two important limitations in MOGRTs that you should be aware of before you start designing them.

Not all text properties are supported

Only a few of these properties are supported in MOGRTs.



You can control the font family, font style, size, and faux styles of a text layer in a MOGRT. But the rest of the settings in the Character and Paragraph panels are unreachable for MOGRTs. This means that you can't make these parameters directly available in Premiere Pro.

There are some workarounds, though. If you need a MOGRT where the user can choose between three text alignment settings, you can use three text layers with different settings and turn them on and off (setting Opacity to 0 or 100) when the user drags a slider or selects a checkbox. This does impact the MOGRT's performance a bit.

To change the text color, you can add an animator to the text layer and let it change the fill color.

Fields (interlacing) are not supported

If you're working in Broadcast and need to output interlaced video, this may be a problem. To get real interlaced video from a MOGRT, use 1080 50p comps for 50i and 1080 60p for 60i, and put the MOGRT in an interlaced timeline in Premiere Pro. Premiere Pro will take care of the interlacing.

You may want to set the Motion Blur to 360 degrees in After Effects when you do this to match the motion blur of video and other elements shot at 50i or 60i.

Special cases where After Effects needs to be installed

For most designers, the goal will be to make their templates work in Premiere Pro without requiring After Effects to be installed. The October 2017 release and newer versions of Premiere Pro can render most MOGRTs without help from After Effects; however, some effects and features must be avoided:

After Effects must be installed on the end user's system if you use these effects and features.

CINEMA 4D and Ray-traced 3D renderers are not supported. Use only the Classic 3D renderer to make sure the MOGRT works without After Effects being installed.

These effects are not supported: Camera-Shake Deblur, the immersive video (VR) category of effects, Lumetri Color, Maxon CINEWARE, and Puppet. If you're using the older 15.1.2 version, add 3D Camera Tracker, Apply Color LUT, and Warp Stabilizer to this list.

Dynamic Link footage is not supported. This means you can't use Premiere Pro sequences or Character Animator scenes in an After Effects composition in a MOGRT.

C4D, FLV, and SWF format footage is not supported.

Third-party plug-ins are not supported.

You can still use these effects in a MOGRT for Premiere Pro, but After Effects must be installed on the computer in addition to Premiere Pro.

If you plan to sell your MOGRTs through Adobe Stock, check for any special submission requirements to Stock at <https://spark.adobe.com/page/6sORsTLKTQVzA/> and on [Adobe Help](#).

Get a warning when you export a MOGRT with limitations

The *Export As Motion Graphics Template* dialog box in After Effects includes two options to help you identify dependencies for the MOGRT:

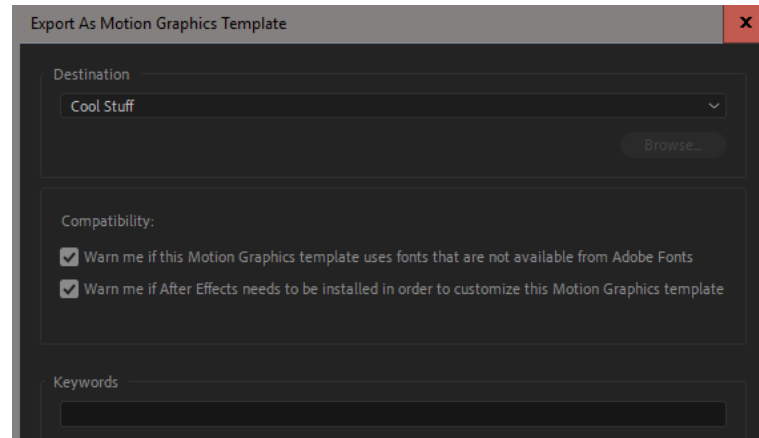
Warn me if After Effects needs to be installed in order to customize this Motion Graphics template

When enabled, this option will warn you if the exported MOGRT does not meet the requirements for rendering in Premiere Pro without After Effects installed.

Warn me if this Motion Graphics template uses fonts that are not available on Adobe Fonts

When enabled, this option will warn you if the exported MOGRT uses fonts that were not synced from Adobe Fonts.

You can choose to be warned if the MOGRT requires fonts not available on Adobe Fonts, or when After Effects needs to be installed.



These options only display warnings; they do not modify the content of the MOGRT.

If you need to make changes to the project to meet the requirements, cancel the export, make the changes, and then export the MOGRT again.

Note: You will not learn After Effects by reading this e-book. If you want to learn the basics, [watch an online course](#), or [find a trainer](#) who can teach you about expressions in After Effects.

A brief history of MOGRTs

It all started with Live Text templates in June 2014—a feature that was the answer to what many broadcasters had asked for. They wanted editors to be able to change the text in lower thirds and titles without the help of the After Effects artists who designed the templates, and without the need to open After Effects.

Live Text was launched in the 2014 version of Premiere Pro and After Effects.



When a comp in After Effects was tagged as a template, all the text layers that were not locked could be changed in the Effect Controls panel in Premiere Pro via Dynamic Link. The Live Text templates feature was a game changer, but it was never meant to be used for advanced workflows. Only text layers could be changed from Premiere Pro.

Soon, some After Effects artists, including me, started “hacking” Live Text templates using expressions. Layer and effect properties other than source text could be linked to invisible text layers, which could then be controlled via the Live Text template in Premiere Pro. With the use of the `parseFloat()` expression that extracts numbers from text, we could make templates where the user could enter “80” in a text field in Premiere Pro to set the Opacity of a layer to 80 and so on.

I used this method to make lower thirds where the user could enter a number for the padding around the text in a rectangle and enter the value of each bar in a bar chart. I also made some templates where users could switch logos by typing numbers 1, 2, 3, and 4, and change colors by entering text, like “green,” “blue,” “red,” and so on.

In 2015, I did a presentation on “Automated Motion Graphics Using Premiere Pro and After Effects” at the Adobe Video World conference in San Jose, where I showed these and many other examples and how to build them.

My presentation at Adobe Video World got me involved in the development of MOGRTs.

Two members of the After Effects team saw my presentation and asked if we could talk afterwards. One of them said, “*We have created a monster!*” And they had.

Of course, they already knew that their simple text replacement tool could be used for so much more than it was designed for, and they had plans to develop it further.

They still wanted to know what I, as a motion graphics artist and editor, wanted a motion graphics template to do.

I said, “*Give me sliders, checkboxes and color controls, and let me change fonts.*”

Adobe then asked me and two other After Effects artists to work with the After Effects team in the development of this new workflow, and the team worked fast.



In November 2016, we got new Live Text template improvements. Here's the text from the marketing people: *"A new file format lets you share Live Text templates between Premiere Pro and After Effects without the need for a separate After Effects license."* Capsules (.aecap) had arrived! After Effects was needed on the system for capsules to work, but it could be a free, expired trial.

Then, in the April 2017 release, MOGRTs were born! The file extension was changed to .mogrt, and the name was changed. Apparently, the word *capsule* does not work well in all languages. Premiere Pro and After Effects got new Essential Graphics panels in this release, and we could share MOGRTs via libraries. MOGRTs were still dependent on After Effects on the system.

Every update of After Effects and Premiere Pro since then has improved the MOGRT workflow. In the October 2017 update, we no longer needed After Effects on the system to use MOGRTs. The After Effects engine was now part of Premiere Pro. And Adobe Stock started selling MOGRTs.

The April 2018 release introduced a new way to browse, search, and preview MOGRTs from local folders, libraries, and even Adobe Stock, right in the Essential Graphics panel in Premiere Pro. We got support for 2D position, rotation, and scale and many other parameters without the need for extra slider controls. We could also edit text layers in MOGRTs by clicking on them in the Program Monitor, and the Smart Replace feature was introduced, making it easier to update existing graphics.

The October 2018 release added drop zones for spreadsheets, font controls and adaptive time-stretching with protected regions and many UI refinements and other improvements.

We can now do everything I asked for back in 2015 and more. I can't wait to see what they have up their sleeve for future versions.

Adobe Stock offers thousands of MOGRTs, available directly from a panel in Premiere Pro.

If you want to sell your
MOGRTs on Adobe Stock,
go to this web page for info:

<https://adobe.ly/contributemogrts>



The Essential Graphics Panel

I've already mentioned the Essential Graphics panel a few times. This panel is where all the magic happens when you create MOGRTs, so let's take a close look at it. The basic rule for using it is very simple:

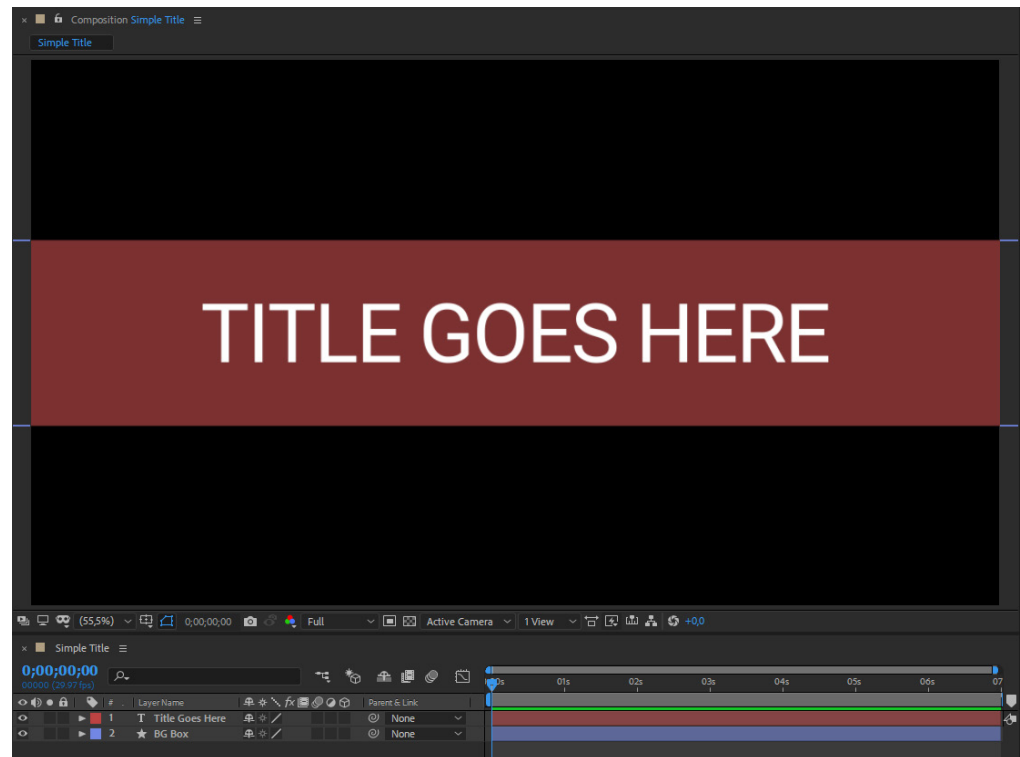
(Only) the properties you add to the Essential Graphics panel in After Effects will be available for the user in the Essential Graphics panel in Premiere Pro.

This means that *you* have total control over which parameters the end user can change. This is what makes the MOGRTs made in After Effects better suited for most purposes than the ones created in Premiere Pro. There's no better way to understand the panel than to build a MOGRT.

Let's create a simple MOGRT

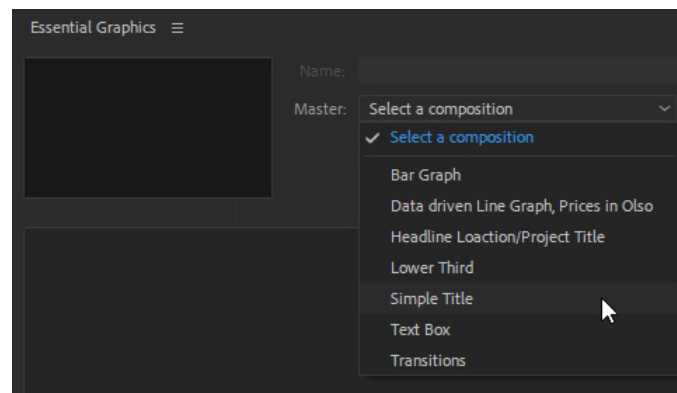
Here's a very basic composition with a shape layer that works as background box, plus a text layer. We'll build a title MOGRT from this comp, named "Simple Title," step by step.

This simple composition will be our starting point for making our first MOGRT.



Open the Essential Graphics panel by choosing **Window > Essential Graphics**. Select the Simple Title comp as the Master in the Select a composition drop-down menu and enter a descriptive name in the *Name* field.

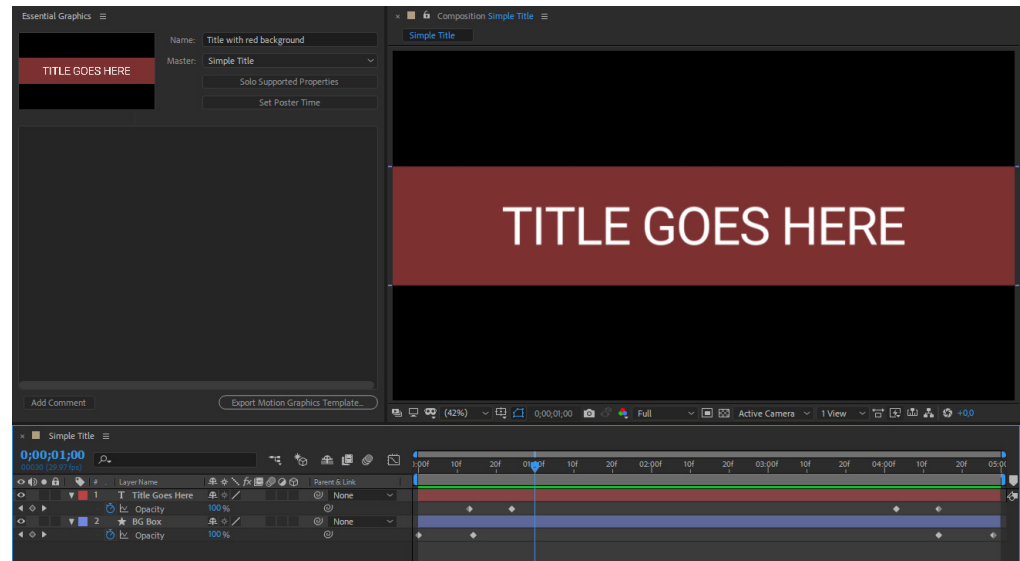
Select the **comp** in the Master drop-down menu.



The poster frame is not rendered when you click the Set Poster Time button; it is rendered when you export the MOGRT.

Clicking the button only sets the time of the composition from which the poster frame is rendered.

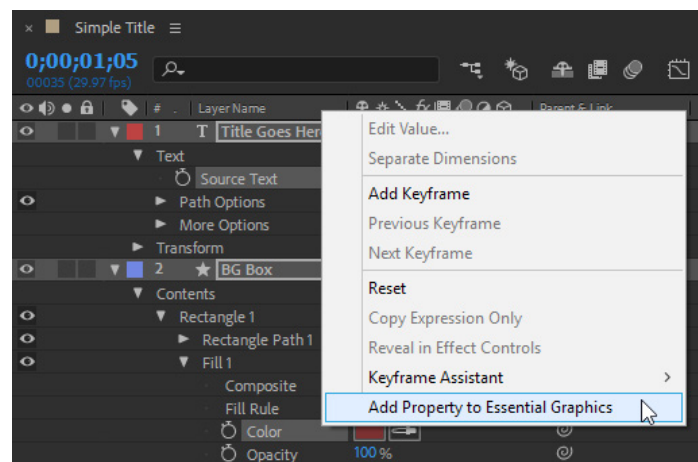
Park the playhead at a time where the Program Monitor shows a representative image of the MOGRT. In this case, the two layers have Opacity keyframes that fade the layers in. At about 1 second, both layers are visible, and this makes a good thumbnail for the MOGRT. So, park the playhead at 1 second and click the *Set Poster Time* button in the Essential Graphics panel.



Park the playhead at 1 second because both layers are visible there, allowing a good thumbnail image of the MOGRT.

Now it's time to decide what the Premiere Pro editor should be allowed to change in the template. In this case, it's a good idea to let the editor change the text and the color of the background. This means that we need to add these to the Essential Graphics panel.

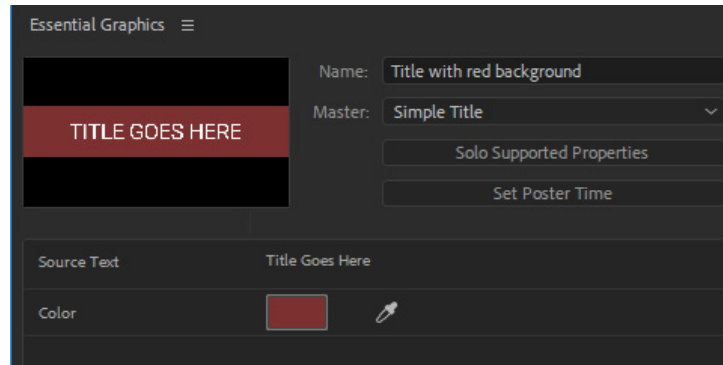
Select the **Source Text** and the **Fill 1 Color** parameters and right-click to add them to the Essential Graphics panel.



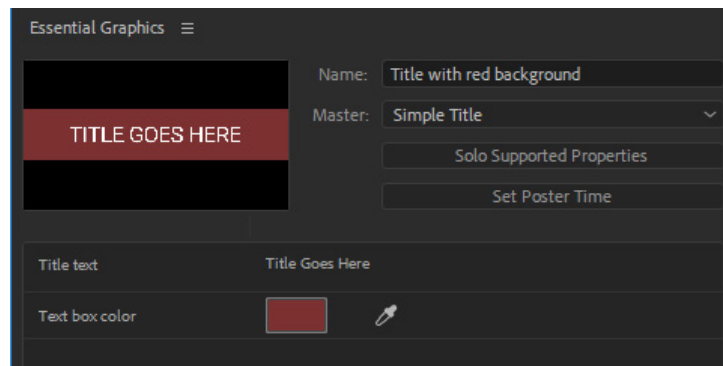
Select the *Source Text* from the text layer and the *Fill 1 Color* from the shape layer, and then right-click to show the *Add Property to Essential Graphics* option.

The two parameters we added to the Essential Graphics panel still have the names they had in the timeline. But *Source Text* and *Color* are not the best labels, so let's change them by selecting the names and typing new ones.

The original names of the two parameters are not very descriptive.



The new names are more descriptive and easier to understand.



Share the MOGRT

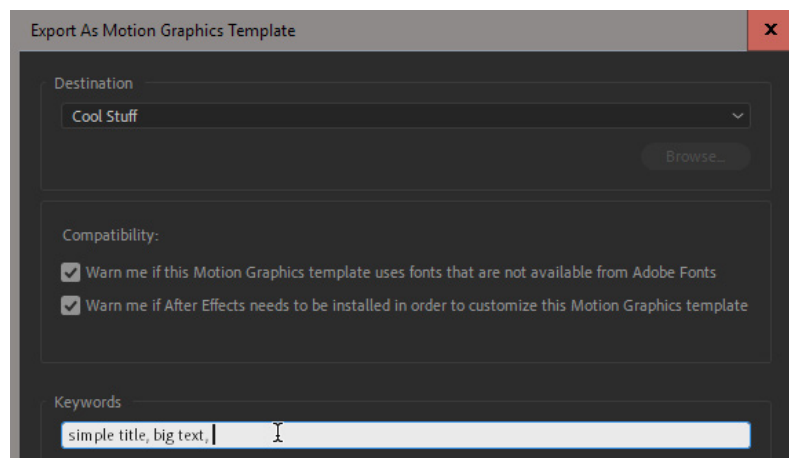
Now we're ready to export the MOGRT. Click the *Export Motion Graphics Template* button at the bottom of the panel.

Export the MOGRT by clicking the button.

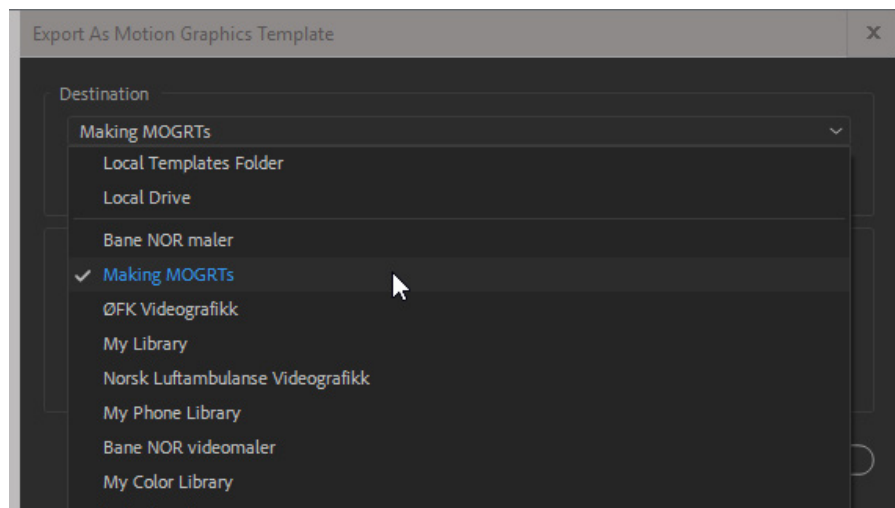


You will be asked to save the project, and if you want to proceed, you'll have to agree. Then a dialog box pops up, asking where you want to export the MOGRT. This is also where you can add keywords that become searchable in Premiere Pro.

You must select where to export the MOGRT. Adding keywords will help the user find the MOGRT.



Select a library, a folder on your local drive, or the local templates folder.



When you click the Destination drop-down menu, you'll see all the different places you can export to. If you select the *Local Templates Folder* option, the template will be available only to you, and only on the system you're working on, in the Essential Graphics panel in Premiere Pro. This is okay if you're a one-man band and don't need to share the template with anyone.

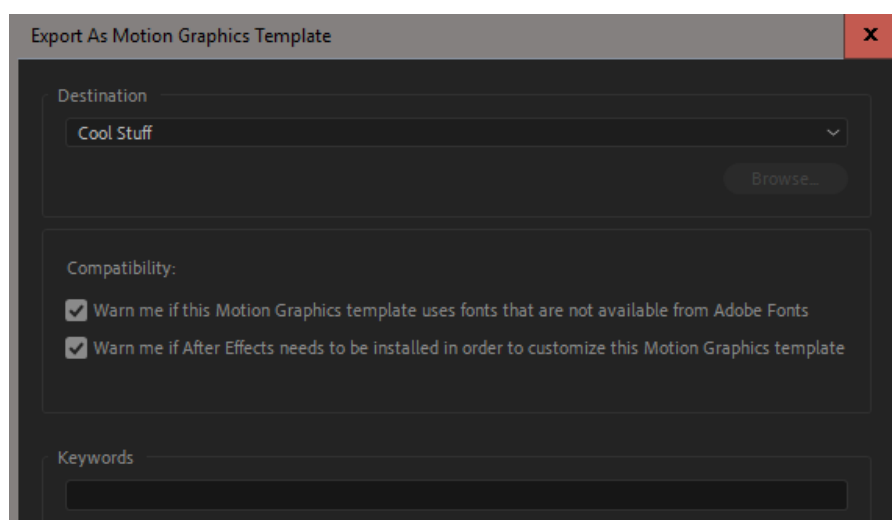
If you select the *Local Drive* option, you'll get a *.mogrt* file that you can send to anyone you want to share it with. You'll have to trust that they actually do install it (and delete the old one, if this is a new version), which we all know is just wishful thinking.

The other options in the list are libraries. Libraries can be shared with everyone who needs the templates, and they will have the template immediately available in the Essential Graphics panel in Premiere Pro. You can delete older versions if necessary to be sure that everyone always uses the latest version.

Libraries are great. They are also secure, but some IT departments seem to think they're not because they live in the "cloud." Often, a conversation with Adobe engineers about encryption and security measures will convince IT people that libraries are okay to use.

I'll export this MOGRT to my Making MOGRTs library that I made for this book project. The following figure shows the *Export As Motion Graphics Template* dialog box.

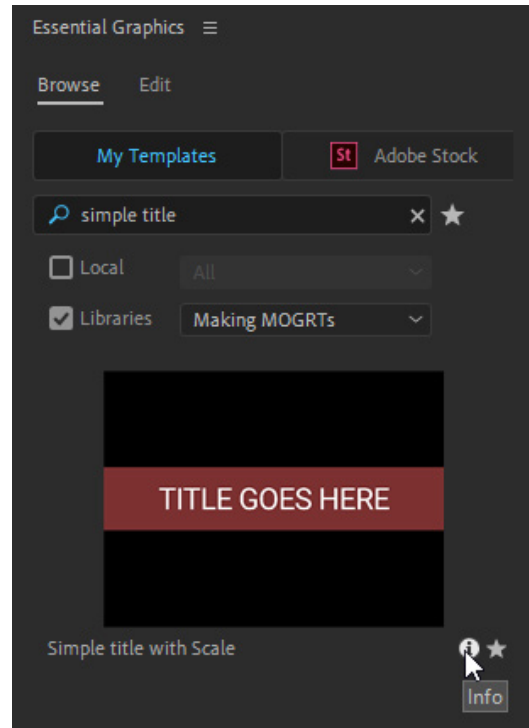
I've chosen to save to a library named "Making MOGRTs" and to get a warning if the template uses fonts that are not available in Adobe Fonts and if the template needs After Effects to be installed on the system.



Using the MOGRT in Premiere Pro

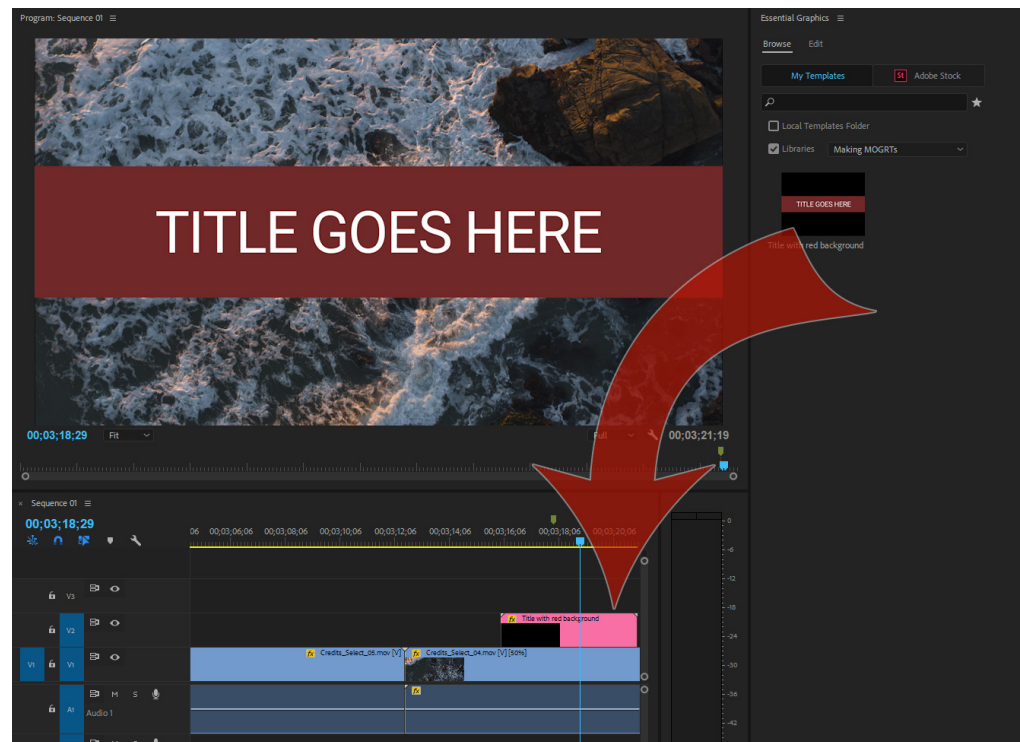
In Premiere Pro, the new MOGRT is now available in the Essential Graphics panel, not only for me, but for anyone who subscribes to this library. This is a great way to automatically distribute your templates to everyone who needs them.

The template is immediately available in the same library in the Essential Graphics panel in Premiere Pro.



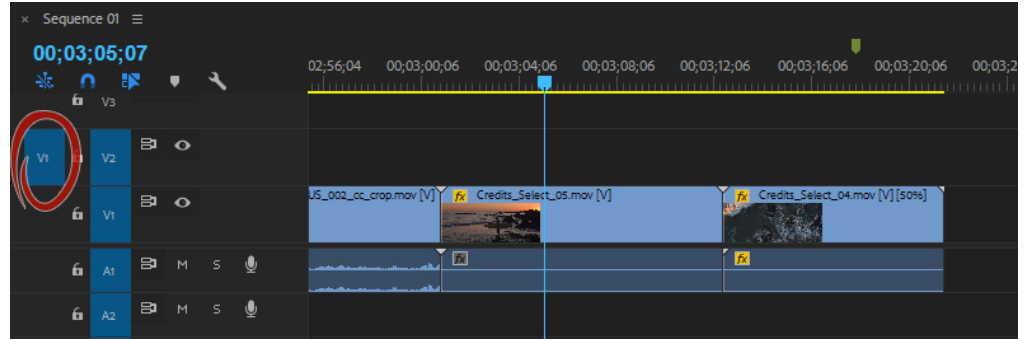
To use the template, editors can just drag it from the Essential Graphics panel to the timeline. When they click on the clip in the timeline to select it, the Essential Graphics panel switches from the *Browse* tab to the *Edit* tab, where they can change the text and the box color.

Drag the template from the Essential Graphics panel to the timeline.

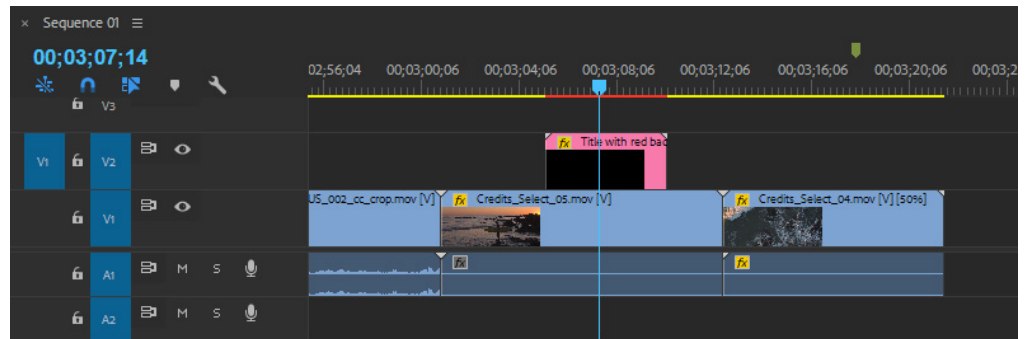


A better way to add the MOGRT to the timeline is to select the MOGRT in the Essential Graphics panel and set the V1 source patching to the track you want the MOGRT on, as shown in the following figure. Then press the Period key (.), and the MOGRT is added to the chosen track.

To avoid drag-n-drop you can create keyboard shortcuts for source patching and use the existing keyboard shortcut for overwrite, which is period (.). Here I've chosen to add it to video track 2.

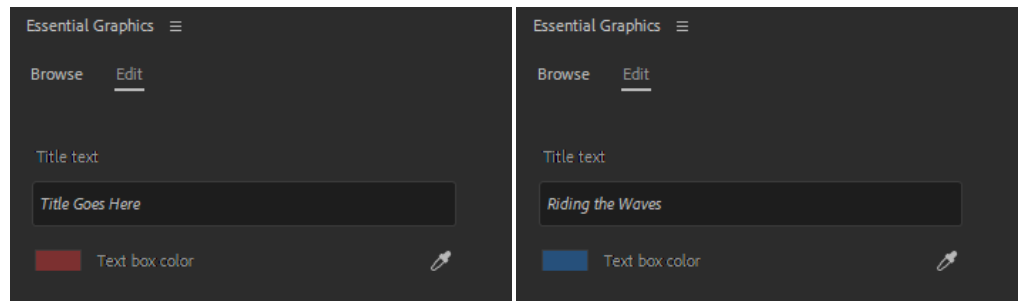


When you press the Period key, the MOGRT is added to the chosen track.



You can change the text by typing in the Title text field. You can also change the box color by clicking the colored rectangle or using the eye dropper to select a color.

Change the text by typing in the Title text field, and select a new color by clicking the colored rectangle, or use the eye dropper to pick a color from the background clip.



The title after the user has changed the text and the box color.



Another way to change the text in a MOGRT is to click on the text in the Program Monitor with the Type tool. A small text box appears, where you can type your new text.

You can click directly on the text in a MOGRT with the Type tool and enter the new text.

Only direct text can be edited directly in the Program Monitor in Premiere Pro. Text that is linked to other sources via expressions in After Effects must be edited in the Essential Graphics panel.



When editors want to duplicate the existing MOGRT in the timeline, they can Alt-click the MOGRT clip and drag to the right or left to create a copy, or use copy and paste (**Ctrl+C** > **Ctrl+V** on Windows, **Cmd+C** > **Cmd+V** on macOS).

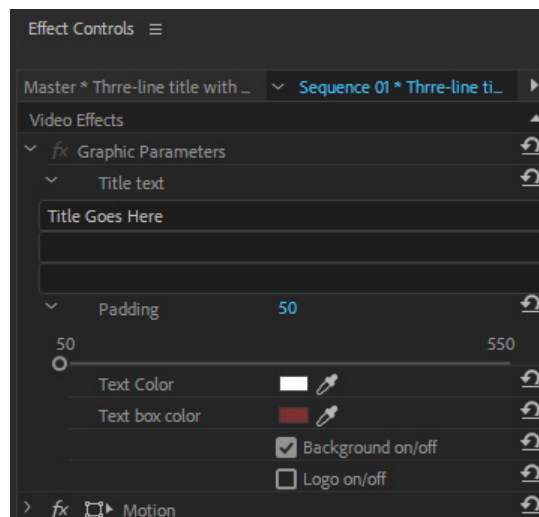
If they drag in a new copy from the Essential Graphics panel, it will have the original settings, not the text and color of the one that's already in the timeline. More often than not, this is *not* what they want.

MOGRT settings in Effect Controls panel

The main panel for controlling your MOGRTs in Premiere Pro is the Essential Graphics panel. But if it's inconvenient to have that panel open, you will find the same settings in the Effect Controls panel.

There are even a few things you can do in the Effect Controls panel that you can't do in the Essential Graphics panel, like resetting text, color controls, and checkboxes to their standard values. You can double-click sliders in the Essential Graphics panel to reset them, but there's no way to reset color controls and checkboxes. Every parameter in the Effect Controls panel has a reset button.

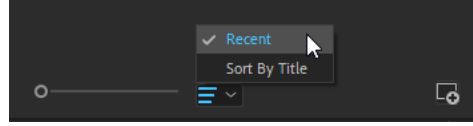
All Graphic parameters in the Effect Controls panel in Premiere Pro have reset buttons.



Sort by Recent or by Title

A button bar at the bottom of Essential Graphics panel lets you select a sort option: Recent or Sort By Title. This means that you can sort by Recently Used or by A-Z. Reverse sort order or any other sorting options are not possible.

The Essential Graphics panel has some rudimentary sorting options.



So, what makes a clip count as Recently Used?

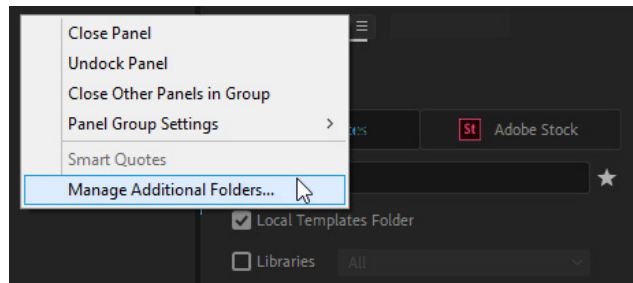
- A MOGRT exported from Premiere Pro to Local Templates Folder or CC Libraries
- A MOGRT exported from After Effects to Local Templates Folder
- A MOGRT added to the timeline by drag and drop in Premiere Pro
- A MOGRT downloaded/licensed from Adobe Stock in Premiere Pro

Add more folder paths

By default, you will see only the default template location in the UI, named *Local Templates Folder*, with a checkbox. In Premiere Pro 13.0 (October 2018 version), you can add more local folders. These can be used in addition to the default Local Templates Folder in the drop-down menu. You no longer need to save templates to the rather strangely located Local Templates Folder, and I find this very helpful.

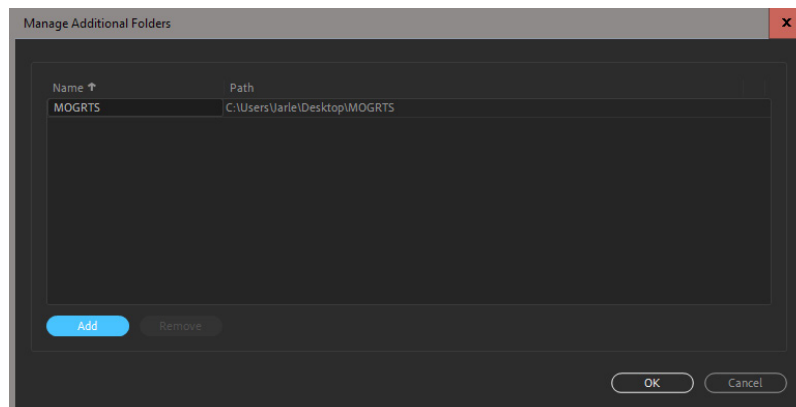
From the Essential Graphics panel menu, you can choose *Manage Additional Folders* to add paths.

You can select which additional folders the panel should show MOGRTs from.

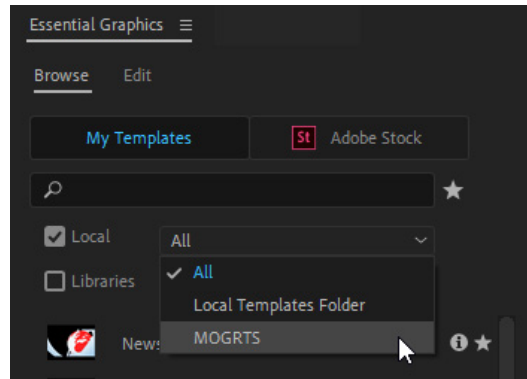


From the Manage Additional Folders dialog box, you can add (and delete) multiple folder paths, give the folder paths custom names, and so on. Clicking OK will add all custom paths to the Browse panel as a drop-down menu next to the Local filter option.

Click the Add button to add a folder. Select a folder and click *Remove* to remove it from the list.



The added folders will be available in the drop-down menu in the Browse panel.



If a custom path is not reachable (maybe because it was renamed or deleted, or your network drive is offline), it will be disabled in the drop-down list.

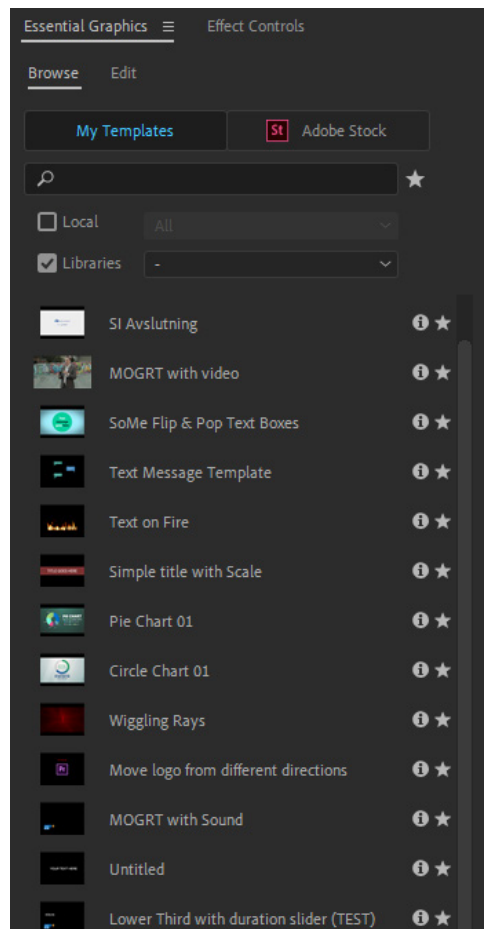
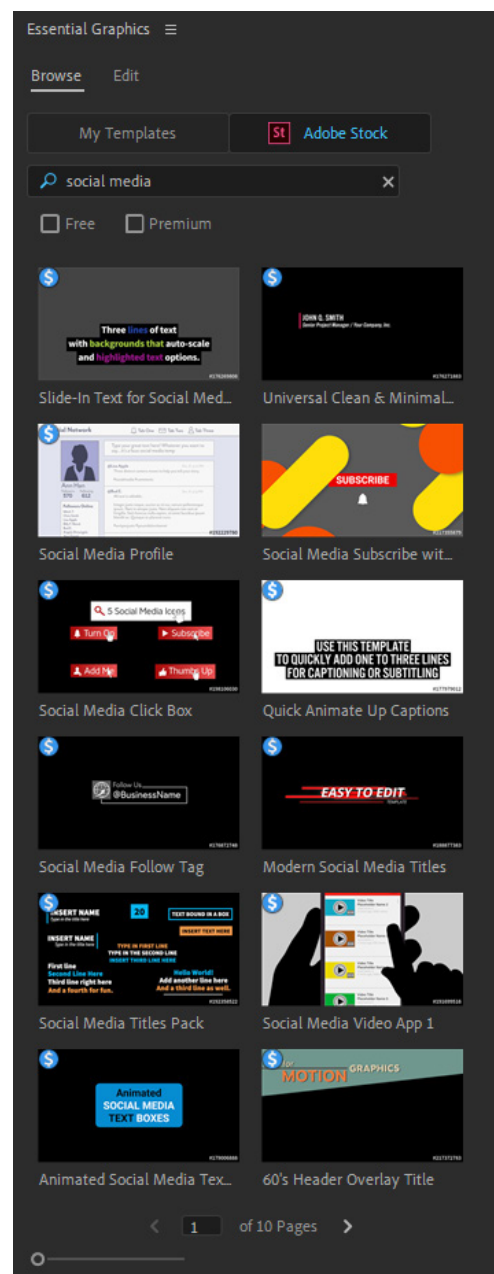
List view in Essential Graphics Panel

If you drag the Size slider at the bottom of the Essential Graphics panel in Premiere Pro to the far left, you enter list view. This view shows small thumbnails and full names of the MOGRTs and can show more MOGRTs than the grid view. Since I tend to have a lot of MOGRTs, this helps me browse them faster.

Adobe Stock search results are shown as pages.

List view is great when you have lots of MOGRTs.

Search results from Adobe Stock will show as pages.



There are several ways to add properties to the Essential Graphics panel:

- Option 1 is what we did in our first example. You can select and add multiple properties in one step to save mouse button clicks. Keeping the amount of clicks down means you'll work faster.

The screenshot shows the Adobe After Effects interface. At the top, the title bar reads 'Simple Title'. The timecode is '00:00:01:05' at '00035 (29.97 fps)'. The main composition window displays a text layer named 'Title Goes Here' at the 1-second mark. The left sidebar shows the 'Text' layer selected, with its 'Contents' panel expanded, showing 'Rectangle 1', 'Rectangle Path 1', and 'Fill 1'. The 'Fill 1' property is currently set to 'Composite'. The 'Keyframe Assistant' menu is open, showing options: 'Edit Value...', 'Separate Dimensions', 'Add Keyframe', 'Previous Keyframe', 'Next Keyframe', 'Reset', 'Copy Expression Only', 'Reveal in Effect Controls', and 'Add Property to Essential Graphics'. The 'Add Property to Essential Graphics' option is highlighted with a blue border and a mouse cursor. The 'Reset' option is also visible. The 'Add Property to Essential Graphics' option is highlighted with a blue border and a mouse cursor.

Option 4, the keyboard shortcut, is fast, but you'll have to create this keyboard shortcut yourself.

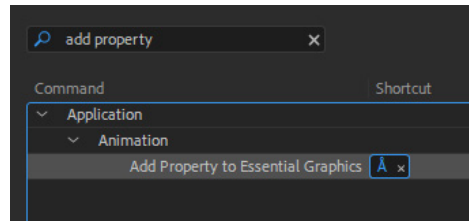
To add a keyboard shortcut for *Add Property to Essential Graphics*, choose **Edit > Keyboard Shortcuts**. This brings up the Keyboard Shortcuts panel.

The screenshot shows the 'Keyboard Layout Preset' dialog in Blender 2.80. The 'Default' preset is selected. The grid displays various shortcuts for different keys. The 'Command' column is highlighted in blue. The 'Shortcuts' column shows the assigned shortcuts for each key. The 'Modifiers' column shows the modifiers for each shortcut. The 'Command' column shows the command for each shortcut.

Key	Shortcuts	Modifiers	Command
F1	Shortcuts de l'Info.		
F2	Shortcuts de l'Info.		
F3	Shortcuts de l'Info.		
F4	Shortcuts de l'Info.		
F5	Shortcuts de l'Info.		
F6	Shortcuts de l'Info.		
F7	Shortcuts de l'Info.		
F8	Shortcuts de l'Info.		
F9	Shortcuts de l'Info.		
F10	Shortcuts de l'Info.		
F11	Shortcuts de l'Info.		
F12	Shortcuts de l'Info.		
Ctrl	Shortcuts de l'Info.		
Alt	Shortcuts de l'Info.		
Shift	Shortcuts de l'Info.		
Space	Shortcuts de l'Info.		
Tab	Shortcuts de l'Info.		
Q	Shortcuts de l'Info.		
W	Shortcuts de l'Info.		
E	Shortcuts de l'Info.		
R	Shortcuts de l'Info.		
T	Shortcuts de l'Info.		
Y	Shortcuts de l'Info.		
U	Shortcuts de l'Info.		
I	Shortcuts de l'Info.		
O	Shortcuts de l'Info.		
P	Shortcuts de l'Info.		
[	Shortcuts de l'Info.		
]	Shortcuts de l'Info.		
Backspace	Shortcuts de l'Info.		
Enter	Shortcuts de l'Info.		
Ctrl	Shortcuts de l'Info.		
Alt	Shortcuts de l'Info.		
Shift	Shortcuts de l'Info.		
Space	Shortcuts de l'Info.		
Tab	Shortcuts de l'Info.		
Q	Shortcuts de l'Info.		
W	Shortcuts de l'Info.		
E	Shortcuts de l'Info.		
R	Shortcuts de l'Info.		
T	Shortcuts de l'Info.		
Y	Shortcuts de l'Info.		
U	Shortcuts de l'Info.		
I	Shortcuts de l'Info.		
O	Shortcuts de l'Info.		
P	Shortcuts de l'Info.		
[	Shortcuts de l'Info.		
]	Shortcuts de l'Info.		
Backspace	Shortcuts de l'Info.		
Enter	Shortcuts de l'Info.		
Ctrl	Shortcuts de l'Info.		
Alt	Shortcuts de l'Info.		
Shift	Shortcuts de l'Info.		
Space	Shortcuts de l'Info.		
Tab	Shortcuts de l'Info.		
Q	Shortcuts de l'Info.		
W	Shortcuts de l'Info.		
E	Shortcuts de l'Info.		
R	Shortcuts de l'Info.		
T	Shortcuts de l'Info.		
Y	Shortcuts de l'Info.		
U	Shortcuts de l'Info.		
I	Shortcuts de l'Info.		
O	Shortcuts de l'Info.		
P	Shortcuts de l'Info.		
[	Shortcuts de l'Info.		
]	Shortcuts de l'Info.		
Backspace	Shortcuts de l'Info.		
Enter	Shortcuts de l'Info.		
Ctrl	Shortcuts de l'Info.		
Alt	Shortcuts de l'Info.		
Shift	Shortcuts de l'Info.		
Space	Shortcuts de l'Info.		
Tab	Shortcuts de l'Info.		
Q	Shortcuts de l'Info.		
W	Shortcuts de l'Info.		
E	Shortcuts de l'Info.		
R	Shortcuts de l'Info.		
T	Shortcuts de l'Info.		
Y	Shortcuts de l'Info.		
U	Shortcuts de l'Info.		
I	Shortcuts de l'Info.		
O	Shortcuts de l'Info.		
P	Shortcuts de l'Info.		
[	Shortcuts de l'Info.		
]	Shortcuts de l'Info.		
Backspace	Shortcuts de l'Info.		
Enter	Shortcuts de l'Info.		
Ctrl	Shortcuts de l'Info.		
Alt	Shortcuts de l'Info.		
Shift	Shortcuts de l'Info.		
Space	Shortcuts de l'Info.		
Tab	Shortcuts de l'Info.		
Q	Shortcuts de l'Info.		
W	Shortcuts de l'Info.		
E	Shortcuts de l'Info.		
R	Shortcuts de l'Info.		
T	Shortcuts de l'Info.		
Y	Shortcuts de l'Info.		
U	Shortcuts de l'Info.		
I	Shortcuts de l'Info.		
O	Shortcuts de l'Info.		
P	Shortcuts de l'Info.		
[	Shortcuts de l'Info.		
]	Shortcuts de l'Info.		
Backspace	Shortcuts de l'Info.		

Search for "add property," and you'll find the *Add Property to Essential Graphics* command. Click the empty field in the Shortcut column so it gets a blue outline, and then press the key that you want to be the shortcut. Since I'm on a Norwegian keyboard, I chose the Norwegian letter Å, which is not in use for any other commands.

Search for "add property" in the search bar.



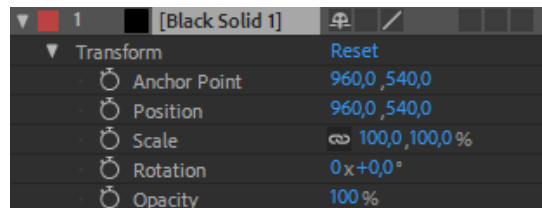
Supported types of property controls

So, what kind of properties can you add to the Essential Graphics panel? A lot! From the timeline, you can add these kinds of properties:

- **2D point properties**, such as Anchor Point, Position, and Point Controls
- **2D scale properties**, such as Scale
- **Angle properties**, such as Rotation and Angle Controls
- **Numerical sliders** (single-value), such as Opacity and Slider Controls
- **Layer Source Text**
- **Checkboxes, sliders, color pickers**

You can add supported properties from any effect or any layer property group (Transform, Masks, Material Options, and so on) as long as it is one of the listed property control types.

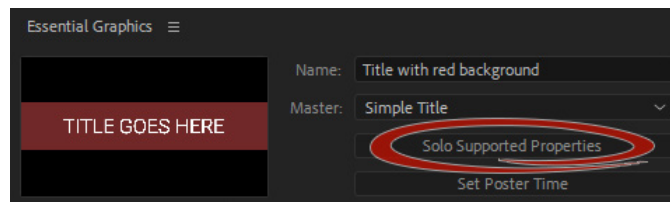
All the Transform group properties of a 2D layer can be added to the Essential Graphics panel.



There's a handy button in the Essential Graphics panel named *Solo Supported Properties*. This will hide all the unsupported properties in the timeline, so you can be sure that everything that's visible is supported. It does not change the view in the Effect Controls panel, though.

There's no indication in the panel if this feature is toggled on or off, but your view in the timeline should make it obvious.

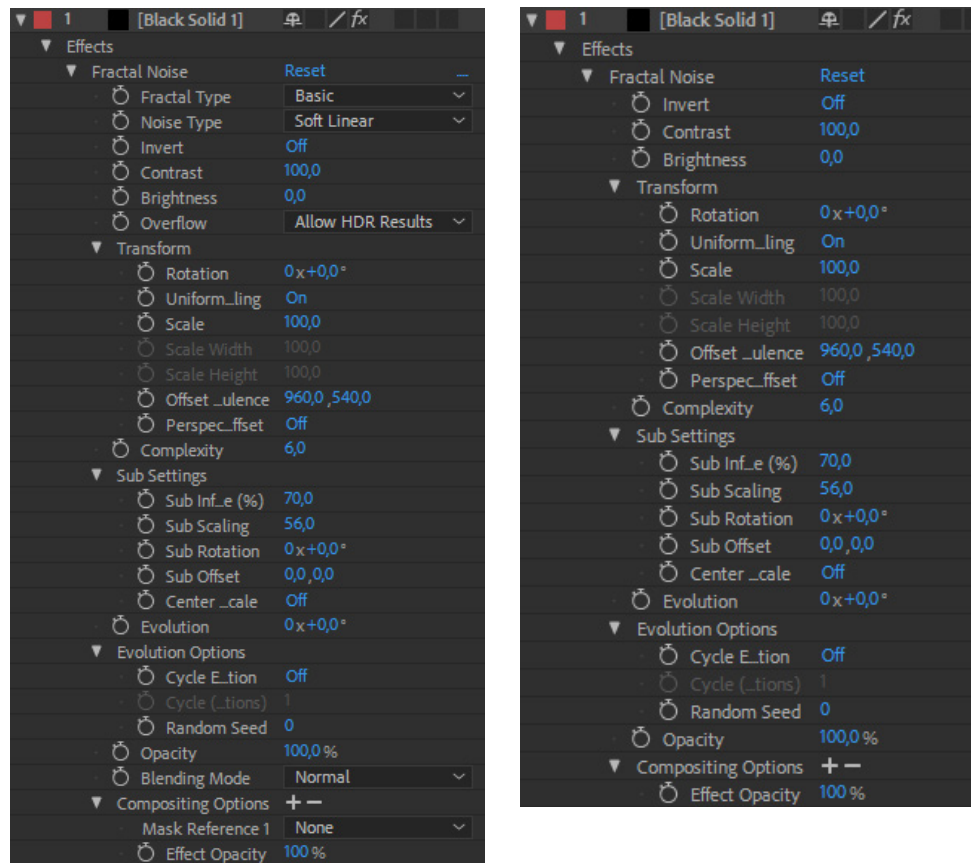
The Solo Supported Properties button in the Essential Graphics panel.



The following figure shows the timeline view of the Fractal Noise effect before and after the *Solo Supported Properties* button is clicked. Basically, every property in this effect is supported, except the drop-down menus. This is pretty much the standard for all property types.

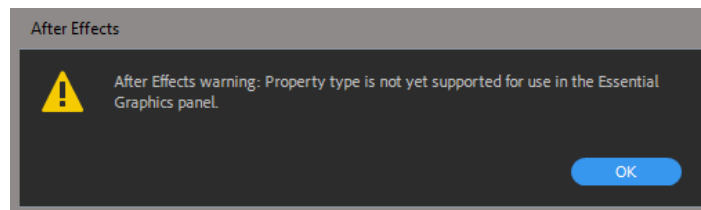
As of this writing, Blending Modes are not supported.

The Solo Supported Properties feature changes the view in the Timeline panel to show only supported properties. The left image shows the timeline view of the Fractal Noise effect before clicking the button, and the right image shows the view with the button toggled on.



If you try to add a non-supported property, you will be politely reminded by After Effects that the property is not supported.

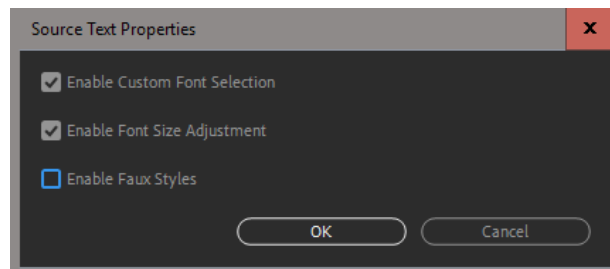
You will get a warning if you try to add a property that is not supported, and the property will not be added.



Just because you can't add 3D points to the Essential Graphics panel, that doesn't mean you can't let the user control them. You can add three sliders and link the x, y, and z positions to them. More on this later.

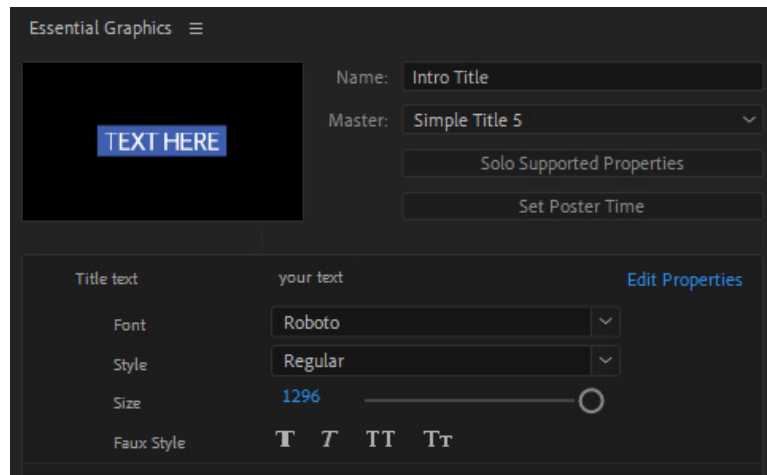
Font properties

Decide which properties you want to allow the user to change.



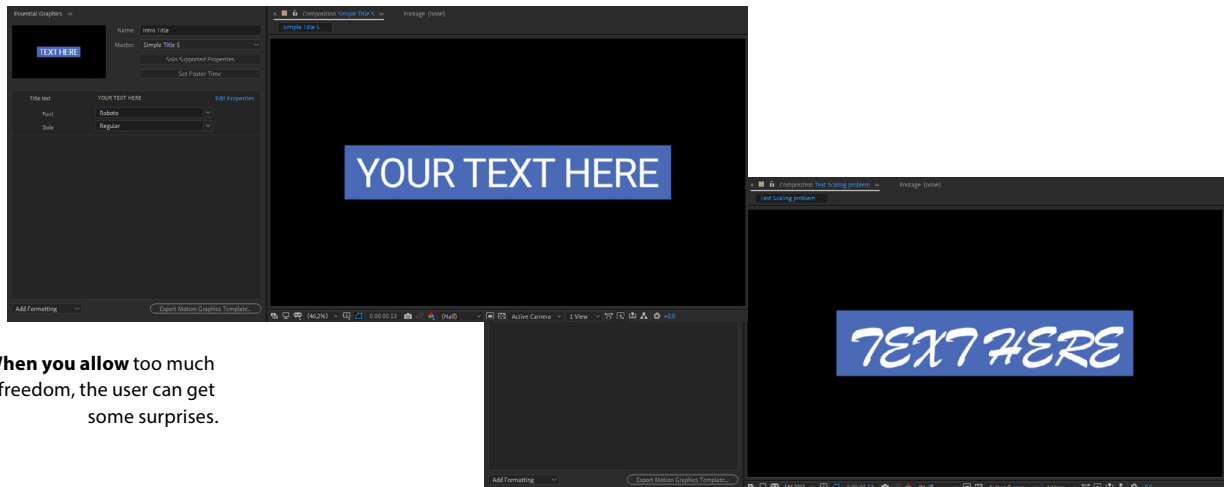
When you've added the source text to the Essential Graphics panel, you can click *Edit Properties* and decide which properties you want to allow the MOGRT user to change.

Here's an example where all the choices have been selected; enabling Faux Styles adds Faux Bold, Faux Italic, All Caps, and Small Caps controls.



Font controls do not work when the source text comes from other layers through expressions.

You should be aware that giving the user total freedom can have some unwanted side effects. See how the text sits lower in the box with the font Brush Script MT than with Roboto. You could give the user an extra slider for Vertical Text Offset, so it could be adjusted in the NLE, but it makes the MOGRT more cumbersome to use.



When you allow too much freedom, the user can get some surprises.

I recommend that you don't give the user full freedom when it comes to Font Size adjustment. You can't limit the size range, so the slider always goes from 0.1 to 1296. Besides, there's no way to access the size slider directly with expressions, so you can't easily make all other parameters scale proportionally.

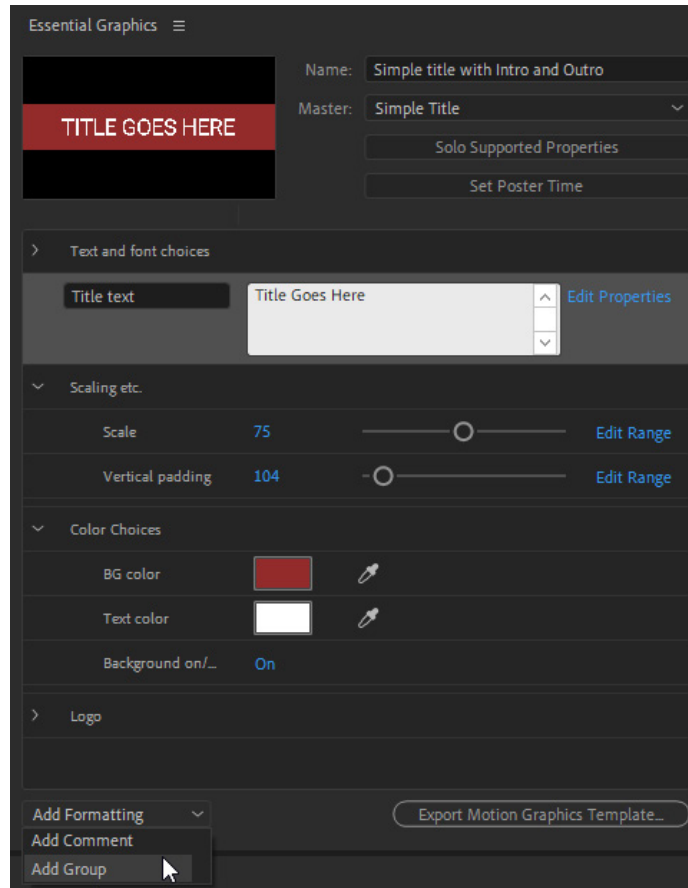
If you give the user a Slider Control to control the text size, you can limit the size range and link other parameters directly to it with ease. I tend to use a custom slider for scaling instead of enabling the Font Size Adjustment.

Add groups and comments in the Essential Graphics panel

Groups can reduce clutter and shorten long lists of editable controls. To make the MOGRT easier to navigate and use, put the controls into groups with descriptive names, and insert comments whenever something needs an explanation.

To create a new comment or group, click the *Add Formatting* drop-down menu in the lower-left corner of the Essential Graphics panel, and then select *Add Group* or *Add Comment*. If you have an item selected in the panel, the comment or group will be added immediately below that item. If you don't have anything selected, it will be added below all the existing controls.

Groups are not just a way of organizing the controls. They also serve as workflow guides for the user.



Groups can contain one level of Sub Groups. To add a control to a group, click and drag it over the group name. The twirl state of a group is written to the MOGRT when exported, so groups are shown in the Essential Graphics panel in Premiere Pro with the same twirl state.

Expressions don't have memory, so they're recalculated for every frame.

Most things that have a stopwatch in the timeline can be controlled by expressions.

Since expressions are based on JavaScript, they're case-sensitive. So, writing *Name* instead of *name* can throw an error message.

Using expressions to make user-friendly MOGRTs

Expressions are snippets of code that add custom capabilities to After Effects and automate repetitive tasks, enabling users to build flexible designs that respond to changes. Expressions can link properties across different effects, layers, or compositions; manipulate timing based on the behavior of other elements; create procedural animation; or drive properties with data.

Expressions can describe relationships between things.

These relationships are created when we add JavaScript-based code. Fortunately, we don't have to write code by hand. After Effects can help us write the code, as we'll see later.

We use expressions in MOGRTs to make the templates user-friendly. For example, instead of giving the user a slider for adjusting the width of a rectangle with text inside it, we can use an expression to make the rectangle automatically adjust as the text layer gets longer.

When you add a keyframe to a property, it's a hard-coded number.
When you add an expression, the code is calculating a number.

What can expressions do?

Expressions can both read and control values. So, in the case of the self-adjusting text background described previously, we can *read* the size of a text layer, and use that to *control* the size of a rectangle in its background shape layer. We can also make the fill color of that rectangle depend on the state of a checkbox, a color control, or a slider.

Expressions can also read and write words in text and understand math. The more math you know, the more advanced stuff you can do. If you know trigonometry, you can do all kinds of crazy things, but you can get a long way using only addition, subtraction, division, and multiplication. For more advanced math, you can copy and paste from web pages, other templates, and text documents.

There are also some things expressions cannot do. Blending modes, some font controls, and turning layers on and off are some examples of stuff that cannot be controlled by expressions. We'll explore some workarounds for these cases later.

Some features in After Effects that use expressions

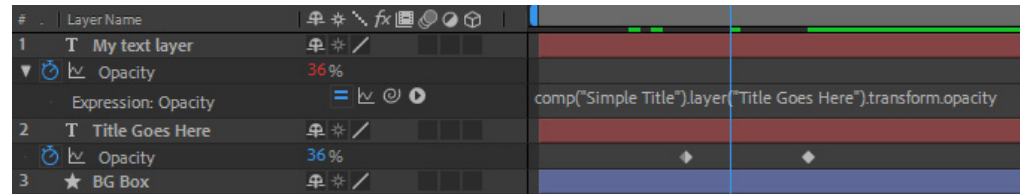
You may have been adding expressions in After Effects without knowing it. If you've ever used *Copy with Property Links* or *Copy with Relative Property Links*, you've added expressions to one or more properties in the targeted layer. Some animation presets also use expressions.

The **Copy with Property Links** feature creates expressions automatically.

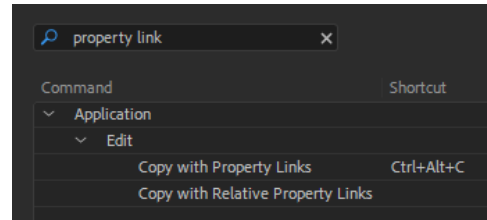
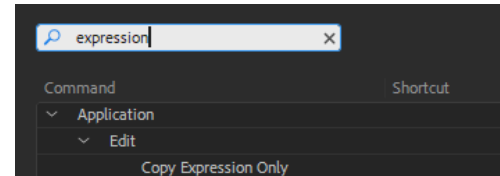
Edit	Composition	Layer	Effect	Animation	View
	Undo	Clear Layer(s)			Ctrl+Z
	Can't Redo				Ctrl+Shift+Z
	History				>
	Cut				Ctrl+X
	Copy				Ctrl+C
	Copy with Property Links				Ctrl+Alt+C
	Copy with Relative Property Links				
	Copy Expression Only				
	Paste				Ctrl+V
	Clear				Delete

For example, after choosing *Copy with Property Links* from the Edit menu with the Opacity of layer 2 selected, I click *Paste* from the same menu with layer 1 selected. The Opacity of layer 1 will now be controlled by the Opacity setting of layer 2, as shown in the following figure.

The red number indicates that the Opacity of layer 1 is now controlled by an expression. The expression reveals that it's the Opacity of layer 2 (named Title Goes Here) that controls it.

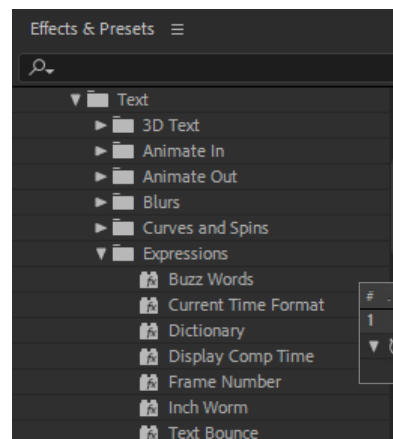


You can create your own keyboard shortcut for Copy with Relative Property Links and for Copy Expression Only.

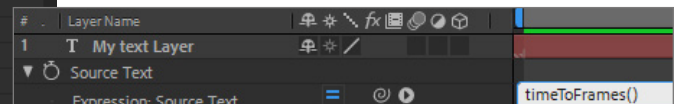


There's a whole category of text animation presets that use expressions. You'll find them in the Effects & Presets panel, at **Presets > Text > Expressions**.

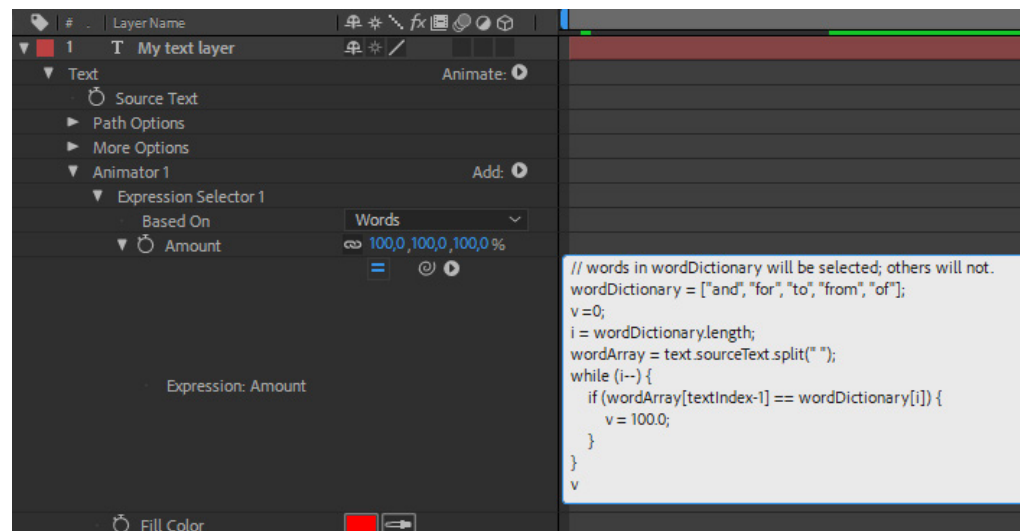
These text animation presets all use expressions.



The **Frame Number** animation preset adds this simple expression to your text layer to show the current frame number as text.

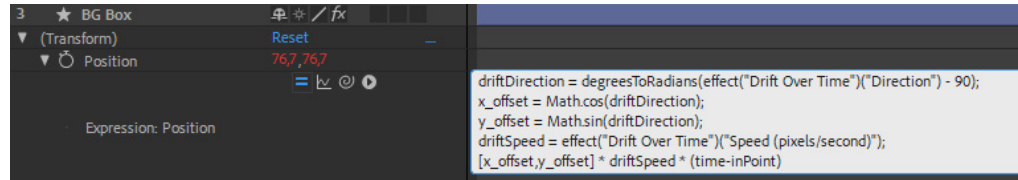


The **Dictionary** animation preset uses a more advanced code.



These are not the only effects presets that use expressions. Another example is the *Drift Over Time* preset, located under Behaviors in the Effects & Presets panel.

The Drift Over Time animation preset also uses an expression to calculate the Position.

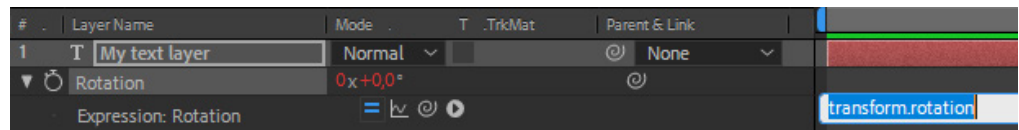


As you can see, there are many ways that After Effects uses expressions internally, without telling you it does. But only when you start adding expressions to chosen parameters yourself will you unleash the real power of expressions.

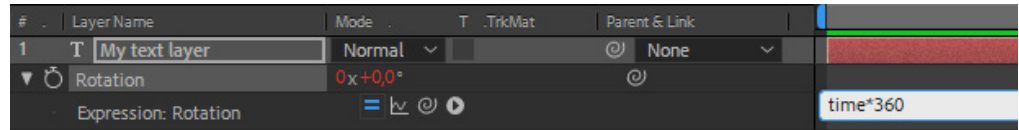
How to add an expression to a property

To manually add an expression to a property, **Alt-Click** on the stopwatch of the property. In the example here, I've Alt-clicked on the *Rotation* stopwatch. The numbers turn red to tell me they're controlled by an expression, and the code is highlighted, so I can start typing (or pasting) my code.

Alt-click on the Rotation stopwatch to add an expression to the Rotation parameter.

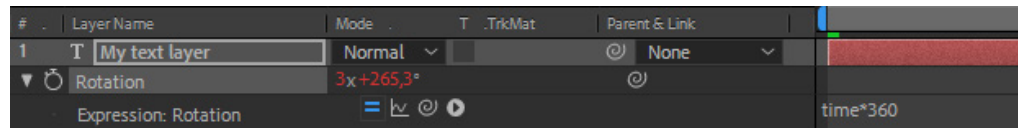


Type the expression, or paste code from a web site, a text document, and so on. Click outside the expression text field to enable the expression.

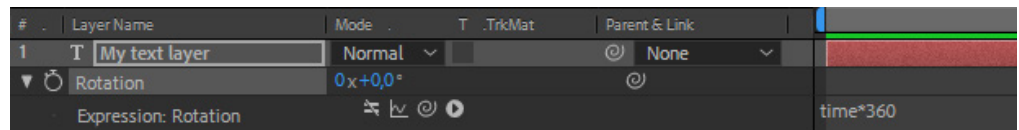


To remove the expression, **Alt-Click** on the stopwatch again or choose **Animation > Remove Expression** while the property is selected in the timeline.

Click the blue equals sign to *temporarily* disable the expression. The equals sign turns white with a slash over it, indicating that it's disabled, and the numbers turn blue again.



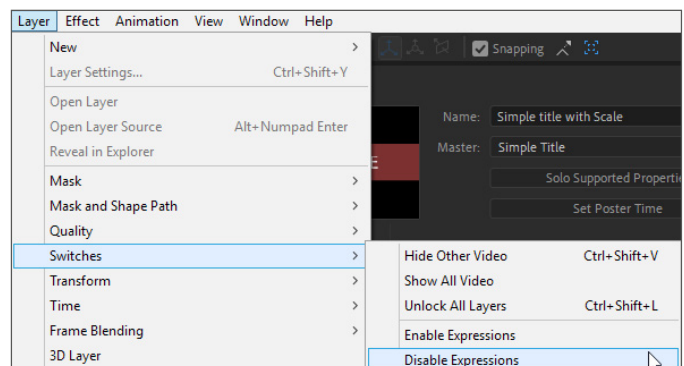
Click the equals sign to disable the expression.



Enable or disable all expressions on selected layers

All expressions on a layer can also be disabled or enabled via two menu commands by choosing **Layer > Switches** submenu, or by right-clicking on a layer and opening the **Switches** submenu.

You can also create keyboard shortcuts for these commands.



Let After Effects write the code for you

Since expressions are code, many designers stay away from them, thinking that they don't know how to write JavaScript code and consequently can't write expressions. Fortunately, they're wrong.

You don't need to know how to code to create expressions!

There are several ways that After Effects can help you write code, and you can also copy and paste from many different resources.

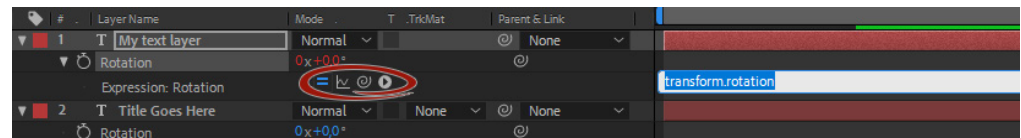


The Expression Pick Whip.

The Expression Pick Whip

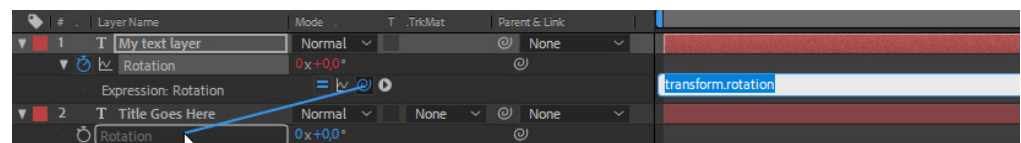
If you **Alt/Option-click** the stopwatch for a property to add an expression to it, the Expression Pick Whip appears. It's the swirly spiral thingy among the four icons that appear close to the red numbers.

This is where the Expression Pick Whip appears.



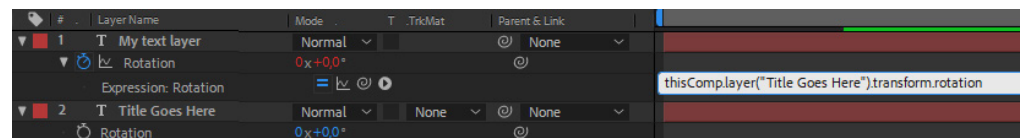
To use the Expression Pick Whip, click on it and keep pressing the left mouse button while you drag the mouse and point the cursor to the property you want the expression to link to. You'll see a blue line appearing from the pick whip icon to the point where the cursor is.

Point the Expression Pick Whip to the property you want to link to.



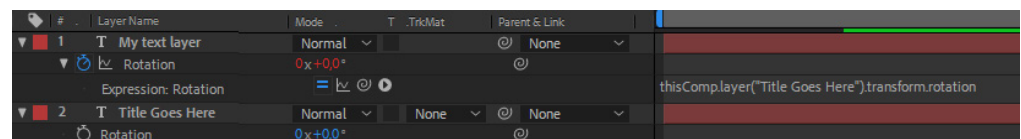
When the cursor reaches the property you want the expression to link to, let go of the mouse. If you succeed, the blue line disappears, and After Effects adds code to the expression text box.

When you release the mouse button, code is added.



The expression doesn't "take" while the expressions text box is white. To activate the expression, click somewhere outside the white expression text box, or use the **Enter** key on the numeric keyboard if you have one. The standard Enter key will not work.

The expression doesn't take until the expression text box is gray.



How the pick whip works

Here are some rules for how the pick whip works.

If no text is selected, and the text cursor is in the expression text box, code is inserted where the cursor is parked.

If some text is selected in the expression text box, the pick whip will replace it.

If you're not in the expression text box when you pick whip, it will replace your whole expression.

You can't pick whip from the Effect Controls panel, but you can pick whip to it.

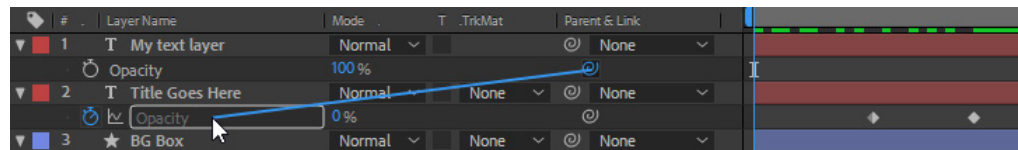
The Property Link Pick Whip

Even though the Expression Pick Whip is easy to use, it's not very discoverable, since you need to **Alt/Option-click** the property stopwatch to activate the expression text field. The 15.1.0 (2018.1) version of After Effects introduced a new way to quickly link properties with expressions: the Property Link Pick Whip.

It works the same as the Expression Pick Whip, except that it saves you a step by automatically adding an expression to the linked property. It's also visible in the Parent & Link column.

So, in the 15.1.0 version, you no longer need to know how to enable or write expressions in order to link properties. This is a very nice addition.

The Property Link Pick Whip lets you link properties together with a simple point-and-click.



You'll find the new pick whip in the renamed *Parent & Link* column of the Timeline panel. You can point it to properties and values in the Timeline, Effect Controls, or Project panels.

To disable a property link (and the expression it created), **Alt/Option-click** the Property Link Pick Whip. This does not delete the expression; it just disables it. The result is the same as when you click the blue equals sign to disable the expression.

This new Property Link Pick Whip is a huge time-saver when making complex MOGRTs where a lot of properties are connected in many different ways.

Expression Language menu

When you **Alt/Option-click** the stopwatch of a property to create an expression, you also get the icon for the Expression Language menu, right beside the Expression Pick Whip icon. It looks like a round Play button, but users of older versions of After Effects will recognize it as the old icon for menus.

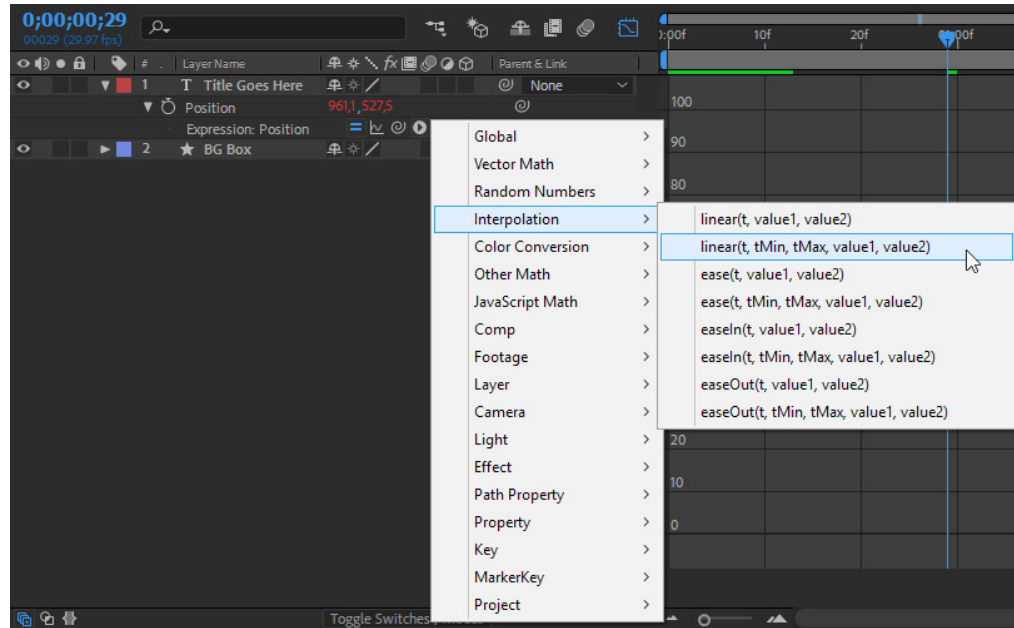
When you click the icon, you get a menu with hundreds of code snippets for achieving many different things, like generating random numbers, getting the size of layers and comps, controlling interpolation, and so on.

When you click on one of these code snippets, it's inserted into the expression text box. Sometimes, you'll get a finished piece of code. Other times, you'll only get the correct syntax, and you must replace the words with property names. But at least you have all the special words written correctly, and the commas, the brackets, and the parentheses are all in the right place.

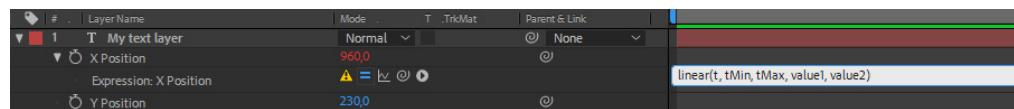
Think of the Expression Language menu more as a guide to writing correct code than as a coding device.

Exploring the Expression Language menu is a good way to get an overview of the endless possibilities you have when you use expressions.

The Expression Language menu helps you write code with the correct syntax.



Using the Expression Language menu, I got the right syntax, but it's not finished code.

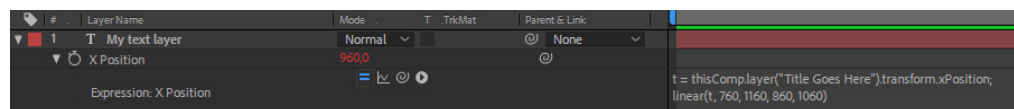


To finish this expression, I inserted a line break before the code, and defined what **t** should be in the following manner: I typed **t =** and then pointed the Expression Pick Whip to the X position parameter in another layer. Then I typed a semicolon (;) to tell After Effects that it's the end of a line of code.

This tells After Effects that every time it sees **t**, it means “the X Position of the layer named Title goes here”. Then I changed the numbers, so that when **t** goes from 760 to 1160, the X position of my text layer goes from 860 to 1060.

I managed to write this code with the help from both the Expression Language menu and the Expression Pick Whip. I would not have been able to write the code without this help because I don't know JavaScript well enough. I'm not a programmer.

The code after my changes.



Make sure you let After Effects help you in different ways, so you don't waste time writing faulty code. There's also another great way to work with expressions: Copy from web pages, text documents, and After Effects comps.

Copy and paste expressions from documents and websites

This is by far my favorite method! There are a lot of web pages, forums, and other resources where you can copy the code, and then just paste it right into the expression text box in After Effects. One great resource is MotionScript.com. The site is getting old, but the info is still solid gold for people who want to learn expression, or just want to borrow some code.

Dan Ebberts runs MotionScript.com, a great resource for learning and finding expressions.

MotionScript.com

ArticlesContact

Dan Ebberts' AE Expressions and Scripting Resource

Welcome

Welcome to my site. What I've tried to assemble here is a resource for folks that want to learn more about expressions and scripting in After Effects. It's been quite a while since I've had time to update things, but I'm hopeful that I'll be able to roll out new topics more frequently now. I have an enormous backlog of content, and I'm trying to organize and present it in a way that should answer many questions that come up repeatedly in the AE forums. If you have comments, questions, or find an error—please [contact me](#).

About Me

I've been doing freelance script and expression design pretty much since those features were added to After Effects. I do a lot of workflow automation where the client needs a script to read in a large amount of data and use that data to drive media and text replacement and batch render (or queue) all the variations. Sometimes this will amount to reading a monthly schedule and generating promos for the entire month. I've set up a bunch of real-time operations, where a script waits for a job to come in via an XML file and then loads the appropriate project, does the media and text replacement and renders the result on the fly.

I also do a lot of math-intensive projects. Most of my projects involve scripting, but some require complex animation rigging via expressions. I've also built a lot of automation tools for animators. I also have experience with scripting for Photoshop, Illustrator, InDesign, and Flash.

If you click the link named *Realistic Bounce and Overshoot* on the front page, or navigate to <http://motionscript.com/articles/bounce-and-overshoot.html>, you'll find a longish article on this subject. Scroll down to the heading *Keyframe Overshoot*.

Copy the blue Keyframe Overshoot code by selecting it and pressing Ctrl/Cmd+C.

Keyframe Overshoot

Perhaps a more common overshoot application would be adding overshoot to a keyframed animation. This is actually fairly straightforward because the expression language gives you access to a property's velocity. In this case we'll use the `velocityAtTime()` function to get the incoming velocity at the most recent keyframe. This is the keyframe overshoot expression:

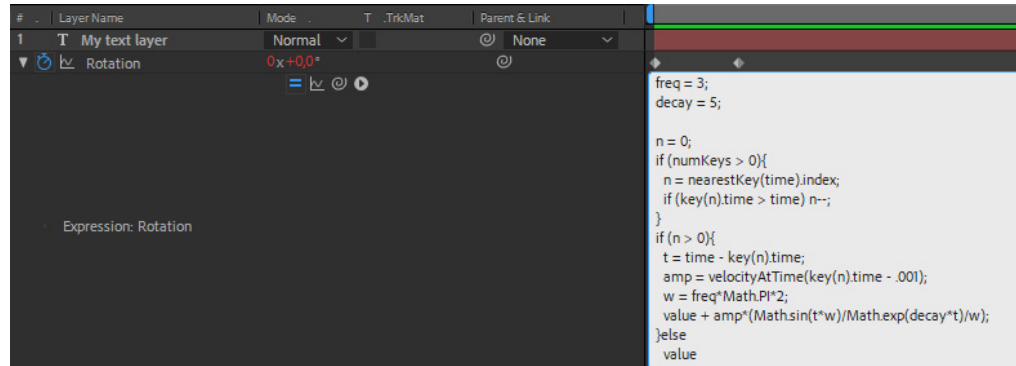
```
freq = 3;
decay = 5;

n = 0;
if (numKeys > 0){
  n = nearestKey(time).index;
  if (key(n).time > time) n--;
}
if (n > 0){
  t = time - key(n).time;
  amp = velocityAtTime(key(n).time - .001);
  w = freq*Math.PI*2;
  value + amp*(Math.sin(t*w)/Math.exp(decay*t)/w);
}else
  value
```

Copy the code from the web page, and then paste it in the expression text field, and you have bouncy movement. If you have at least two different keyframes, that is.

It works on a lot of different parameters, like Position, Scale, Rotation, and so on, but you must have at least two keyframes and a reasonably fast animation to see the effect. The clever code calculates the velocity in to a keyframe and uses that velocity as the amplitude for the overshoot calculation. Dan also made the code easy to use. Changing the *frequency* and *decay* numbers in the first two lines will change the look and feel of the bounce. He obviously didn't sleep during those trigonometry classes.

Using Dan Ebberts' Keyframe Overshoot expression on my text layer made the boring animation (Rotation goes from -180° to 0°) go from a dead stop to a nice bouncy movement after the last keyframe.

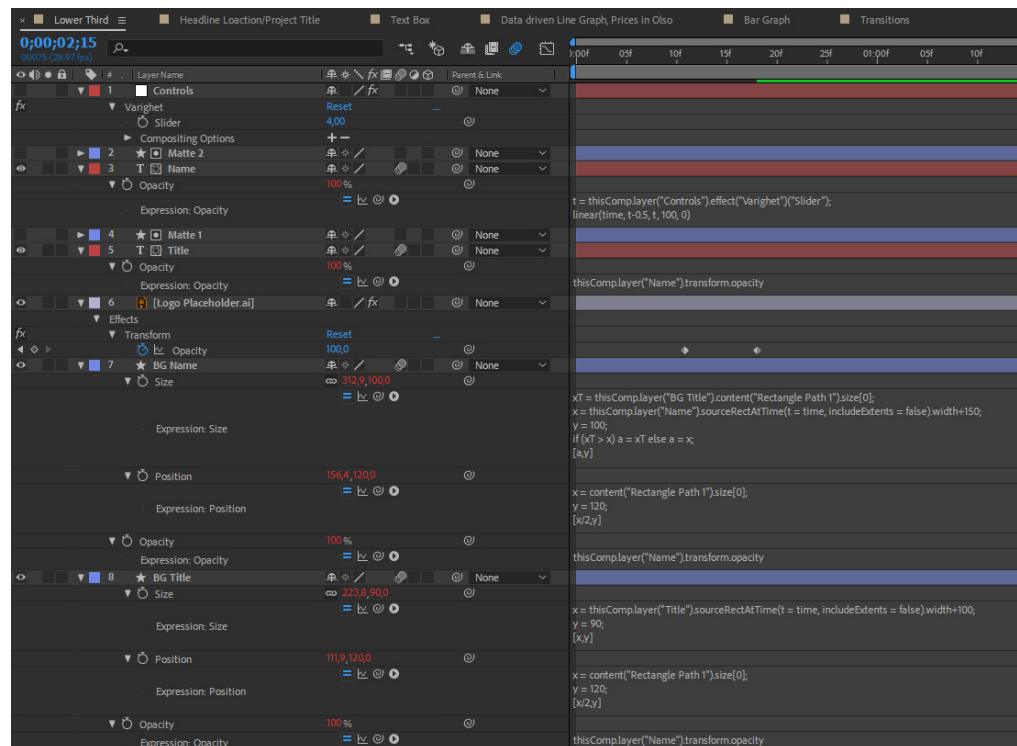


You don't have to limit the bounce expression to the standard Transform parameters. You can use it on almost anything, like the *Trim Paths* parameters on shape layers, or the keyframes in text animation presets. You can even put some more life into the old Glow effect with this code, if you use it on Glow Radius and Glow Intensity.

There's a list of online resources in the appendix. Explore them to find nice expressions you can use in your projects. I recommend that you create a collection of expressions in a text document, so you can easily find them later.

When you have made a few MOGRTs, you can often reuse the code in new MOGRTs, which can help you save a lot of time. Here are all the expressions in a lower third comp in my MOGRT Creation Start Project. I copy and paste like a crazy person from comps in this project when I'm building new MOGRTs.

Expressions in a simple lower third MOGRT.



Expression language fundamentals

Now that you know about the ways you can use code without writing it yourself, let's have a look at the basics of the code syntax and some examples that show how you can use expressions. The intention is not to make you an expert coder, but to help you understand some of the syntax. This makes it easier for you to change the code you borrow from web pages so they work in your projects.

Expressions are case sensitive.

Case sensitivity

As mentioned before, expressions in After Effects are based on JavaScript and adhere to the JavaScript rules. This means they're case sensitive, so there's a difference between writing *myValue* and *MyValue*. *Slider* is different from *slider* and so on.

There are expressions that can change all text from uppercase to lowercase or the other way around (*toLowerCase* and *toUpperCase*). To change single characters or words with expressions requires different and more advanced code.

Anyway, when you write expressions, make sure you type them correctly.

Expressions ignore spaces and line breaks

Since JavaScript pretty much ignores spaces between words in text, we can use them to make the code easier to read.

I often find it hard to read code like this:	<code>linear(time,3,4,100,0)</code>
It's sometimes easier to read code like this:	<code>linear (time, 3, 4, 100, 0)</code>

The first code feels crowded. The second one has more "air" due to the extra spaces, but After Effects will treat these as the same code. Adding some spaces also makes it easier to change individual bits of the code. If I double-click one of the words or numbers in the code where I added no spaces, it will select the parentheses plus everything inside, which is almost never what we want. If I double-click the *time* in the code with extra spaces, only that word plus the comma will be selected.

Expressions don't care much about indents and line breaks either, as long as you don't add them in the middle of names and so on. Adding indents and line breaks helps the readability and shows the code structure. So, to make the code readable by humans, we use them a lot. Here's an example from <http://conigs.com/after-effects-a-better-bounce/>. See how the indents make the structure of the code more apparent.

```
bounces = 4;           //total number of bounces
duration = .25;        //duration of each bounce in seconds
amp      = .05;        //multiplier for incoming velocity used in bounce
decay    = 3;          //exponential decay of bounce height
n=0;
if(numKeys>0){n=nearestKey(time).index;if(key(n).time>time){n--;}}
    n==0?t=0:t=time-key(n).time;
    freq=1/duration;
    mult = (bounces-Math.floor(t*freq))/bounces;
if (n>0 && mult>0) {
    v=velocityAtTime(key(n).time-0.001)*amp; //velocity to use
    b=Math.abs(Math.sin(freq*t*Math.PI))*Math.pow(mult,decay); //bounce calculation
    value=v*b;
} else {value;}
```

Add comments to your code

If you wonder what the double forward slashes mean, those mark the beginning of a comment. Everything after them on a line of code will be ignored by After Effects.

Comments are meant for humans to read. Adding comments will help the user (or the coder, three months later) understand the code. Don't confuse the double forward slashes with a single forward slash, which means divide.

// means "Comment starts here."

Comments are ignored by After Effects.

Use existing values as a starting point for expressions

A good way to start writing expressions is to use the pick whips to start the expression, and then change the expression using simple math operations, such as these:

+ *Add*

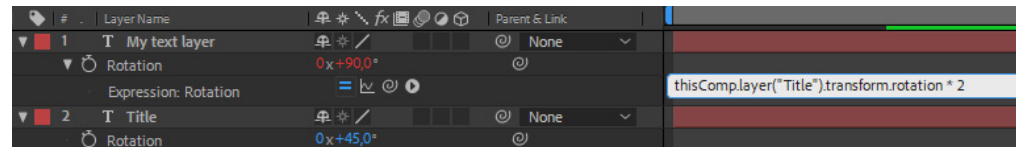
– *Subtract*

* *Multiply*

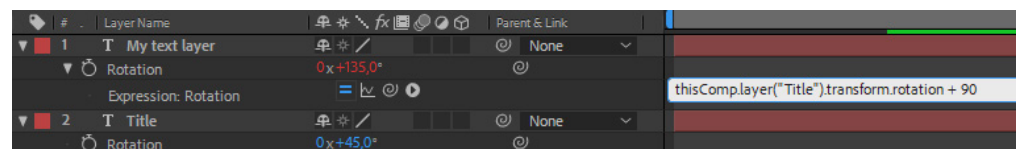
/ *Divide*

For example, adding ***2** at the end of the expression will double the result. Adding **+90** will, well, add 90 to the result. Adding **/4** after the code will divide the result by 4, and so on.

I pointed the Property Link Pick Whip to the Rotation of the Title layer, and then typed ***2**. The upper layer will now rotate twice as fast as the lower one.



I pointed the Property Link Pick Whip to the Rotation of the Title layer, and then typed **+90**. The upper layer rotation value is now always 90 degrees higher than on the lower layer.



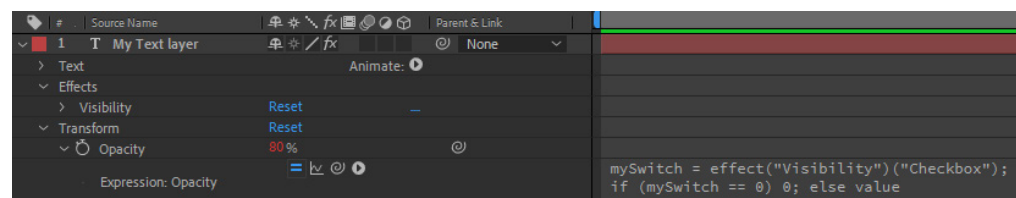
There's also an expression code name for the existing value of a parameter, and it's simply **value**. We often use this in if/else statements. Say I use this expression on Opacity on a layer:

```
if (mySwitch == 0) 0; else value
```

Since checkboxes output **1** when they're on and **0** when they're off, this will set the Opacity to 0 if the switch is off, and to whatever its value was if it's on. Of course, we also need to tell After Effects what this thing I called *mySwitch* is. I did this by pointing the Expression Pick Whip to the checkbox that I added.

I wrote `mySwitch =`, and then pointed the Expression Pick Whip to the checkbox. Then I added a semicolon and wrote the second line.

Don't worry if you don't understand the code now. You'll learn more about if/else expressions later.



The original name of the effect was *Checkbox Control*, but I renamed it to *Visibility*, which is much more informative. Note that I ended the first line with a semicolon. Ending lines (except the last line) with a semicolon is a JavaScript rule that you should stick to if you want to avoid trouble.

Punctuation in expressions

The Expression Pick Whip and the Property Link Pick Whip will always write code that sticks to the rules. So, using them is a good way to learn the structure of the expression code and the punctuation rules.

After pointing the Property Link Pick Whip to the Opacity of the layer named *Title*, I got this code:

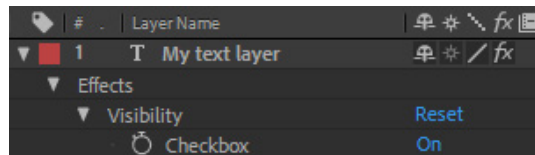
```
thisComp.layer("Title").transform.opacity
```

As you can see, each step (comp > layer > property) is separated by a period. The following code has no periods because the effect I linked to was on the same layer.

```
mySwitch = effect("Visibility")("Checkbox");  
if (mySwitch == 0) 0; else value
```

But you can see that the effect name (Visibility) and the parameter (Checkbox) are both within parentheses. These reflect the hierarchy in the timeline, so *effect("Visibility")("Checkbox")* means “the *Checkbox* parameter in the effect named *Visibility*.”

The parentheses in the code reflect the hierarchy in the timeline.



Parentheses are also used for grouping items together, and we use commas to separate them. Even the simple wiggle expression uses them. The code here, if used on Position for a layer, will make it shake 10 times per second and move up to 20 pixels each time. We'll get back to such grouped items later.

```
wiggle ( 10, 20 )
```

Layer names, effect names, and so on must be in quotation marks, like this:

```
thisComp.layer("Title")
```

You must use straight quotation marks in expressions.

Make sure you don't get the wrong kind of quotation marks when copying expressions from websites and Word documents. Only straight quotation marks will work. The “ ” quotation marks are not the same as the " " quotation marks, so using the wrong ones will return an error message.

What's a variable?

I've been using variables a few times without telling you what they are. Sorry about that. Variables are names that we give a value and that we can use later within the expression. You can use any word as a variable, except those reserved by After Effects like comp, time, opacity, and so on.

We define a variable by writing its name followed an equals sign.

```
txtColor =
```

Then we point to a parameter or write code that says what we mean when we write the name of the variable later in the expression. Here, I linked to a slider named *Text Color Choice* on a layer named *Controls*.

```
txtColor = Math.round (thisComp.layer("Controls").effect("Text Color Choice")  
("Slider"))
```

This tells After Effects that whenever I write `txtColor`, it really means `Math.round (thisComp.layer("Controls").effect("Text Color Choice")("Slider"))`. Using variables can greatly simplify your code, so it looks better and is easier to understand.

```
txtColor = Math.round (thisComp.layer("Controls").effect("Text Color Choice")
("Slider"));

if ( txtColor == 1 ) 0; else 100
```

The result is the same as if I write the following code, which is a bit harder for humans to read.

```
if ( Math.round (thisComp.layer("Controls").effect("Text Color Choice")
("Slider")) == 1 ) 0; else 100
```

The variable name doesn't have to be long. One letter is actually enough, but naming a variable just `A` or `B` doesn't tell you what it is, so I like to use more descriptive names. When your expressions get more advanced, the difference in readability with and without variables increases.

Some designers type `var` before the variable name to make it really clear that it's a variable. This is a good programmer's habit, but it's not really necessary. These two lines of code are equal.

```
mySwitch = effect("Visibility")("Checkbox")

var mySwitch = effect("Visibility")("Checkbox")
```

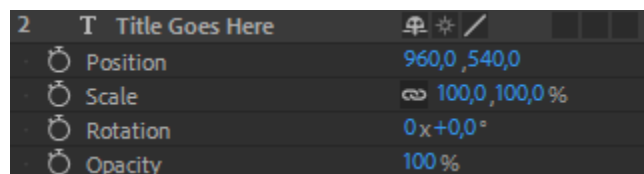
Variables only work within the same expression.

Since expressions do not have a memory, variables need to be defined in every expression. You can't reuse variables from expressions on other properties.

What's an array?

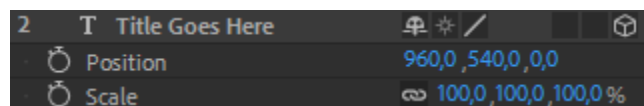
An array is a programmer's way to store related data in an orderly fashion. It's often used for multidimensional parameters. *Say what? Multidimensional?* It means a parameter that has more than one value. Opacity and Rotation have only one value. Position and Scale have two values (x and y) by default.

A 2D layer has both one-dimensional and two-dimensional parameters.



If we make the layer a 3D layer, Position and Scale suddenly have three values.

3D layers also have three-dimensional parameters.



After Effects even uses four-dimensional parameters, like Colors, which have values for Red, Green, Blue, and Alpha. You can create your own arrays with as many values as you want. Arrays are always held in square brackets, and values are separated by commas, like this.

```
myArray = ["100","200","300"];
country = ["Norway","Sweden","Finland","Denmark","Iceland"];
```

Here's something you need to know when writing code: Programmers start counting from 0! Yes, it's confusing, and you can love it or hate it, but you can't change it. So learn to live with it.

0 is the first, 1 is the second, 2 is the third, and so on. A number in square brackets tells the expression which value (dimension) you want.

```
country = ["Norway","Sweden","Finland","Denmark","Iceland"];
country[1]
```

Programmers start counting from 0.

This means that the X component (horizontal position) of a layer's Position property is accessed with "position[0]" and the Y component (vertical position) is accessed with "position[1]".

For 3D layers the Z component (depth) is accessed with "position[2]".

The code here will return *Sweden*, since the second value in the country array is *Sweden*, and *[1]* means *value number 2*. Ending the expression with *country[0]* instead would return *Norway*, which is the first value. You'll get used to it.

To watch After Effects write code with arrays, use the Pick Whips. Linking *Position* or *Scale* on one layer to *Opacity* on another will give you code similar to this.

```
temp = thisComp.layer("My text layer").transform.opacity;
[temp, temp]
```

Why did this happen? Because Opacity has only one dimension, and Position needs two. After Effects knows this and makes sure the code works by defining the variable named *temp* and then uses it for both x and y position.

Make the layers 3D layers, and do the same thing, and After Effects creates an array with three dimensions (x, y, and z position).

```
temp = thisComp.layer("My text layer").transform.opacity;
[temp, temp, temp]
```

You can define an array as a variable, but that will not be explained in this book. Explore tutorials on JavaScript and expressions to learn more about this.

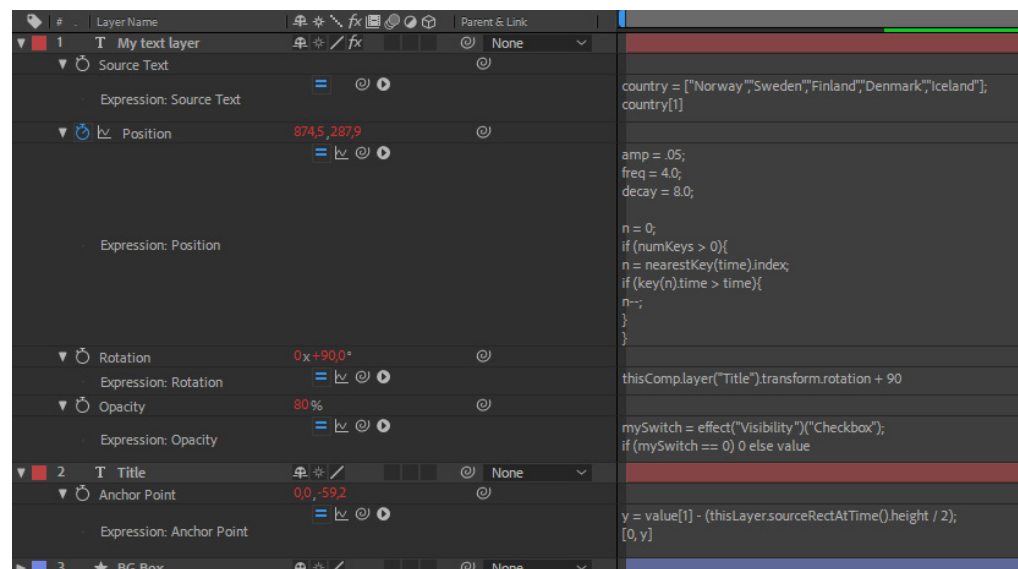
What's a function?

Functions are blocks of code designed to perform a particular task. With functions, you can reuse code within an expression. Define the code once and use it many times. But this topic is beyond the scope of this book. If you want to learn about functions, I recommend that you read up on JavaScript coding. There are many places on the web where you can learn this, and you can also get free apps for your smartphone, so you can learn how to code while you're on a plane.

Useful keyboard shortcuts for expressions

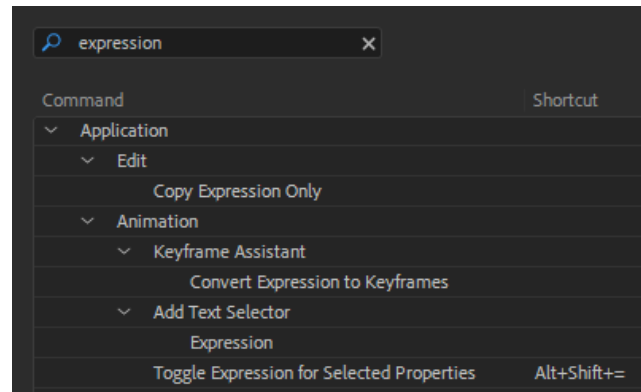
There's one keyboard shortcut for expressions you just have to learn, and that's **EE**. Pressing E twice will reveal the properties that have expressions on all selected layers.

Type EE to reveal all parameters with expressions on selected layers.



If you search for *expression* in the keyboard shortcuts dialog you don't get many hits. The *Copy Expression Only* feature is very useful, so I recommend that you create a shortcut for that one. Be aware that these keyboard shortcuts do not work when you're typing inside the expression text box.

There aren't many keyboard shortcuts available for working with expressions

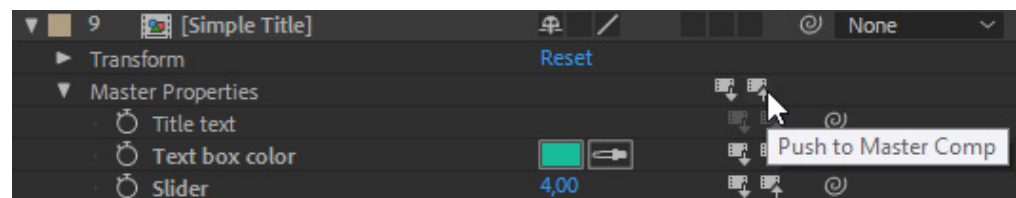


Don't forget that keyboard shortcuts for *Copy With Property Links* and *Copy With Relative Property Links* are also useful, since they create expressions.

Be careful when using expressions with Master Properties

The 15.1.0 version of After Effects introduced Master Properties, which will probably change the way you work with motion graphics design. They're especially useful when you have multiple instances of comps nested in other comps. Before 15.1.0 we had to use multiple duplicate comps to achieve the same as we can now do with just one comp.

Master Properties are great, but you should double-check that your expressions work as expected when you do changes with Master Properties.



But Master Properties work with expressions differently from property values and keyframes. The context in which expressions are evaluated can significantly change the result of an expression, and Master Properties creates a different context for expressions: The composition, layer, and property group are all different.

This doesn't mean that expressions won't work when you use Master Properties—it just means that you need to take extra care when you're copying and pasting and writing code. And make sure the expressions work as expected.

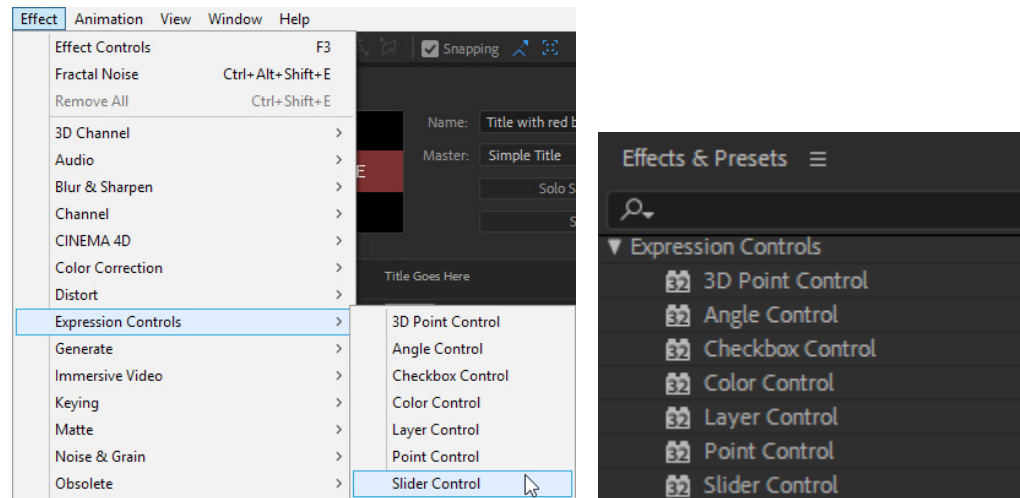
Expression Controls—your best friends

Expression Controls are effects that do absolutely nothing, but they are still extremely useful. You can use them to control *other* parameters using expressions.

You can add a slider that moves a layer only within certain limits, or let the user select between several options by dragging a slider. You can add checkboxes that turn layers on or off, or change color schemes on text and graphics layers. You can add angle controls that drive the rotation of several layers, and you can add color controls that change the color of text, glows, or shapes.

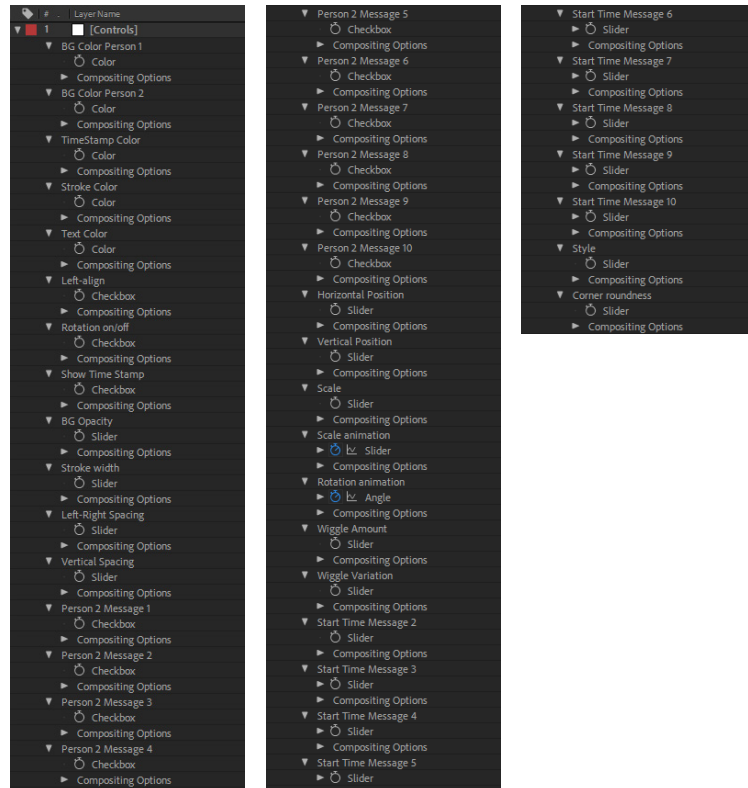
There are seven types of Expression Controls in After Effects, but the 3D Point Control and the Layer Control are not supported by the Essential Graphics panel, so they're not used for MOGRT design.

Expression Controls

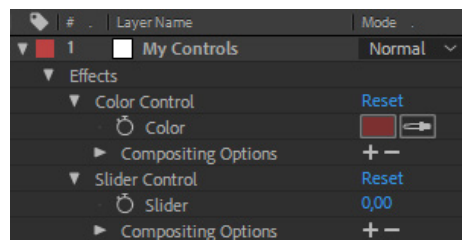


You will find these Expression Controls in the Effects & Presets panel and if you choose **Effect > Expression Controls**. When you're creating flexible MOGRTs with lots of choices and adjustments, you'll use a lot of Expression Controls. For more restricted MOGRTs you'll probably only use a few.

Flexible MOGRTs made for Adobe Stock and other places you can buy MOGRTs can have a lot of sliders, color controls and checkboxes.



Restricted MOGRTs you create for TV or web series will often have very few controls available.

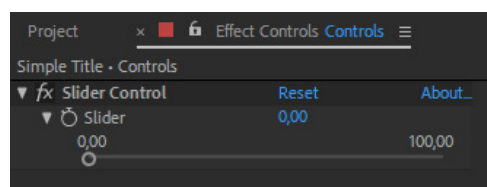
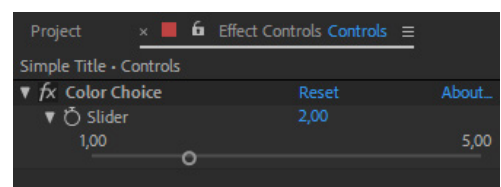


The Expression Controls I use the most when building MOGRTs are Slider Controls, Checkbox Controls, and Color Controls. Let's have a closer look at them.

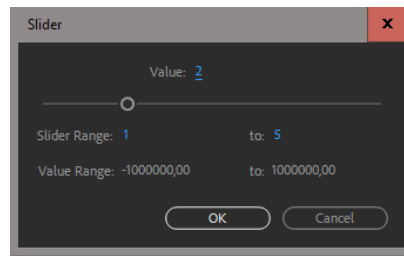
The Slider Control

This is my favorite Expression Control because it has so many uses. When you add a Slider Control to a layer, it will be named *Slider Control* and have a slider that goes from 0 to 100. You can rename it, of course. Right-click on the word *Slider* to find *Edit Value*, which opens a dialog box where you can set the value and the range of the slider.

The slider range is 1–100 as standard, and the value is 0, but it can be anything you want.



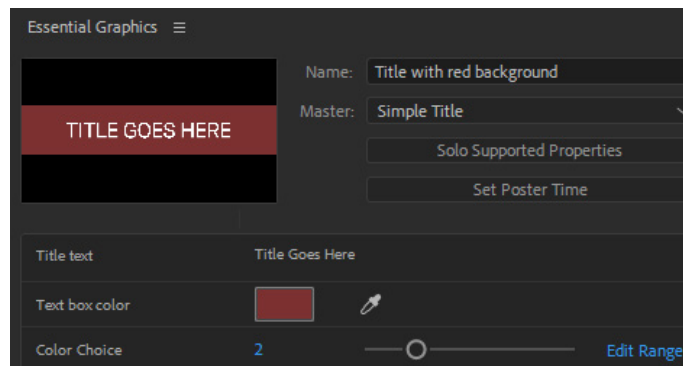
You can set the slider range and value to whatever you need (within limits).



Since this slider will be used for choosing between five colors, I changed the range to 1–5. With some if/else statements in my expression, I can make the background color change according to the slider value. We will take a look at if/else expressions shortly.

Instead of right-clicking to get to *Edit Value*, I usually wait until I've added the slider to the Essential Graphics panel, where I can click *Edit Range* to get to the same dialog box using one click less.

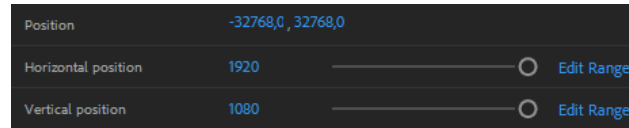
It's easier to edit the slider range from the Essential Graphics panel since there's a dedicated button for it.



Combine Separate Dimensions and Slider Controls

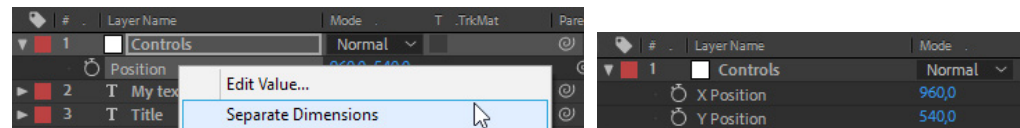
If you add the Position parameter of a layer to the Essential Graphics panel, the numbers can be changed all the way up to 32,768, which is 2^{15} . This means that the user can really make a mess, since it's possible to place the layer way out of the visible area.

Position values go all the way up to +/- 32768, which can cause some trouble. Slider Controls helps the MOGRT user keep the values within certain limits.

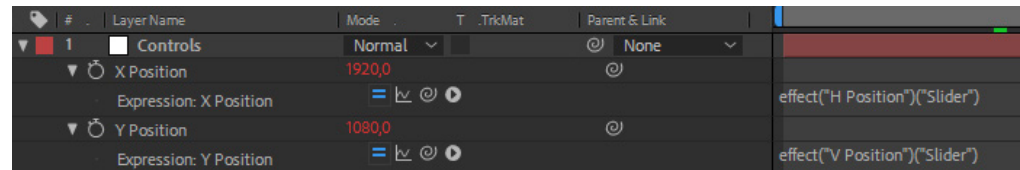


A much better approach is to use a Slider Control that's linked to Position with expressions. This lets you limit the range of the slider, making it impossible for the user to place the layer outside the visible or legal area.

Right-click on Position and choose Separate Dimensions to change from the two-dimensional Position to one-dimensional values for X and Y Position.



Add expressions to X Position and Y Position that link the position values to sliders.



The preferred way to add positioning controls into the Essential Graphics panel is to separate position dimensions and connect the position values to sliders with expressions.

Sliders are incredibly flexible in MOGRT design. I've used them for so many things that it's impossible to mention them all. Here's a short list of things they can do. We'll take a closer look at some examples later.

- Position layers
- Control animation start points and end points
- Restrict choices between predetermined positions
- Choose between predetermined color schemas
- Choose between several logos
- Control the number of decimals in numbers
- Adjust the padding in a text box
- Adjust the timing of an animation
- Choose the number of bars in a bar graph
- Control the duration before a layer is animated out
- Control the amount of distortion in a "hologram" title
- Choose between animation types
- Turn groups of layers on and off

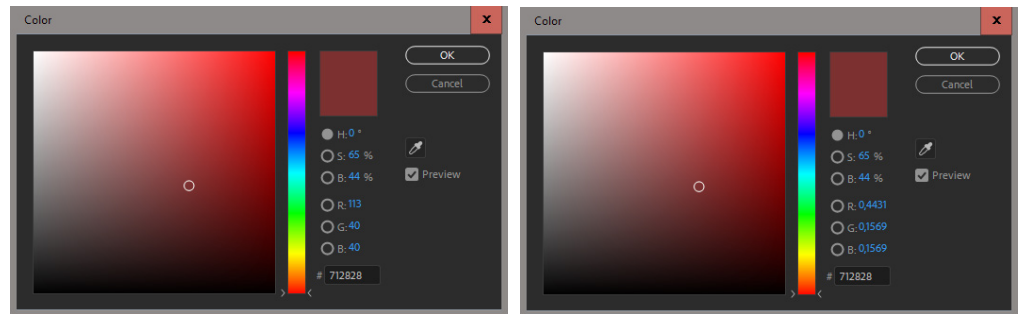
Color Control

Another Expression Control that we use a lot for MOGRT design is the Color Control. As you have guessed, it can control the color of any parameter in the timeline that has a color.



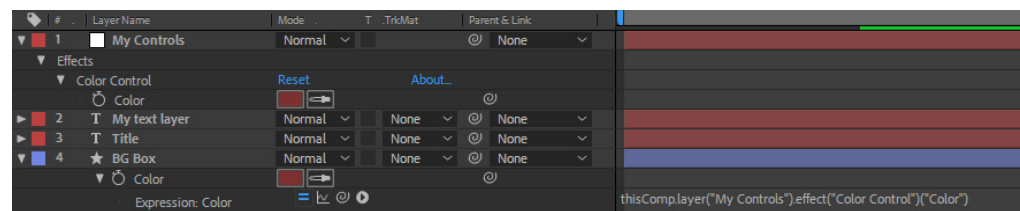
It's possible to control colors directly using expressions with arrays (for R, G, B, and Alpha), but since the values will have a very different meaning depending on the project bit depth (8-bit color values go from 0 to 255, 32-bit values go from 0 to 1), it's not a particularly safe way to control colors.

The color values change depending on your project's bit-depth settings.



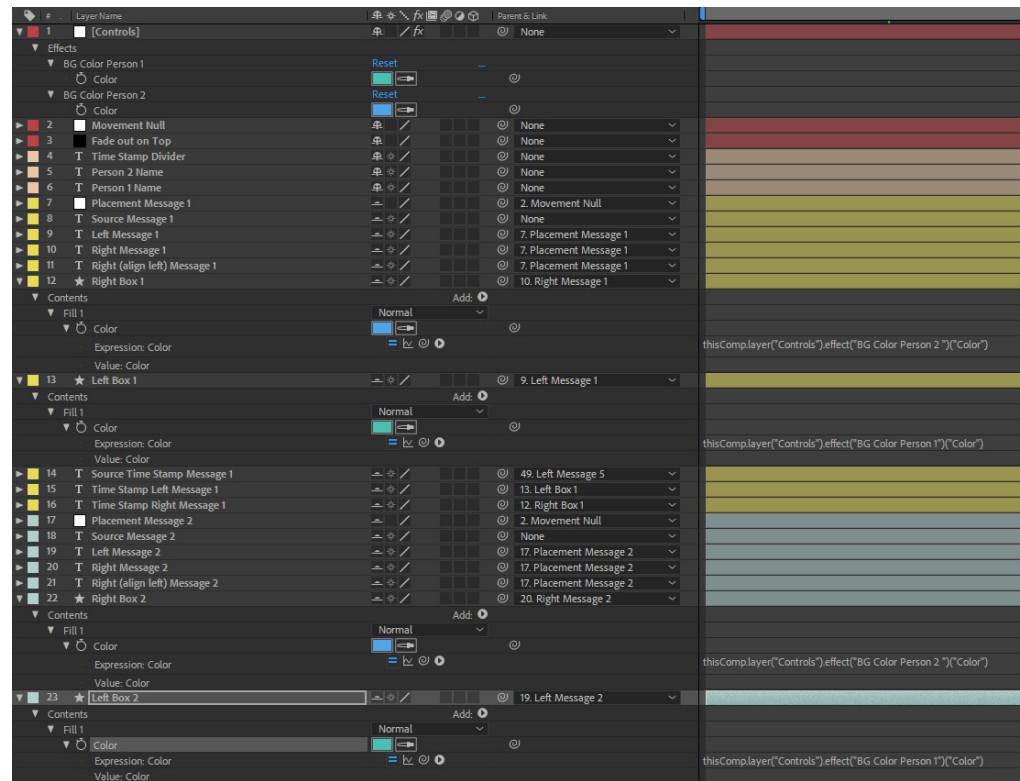
So, a better way to adjust colors is via Color Controls. If you're only going to change the color of one layer, then there's no need to use a Color Control. Just add the Color parameter, often named *color* or *fill color*, to the Essential Graphics panel. But if you want to change the color of 12 bars in a bar graph, then it's better to use one Color Control and link the fill color of all the bar layers to that control.

Here, the Color parameter of the fill color for the shape layer (BG Box) is linked to the Color Control on the Controls layer.



The more layers you have with color choices, the more sense it makes to link them to Color Controls instead of changing them one by one. Here's a 106-layer comp where linking the color choices to Color Controls was the only logical thing to do.

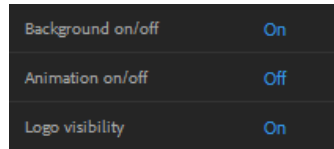
The first 23 layers in a 106-layer template: 20 of them link to the same two Color Controls.



Checkbox Control

These are great when you want to give the MOGRT user the ability to quickly turn things on or off. Unlike Slider Controls, where you can define as many options as you want, a Checkbox Control can only be on or off.

Checkbox Controls

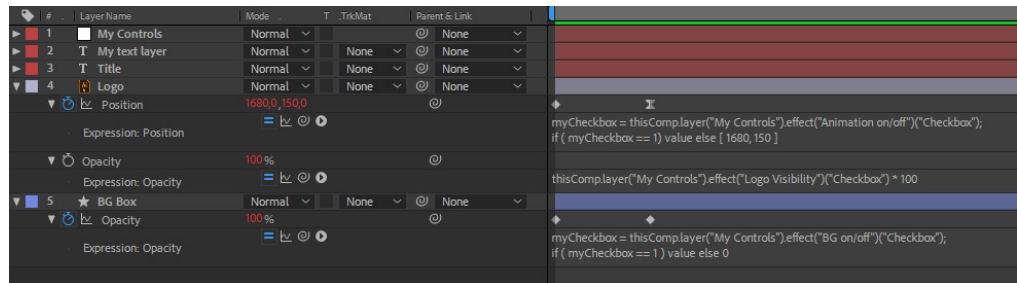


A Checkbox Control returns 0 or 1. This means we can use both math and if/else statements to describe what the checkbox should do.

To turn layers on and off, the **checkbox value times 100** is often used for Opacity.

Here's an example where I've used three Checkbox Controls to turn animation on and off for a logo, to turn the logo's visibility on and off, and to show the background or not.

Three Checkbox Controls used in different ways.



Here's a breakdown of the code for each expression. First, we'll take a look at the expression on Position for the logo layer.

```
myCheckbox = thisComp.layer("My Controls").effect("Animation on/off")("Checkbox");
if ( myCheckbox == 1) value; else [ 1680, 150 ]
```

It starts by defining the variable *myCheckbox* and says that it means the Animation on/off checkbox value. Then comes an if/else statement that says to use the original position, animated with keyframes if the checkbox is on, but to keep the position at 1680 horizontally, 150 vertically if the checkbox is off.

The result is that the logo stays still when Animation is set to off, and animates in when Animation is set to on. We'll dive deeper into if/else expressions shortly, so if you don't fully understand it now, that's totally okay.

Now let's take a look at the opacity expression for the same layer.

```
thisComp.layer("My Controls").effect("Logo Visibility")("Checkbox")
* 100
```

This simply multiplies the checkbox value by 100. When it's off, the result is 0. When it's on, the result is 100. So, the layer is only visible when the checkbox is on.

So, why didn't I use the same method for the opacity expression on the background box layer?

Because it's already animated with Opacity keyframes. This animation (fade in and out) would be lost if I used the checkbox value times 100. Therefore, I used an if/else statement instead.

```
myCheckbox = thisComp.layer("My Controls").effect("BG on/off")("Checkbox");  
if ( myCheckbox == 1 ) value; else 0
```

If the checkbox is on, this expression returns the Opacity value it would have without the expression, and it sets Opacity to 0 when the checkbox is off.

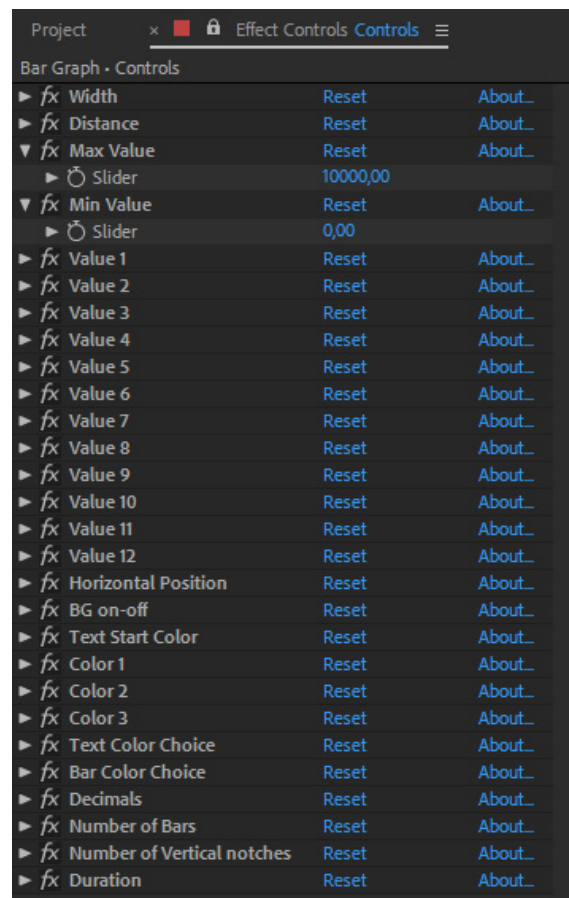
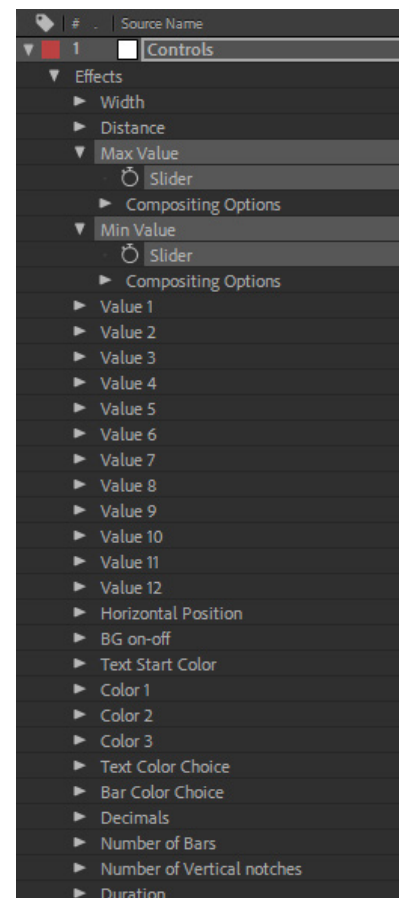
This gives you a bit of insight into the flexibility of Checkbox Controls. I think you'll use a lot of Checkbox Controls in your MOGRTs.

Use a Controls layer

It's very common to use a null object to hold these Expression Controls, and it's also common to name it "Controls", as I've done in the examples you've seen. Keeping the controls on a separate layer helps when you start adding them to the Essential Graphics panel. Just select the ones you need and use your custom keyboard shortcut for *Add Property to Essential Graphics*.

You can point to sliders in the Effect Controls panel with the pick whips. Note the padlock icon in the Effect Controls panel in the figure that shows that I've locked the panel to the Controls layer. This means that I can always pick whip to these properties.

Using a null object to hold all the controls lets you work with the Effect Controls for that layer showing at all times so you can pick whip to sliders and other controls.



Most useful expressions for MOGRT design

Different designers use very different approaches for achieving the same thing, so consider this chapter the author's take on which expressions are the most useful. I did ask around a bit when writing this book, though, and it seems that many designers agree with me.

The linear and ease expressions

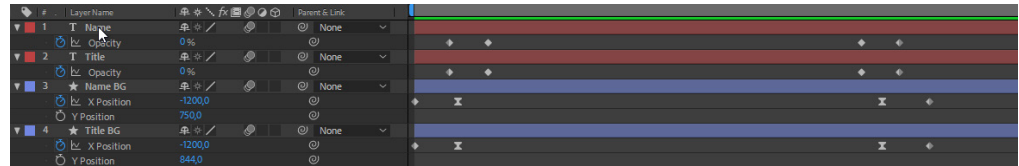
These are fantastic for MOGRT creators, especially for timing control and for transforming values. The syntax may seem a bit strange when you see it for the first time, but it's really simple when you get used to it.

You can find the linear and ease expressions in the Expression Language menu.

Interpolation	>	linear(t, value1, value2)
Color Conversion	>	linear(t, tMin, tMax, value1, value2)
Other Math	>	ease(t, value1, value2)
JavaScript Math	>	ease(t, tMin, tMax, value1, value2)
Comp	>	easeIn(t, value1, value2)
Footage	>	easeIn(t, tMin, tMax, value1, value2)
Layer	>	easeOut(t, value1, value2)
Camera	>	easeOut(t, tMin, tMax, value1, value2)

Let's look at some examples. I use linear and ease a lot to enable the user to control the timing of the out-animation. If you use keyframes to animate out a text background (a shape layer) in a lower third after 4 seconds, the user cannot control that. It will always animate out after 4 seconds.

If you use keyframes to animate the layers out, the user has no control over the duration.

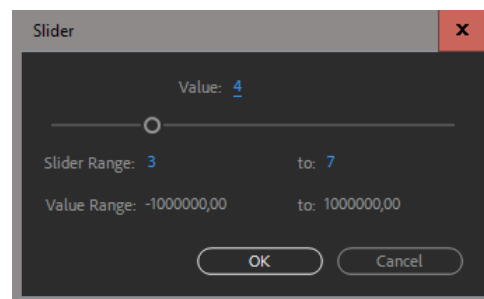


Controlling out-animation with linear and ease

The MOGRT would be much more user-friendly if the user could control the duration with a slider, so let's change the design to make that happen. Add a new null object (**Layer > New > Null Object**) and name it *Controls*. Then add a Slider Control to it and rename the slider to *Duration*.

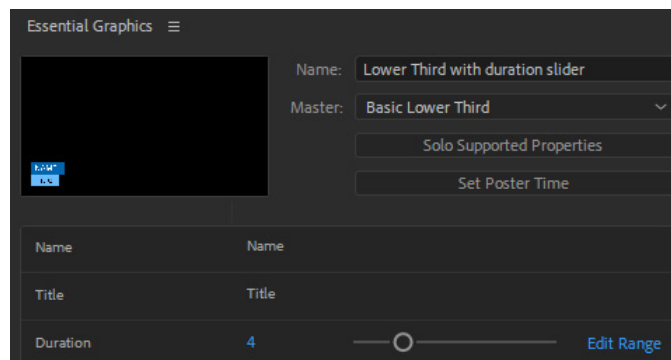
Drag the slider to the Essential Graphics panel, and click the Edit Range button.

These settings will let the user choose a duration between 3 and 7 seconds when we add some expressions.



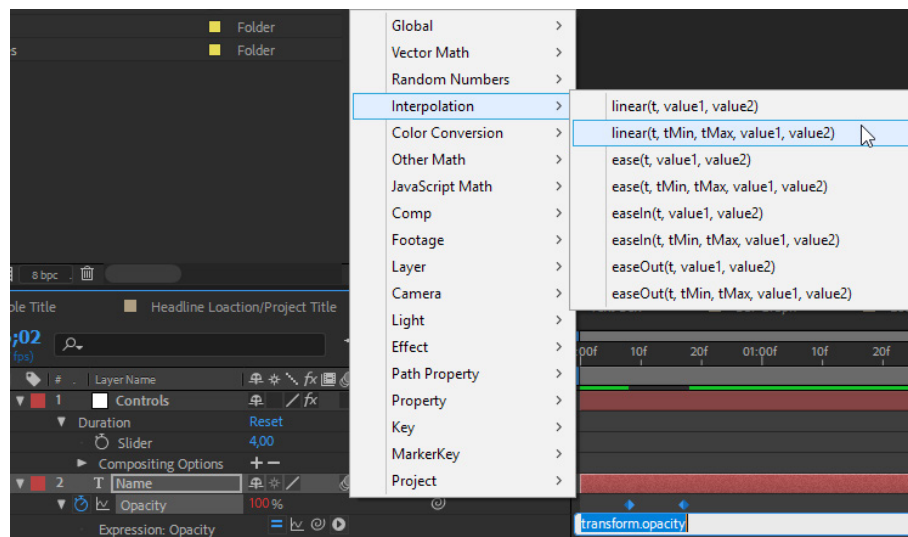
Change the range so it goes from 3 to 7 and set the value to 4. The Essential Graphics panel is still very tidy, so the MOGRT will be very easy to use when it's finished.

A tidy Essential Graphics panel is a good sign that the MOGRT will be easy to use.



We'll animate the *Name* and *Title* layers out with a simple Opacity change. **Alt-click** the Opacity stopwatch and then click the Expression Language menu button and navigate to the second linear expression.

Use the Expression Language menu to navigate to the Interpolation category and select the second linear expression.



This adds the code to the expression text box. Add a line break before the existing code and write **t =**. Then point the Expression Pick Whip to the Duration slider on the Controls layer. Add a semicolon to the end of the line. Your expression should now look like this:

```
t = thisComp.layer("Controls").effect("Duration")("Slider");
linear(t, tMin, tMax, value1, value2)
```

The best part is that we haven't written much code ourselves. We defined the variable named *t*. After Effects did the rest. Nice! But our expression doesn't do anything yet, apart from throwing an error message if you click outside the expression text box. Let's do the necessary changes to the second line.

```
t = thisComp.layer("Controls").effect("Duration")("Slider");
linear (time, t-0.6, t-0.3, 100, 0)
```

I'll try to explain what this code means by now. Since it's on the Opacity parameter for the layer it says, "As time changes from 0.6 seconds before the slider value to 0.3 seconds before the slider value, linearly change Opacity from 100 to 0." So if the slider is set to 4, the Opacity will change from 100 to 0 between 3.4 seconds (4-0.6) to 3.7 seconds (4-0.3).

You may be wondering about the strange numbers. It's because I want this layer to start fading out before the background layer starts moving out of the frame.

In a linear expression, the first argument inside the parentheses is telling After Effects which property to look for changes in. In our case, that's time, which means the composition's time, measured in seconds.

The next two entries describe the changes in the property that After Effects should care about. The next two entries tell After Effects how the output value should change when that parameter changes. Read this paragraph a few times, and it will make sense, eventually.

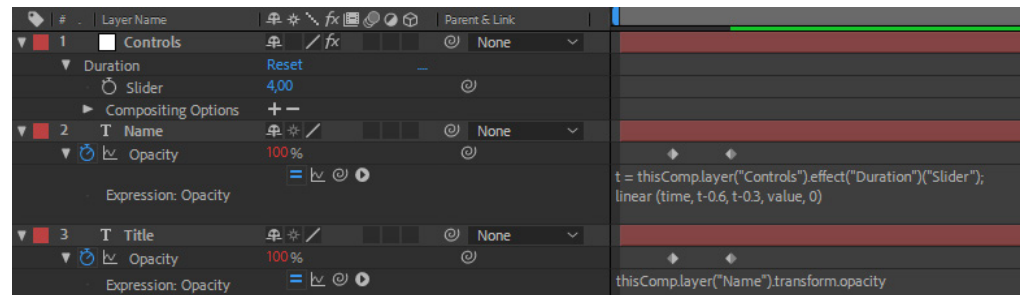
If you play back the timeline now, you'll see that the in-amination on that layer no longer works. The expression overrides the keyframed values. But we can tell After Effects to use the existing value until it's time for the expression to do its thing. We do this by typing *value* instead of *100* in the expression.

```
t = thisComp.layer("Controls").effect("Duration")("Slider");
linear (time, t-0.6, t-0.3, value, 0)
```

This expression says something like this: *"As time changes from 0.6 seconds before the slider value to 0.3 seconds before the slider value, linearly change Opacity from whatever is to 0."*

We could copy this expression to the Opacity of the *Title* layer, but that means we'll have to change the expression of both layers if we want to change the timing. It's a better idea to just link the Opacity of the *Title* layer to the Opacity of the *Name* layer using the *Property Link Pick Whip*. This means we only need to change the code in one place to change the timing.

The two text layers with their new expressions.



Let's try to copy the code from the *Name* layer to *X Position* on the *Name BG* layer and see what happens. It's not perfect, but it's a good start. There are two things we need to fix: This animation should start and end a bit later than the Opacity change on the other layer, and the position at the end is set to 0, which is not what we wanted.

Let's fix both with a few changes in the code. Here's my code.

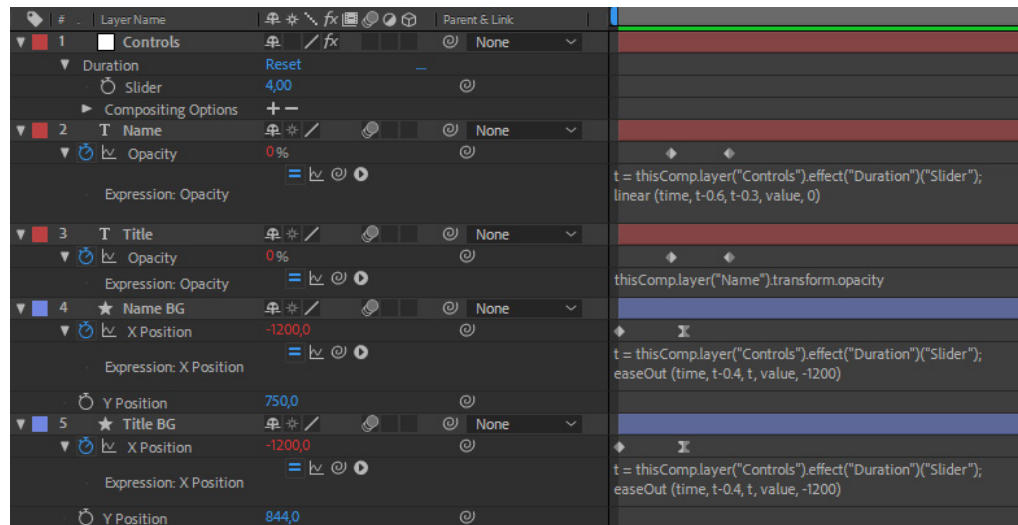
```
t = thisComp.layer("Controls").effect("Duration")("Slider");
linear (time, t-0.4, t, value, -1200)
```

This makes the animation start when the text layers have almost faded out (0.4 seconds before the end) and end at the time set by the Duration slider.

The X Position will also end up at -1200, which is well outside the frame, even for long names. This looks good so far, but the out-animation would look better with an ease-out instead of a linear movement. This is easy: Just change the word *linear* to *easeOut*.

```
t = thisComp.layer("Controls").effect("Duration")("Slider");
easeOut (time, t-0.4, t, value, -1200)
```

The timeline after our changes.



It took a lot of words to explain this, so it might seem like a lot of work, but in reality, it's very quick. We did a lot of copy and paste, and the pick whips wrote most of the code. You'll probably use the **value** trick a lot instead of typing hard-coded numbers. It got us out of trouble this time by avoiding conflicts with existing keyframed animations.

We could also have used an if/else expression that tells After Effects to use the existing values before 2 seconds and to use the values from the expression after 2 seconds, but that means we'd have to write more advanced code.

Another method for avoiding conflict with existing keyframed values is to use an extra Transform effect and do the in- or out-animation there.

if/else expressions

Combined with sliders, checkboxes, and so on, the if/else expression lets you make highly flexible and user-friendly MOGRTs. You can decide what should happen when a checkbox is on or off, and what should happen when a slider is set to 2, or 4, or 17.

Read about the differences

between the two expressions engines here:

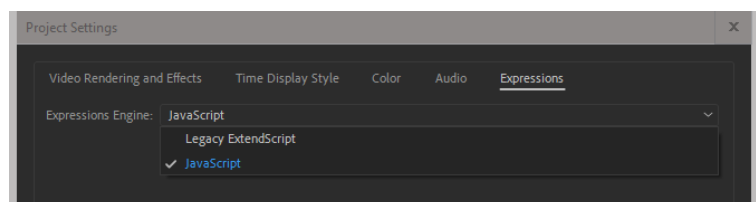
<https://premierepro.net/engines/>

Choose which Expressions engine to use

After Effects 16 (the October 2018 release) uses a faster JavaScript engine for expressions than older versions do. Some old if/else expressions will not work in the new engine, so beware if you're copy/pasting expressions from older websites and training.

There's a good reason we got a new engine: Expressions can render up to 5–6 times faster than with the old (ExtendScript) engine. In addition, the new engine offers higher mathematical precision in floating-point operations and lets you use array and string methods from modern JavaScript. <https://www.ecma-international.org/publications/standards/Ecma-262.htm>

New projects will default the Expressions Engine option to the new JavaScript engine. Projects saved in previous versions of After Effects will default to the legacy ExtendScript.



When saving a project that uses the new engine to earlier versions of After Effects, you may also cause some trouble. Use the old engine if you know that you must save to older versions.

For backwards compatibility, you can choose whether a project uses the new JavaScript engine or legacy ExtendScript: In the Project Settings dialog box, click on the Expressions tab, and change the Expressions Engine option.

For the nerds:

On Windows, After Effects now uses the V8 open source JavaScript engine. https://en.wikipedia.org/wiki/Chrome_V8

On macOS, it uses the JavaScriptCore framework. <https://developer.apple.com/documentation/javascriptcore?t=201812>

if/else statements in the new JavaScript engine

The syntax for if/else statements is stricter in the new JavaScript engine, and they need to be written for standardized JavaScript. The old engine would accept the following short code for if/else without any errors:

```
if (colorChoice == 3) 100 else 0
```

The new engine would throw an error for that code. You need to add the curly brackets or add a semicolon or a line break before the word *else*.

```
if (colorChoice == 3) {100} else {0}  
if (colorChoice == 3) 100  
else 0  
if (colorChoice == 3) 100; else 0
```

If you're a JavaScript expert, you may also know how to use the ternary operator. That will also work. The question mark means if, and the colon means else.

```
(colorChoice == 3) ? 100 : 0
```

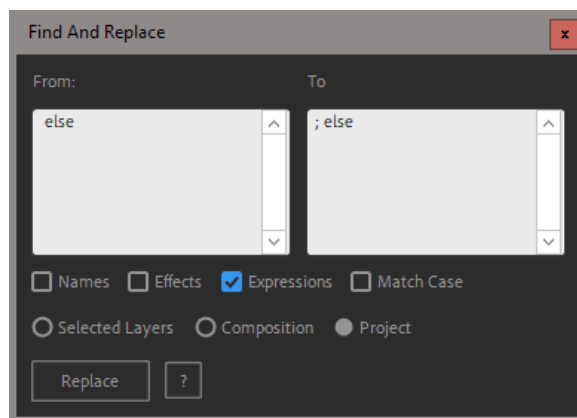
As you may see, the syntax goes like this:

```
condition ? value if true : value if false
```

Sorry, I got carried away. Anyway, if you open an old project and change the expressions engine to the new JavaScript one, some old if/else statements may need a change. The fastest way would be to use the *Find And Replace* script from Videohive.net (not to be confused with the *Find And Replace Text* script) to change all instances of " else " to "; else ".

<https://videohive.net/item/find-and-replace-script/10855153>

The Find And Replace script is perfect for changing all instances of the old code in one go.



Character Index in the new JavaScript engine

When accessing the indexes of characters on a text layer as an array, use `text.sourceText.value[i]` instead of `text.sourceText[i]`.

if/else used with Checkbox Controls

We've already used if/else to control the Opacity of a text background (a shape layer) in the discussion of Checkbox Controls.

```
myCheckbox = thisComp.layer("Controls").effect("BG on/off")("Checkbox");  
if ( myCheckbox == 1 ) value; else 0
```

This code first defines the variable *myCheckbox* as the value of the checkbox named *BG on/off* on the *Controls* layer. Then comes the actual if/else statement.

```
if ( myCheckbox == 1 ) value; else 0
```

This code says that if the checkbox value is 1, the Opacity should be set to whatever value it already has. But if the checkbox does not output 1, the Opacity should be set to 0. The structure is like this:

- You start by typing the JavaScript keyword **if**.
- Then you put the condition you want to evaluate inside parentheses.
- After the closing parenthesis you tell the expression what to output if the argument is true.
- Then you add a semicolon.
- Then you type the JavaScript keyword **else**.
- Lastly, you tell the expression what to do if the argument is not true.

In this example, we wanted to check if the checkbox value is 1 or not. So we wrote **value** after the closing parentheses to tell After Effects to output the existing value if the checkbox outputs 1.

Then after the word *else*, we wrote **0**. This tells After Effects to set the Opacity to 0 if the checkbox outputs something other than 1.

Here's another example that you can use on the Source Text parameter of a text layer.

```
Norwegian = effect("Norwegian")("Checkbox");  
if (Norwegian ==1) "Hei"; else "Hello"
```

This will output the Norwegian word *Hei* if the checkbox is on, and *Hello* if it's off. I like to write my if/else statements this way because it's compact, and I find it easy to read. But if you read about JavaScript coding, you'll see that many programmers recommend that you write the code like this, with curly brackets.

```
If ( Norwegian == true ) {"Hei"} else {"Hello"}
```

Some go even further and tell you that you should also insert line breaks to make the code easier to read. This is definitely the proper way to write if/else code, and I use it on more complex if/else statements; especially if I need more than one if/else statement in one expression.

```
if ( Norwegian == true ) {  
  "Hei"  
} else {  
  "Hello"  
}
```

Logical operators

There are more ways to compare than the double equals sign, of course.

Different ways to compare values.

Operator	Description	Example Code for Comparing
==	equal to	If (x == 4)
!=	not equal	If (x != 4)
>	greater than	If (x > 4)
<	less than	If (x < 4)
>=	greater than or equal to	If (x >= 4)
<=	less than or equal to	If (x <= 4)

Note that "equal to" uses two equals signs. It's a JavaScript thing again. It is a very common mistake to use a single equals character as the equality operator. This can give you strange results.

You can use logical operators to combine two or more statements in one if/else expression. The operators you can use are logical "and" (**&&**), logical "or" (**||**), and logical "not" (**!**).

Here's a logical operator example from Dan Ebberts that uses the logical "or" in an expression on a layer's Opacity.

```
if (rotation < 90 || rotation > 270) 100; else 0
```

It basically says that *"if the rotation is less than 90 or more than 270, set Opacity to 100. If it's not, set Opacity to 0."* The logical "and" works pretty much the same way.

if/else used with Slider Controls

The following expression uses if/else to make the layer visible (Opacity set to 100) only if the Color Choice slider is set to 3 and the Background checkbox is set to on. Both conditions must be met.

```
myCheckbox = thisComp.layer("Controls").effect("BG on/off")("Checkbox");
colorChoice = thisComp.layer("Controls").effect("colorChoice")("Slider");
if ( myCheckbox == 1 && colorChoice == 3) 100; else 0
```

You can use multiple logical operators. This expression on Opacity sets Opacity to 100 if the color choice slider is set to 3, **and** the layer's Rotation is less than 90 **or** more than 270.

```
colorChoice = thisComp.layer("Controls").effect("colorChoice")("Slider");
if ( (rotation<90 || rotation>270) && (colorChoice == 3) ) 100; else 0
```

I recommend that you use parentheses to separate the conditions, although there are rules for which operators are evaluated first, so sometimes the extra parentheses are not required.

When you need to choose between multiple conditions, you can use **else if** statements. It's an extension of the if/else statement, and the extra lines of code start with *else if* instead of just *else*. The first line (after defining variables) still starts with a standard *if*, and the last conditional argument starts with a normal *else*.

Only the extra lines start with *else if*. Here's some code from a MOGRT where I've used logical operators and multiple *else if* statements to place a layer according to the value of two sliders. I used it on the Position parameter, of course. Note the placement of line breaks between the opening and closing curly brackets. This is not an error—it's done to make the lines with the position numbers in brackets cleaner and easier to change.

```
Pos = Math.round(thisComp.layer("Controls").effect("Position")("Slider"));
AR = Math.round(thisComp.layer("Controls").effect("Aspect Ratio")("Slider"));

if ((AR == 1) && (Pos == 1)){
[100,100]
} else if ((AR == 1) && (Pos == 2)) {
[1820,100]
} else if ((AR == 1) && (Pos == 3)) {
[100,980]
} else if ((AR == 1) && (Pos == 4)) {
[1820,980]
} else {
[960,540]
```

Yeah, **Math.round** is a JavaScript thing again, and it rounds the value to the nearest whole number (integer). We'll come back to this. There are other ways to do the same as multiple *else if* statements. Read about the switch statement in a JavaScript reference to see if you like that method better.

If/else shorthand

It's possible to write more compact code for if/else statements with the conditional operator, commonly known as the ternary conditional operator. Especially for longer expressions, it can help improve code readability.

Nick Howell explains: *You put a condition in the parenthesis first, then a ? mark to show you're making this a ternary operator, then the next line of code will be executed if true. The : works as an else.* Here's the syntax with the ternary operator:

```
(condition) ? value if true : value if false
```

That short code equals this standard if/else code.

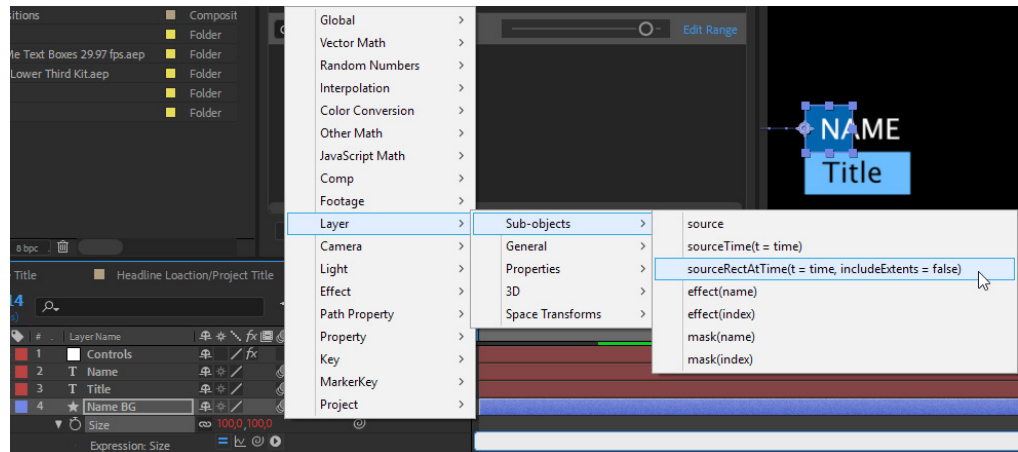
```
if (condition) {  
  value if true;  
} else {  
  value if false;  
}
```

So, if you see some question marks and colons in expressions forums online, now you know what they are. If you want to know more about using the ternary operator, [here's a good explanation from CodeBurst](#).

sourceRectAtTime

This expression measures the width and height of text layers and shape layers. It's very handy for making a shape layer adapt to the size of a text layer and for accurate and automatic placement of layers even when text and shapes are changed. It appears in almost every advanced MOGRT out there.

You'll find the code in the Expression Language menu.



You can use the Expression Language menu to see how the code is written, but I generally just write a simpler version of the code by hand or copy it from my expressions collection text document.

Measuring the width and height of text or shape layers

Normally you're interested in getting the width or the height.

```
sourceRectAtTime().width  
sourceRectAtTime().height
```

This omits the stuff inside the parentheses that you'd get from the Expression Language menu.

The code inside the parentheses tells After Effects what time to measure the size and if extents should be included.

```
sourceRectAtTime(t = time, includeExtents = false)    // full code
sourceRectAtTime(time, false)                        // (shorter code)
sourceRectAtTime()                                   // (even shorter code)
```

If you delete the text inside the parentheses, After Effects will use the current time, and extents will not be included. To include extents, you'll have to specify it in the parentheses.

```
sourceRectAtTime(time, true)
```

But what exactly are these "extents"?

For a shape layer, including extents will include strokes, repeaters, and other extra stuff, plus some room for anti-aliasing.

A standard text layer (point text) does not have any extents.

For a paragraph text layer, including extents will measure the bounding box, instead of only the bounds of the text's visible pixels.

Some old MOGRTs where paragraph text is used may behave differently in version 15.1.0 and newer versions.

Please note that support for extents in paragraph text was added in the 15.1.0 (CC 2018.1.0) version of After Effects. In older versions, only shape layers have extents.

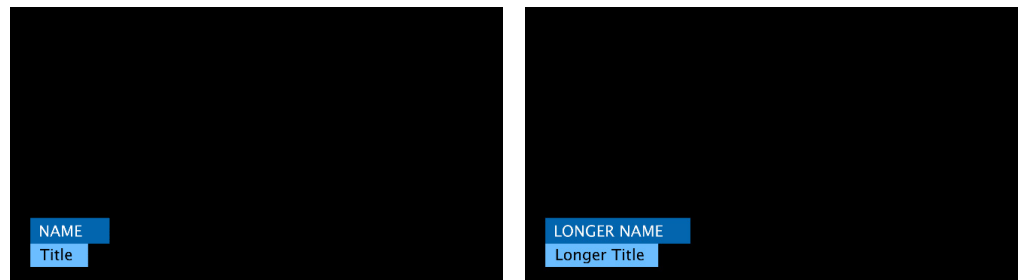
This change created problems for some users who had used "sourceRectAtTime(time, true)" for paragraph text in older versions, where the extra code didn't do anything. In newer versions of Premiere Pro, the MOGRTs suddenly behaved very differently, and users had to delete that extra code in their expressions and export the MOGRTs again.

Adaptive background layers

Here's an example that I often use on the Size parameter of a rectangle in a shape layer. It finds the width of a text layer named *Name* and adds 100 pixels to it. Then I use that for the width of the rectangle and set the height to 100 pixels.

```
x = thisComp.layer("Name").sourceRectAtTime().width + 100;
y = 100;
[x, y]
```

The background shape layers adapt to the length of the text.



I could, of course, have linked the rectangle height to the height of the text layer too.

```
x = thisComp.layer("Name").sourceRectAtTime().width + 100;
y = thisComp.layer("Name").sourceRectAtTime().height + 50;
[x, y]
```

But this sometimes causes problems where the height changes, depending on if the text has ascenders and descenders (É, Õ, g, j, y, and so on) or not. Dan Ebberts has a clever hack, putting this code in the Source Text parameter of the text layer:

```
if (time < 0) "Ég"; else value
```

So, before the comp start time, the text will be Ég, which has both ascenders and descenders. Then he modifies the code on the Size parameter to something like this, so it looks at the size at minus one second:

```
y = thisComp.layer("Name").sourceRectAtTime(-1,false).height + 50;
```

This works beautifully unless you are scaling the text from 0 to 100 at the start, in which case the text height would be 0 at minus one second. You could, of course, add another expression that sets the scale to 100 before the comp start time, but it gets more complicated.

Use Slider Controls with sourceRectAtTime

To make the template more flexible, we could also link these expressions to sliders that control the horizontal padding and the box height. This way, users can drag the sliders in Premiere Pro until they like the padding they see.

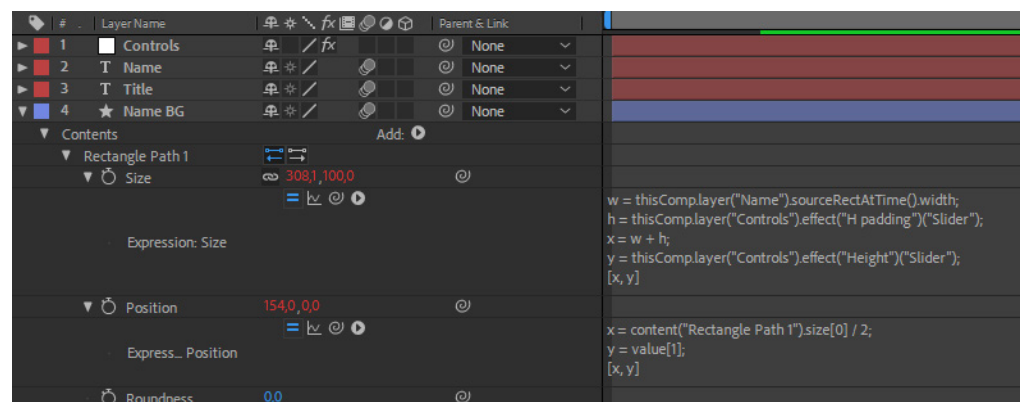
```
w = thisComp.layer("Name").sourceRectAtTime().width;
h = thisComp.layer("Controls").effect("H padding")("Slider");
x = w + h;
y = thisComp.layer("Controls").effect("Height")("Slider");
[x, y]
```

Scale a rectangle from the left

Rectangles in shape layers scale from the middle by default. Use this expression to make a rectangle scale from its left edge. It moves the position so it's always half of the width, and the result is that the left edge stays still.

```
x = content("Rectangle Path 1").size[0] / 2;
y = value[1];
[x, y]
```

This is a very common combination of expressions for Size and Position of a rectangle when the size is controlled by a text layer.



To make it scale from the right edge, just change the number 2 to negative 2.

```
x = content("Rectangle Path 1").size[0] / -2;
y = value[1];
[x, y]
```

Finding the edges of text and shapes

You'll often need to find the top, left, bottom, and right edges of a shape layer or text layer. There are special code names for the top and left edges, but to get to the right edge and the bottom, you'll have to do some math.

```
Top edge = top
Left edge = left
Right edge = left + width
Bottom edge = top + height
Vertical middle = top + height/2
Horizontal middle = left + width/2
```

Remember that this will give you the position of the edges of the rectangle internally in the shape layer. You must also take the position of the layer itself into account, like this:

```
rectLeft = thisComp.layer("Name BG").sourceRectAtTime().left;
rectTop = thisComp.layer("Name BG").sourceRectAtTime().top;
xPos = thisComp.layer("Name BG").transform.xPosition;
yPos = thisComp.layer("Name BG").transform.yPosition;
x = rectLeft + xPos - 50;
y = rectTop + yPos - 50;
[x, y]
```

Using this code on the Position for a logo layer will place it 50 pixels above and to the left of the top left corner of the text layer. Two new variables for height and width, plus a small change in the code for x and y, place the logo 50 pixels to the right of the rectangle.

```
rectHeight = thisComp.layer("Name BG").sourceRectAtTime().height;
rectWidth = thisComp.layer("Name BG").sourceRectAtTime().width;
rectLeft = thisComp.layer("Name BG").sourceRectAtTime().left;
rectTop = thisComp.layer("Name BG").sourceRectAtTime().top;
xPos = thisComp.layer("Name BG").transform.xPosition;
yPos = thisComp.layer("Name BG").transform.yPosition;
x = rectLeft + xPos + rectWidth + 80;
y = rectTop + yPos + (rectHeight/2);
[x, y]
```

An even better way to write this if you want to reuse the code is to define a variable for the name of the shape layer. This way, you only have to change the layer name once. But there's another problem that can possibly give you headaches: If you parent the shape layer with the rectangle to another layer, the whole thing breaks, since the position of a child is relative to its parent.

We'll fix this soon with the **toComp** expression, but before that I want to mention a couple of popular ways to use the **sourceRectAtTime** expression.

Center the anchor point of a text layer

When you create a text layer, the anchor point is at the baseline of the text. Sometimes, it's much more convenient to have the anchor point in the middle of the text for scaling and rotation purposes. The following expression uses **sourceRectAtTime** on the Anchor Point parameter to center the anchor point vertically.

```
y = value[1] - (thisLayer.sourceRectAtTime().height / 2);
[0, y]
```

Use **sourceRectAtTime** to center the anchor point of a text layer vertically.



You can control where the anchor point is horizontally with the Paragraph panel setting for Left Align, Right Align, and Center Align text.

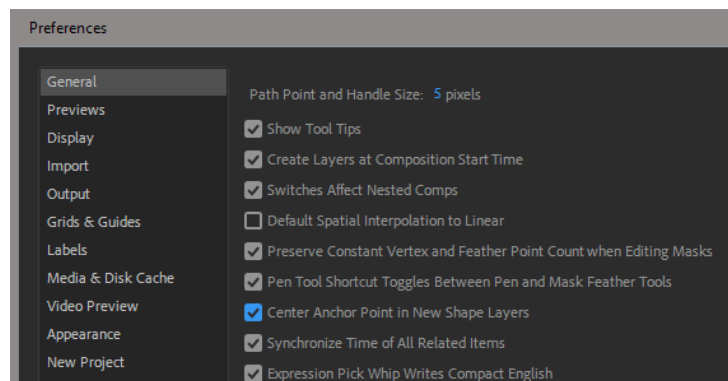
Place anchor point in middle of shape layer

It's sometimes nice to have the anchor point in the middle of the shape layer. Here's some code you can use on the anchor point of the shape layer to achieve this.

```
top = sourceRectAtTime().top;
left = sourceRectAtTime().left;
x = left + (sourceRectAtTime().width/2);
y = top + (sourceRectAtTime().height/2);
[x,y]
```

This works as expected only on shape layers with one shape, though. I also think this will be less useful in the future since the 15.1.0 (CC 2018.1.0) version of After Effects introduced a preference to make this happen automatically.

After Effects can center the anchor point of a shape layer by default.



Use toComp to fix parenting problems

As I mentioned previously, some expressions will break when you parent a layer, since the position is relative to the parent layer. The **toComp** expression will calculate the position relative to the comp, so the expression no longer breaks because of parenting.

If we change the expression that we used to place a logo layer in relation to a rectangle so that it uses **toComp** to find the x and y position of the layer, it will work across any amount of parenting.

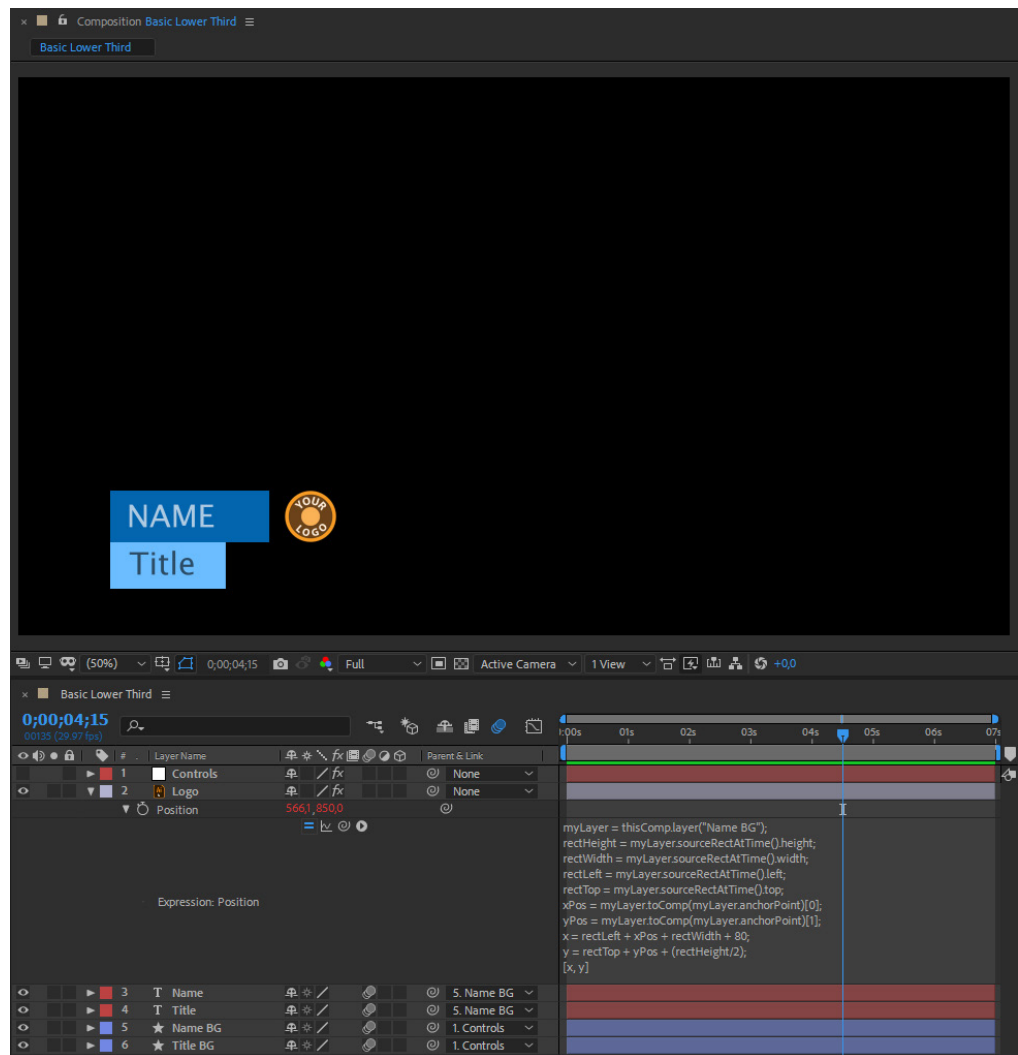
```
myLayer = thisComp.layer("Name BG");
rectHeight = myLayer.sourceRectAtTime().height;
rectWidth = myLayer.sourceRectAtTime().width;
rectLeft = myLayer.sourceRectAtTime().left;
rectTop = myLayer.sourceRectAtTime().top;
xPos = myLayer.toComp(myLayer.anchorPoint)[0];
yPos = myLayer.toComp(myLayer.anchorPoint)[1];
x = rectLeft + xPos + rectWidth + 80;
y = rectTop + yPos + (rectHeight/2);
[x, y]
```

Another expression does a similar thing, and that's **toWorld**. This is often used when working with 3D layers in After Effects.

In many cases toWorld and toComp will do the same thing, but if you use a camera in 3D space, toComp also takes the position of the camera into the equation.

You can read more about *Layer Space Transforms* in the [Adobe expression language reference](#).

This code may be a bit of an overkill for achieving this. But since it measures every aspect of the shape layer, we can reuse most of the code no matter where we want to place a layer in relation to the text.

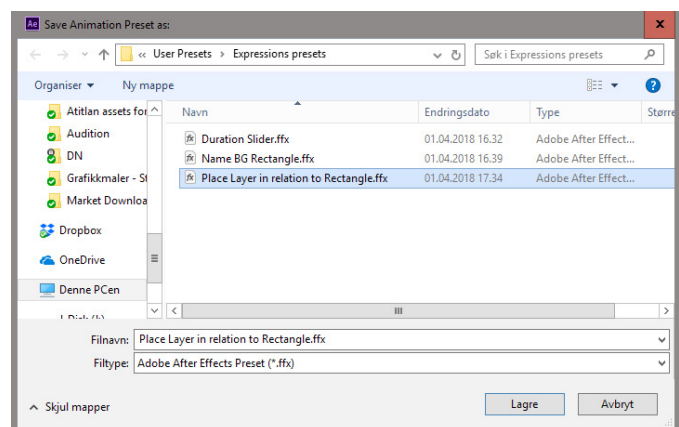


Save your expressions as part of an animation preset.

Saving your expressions for easy access

You can save this code in your favorite expressions text document and save a lot of time the next time you need to place a layer in relation to a rectangle.

You can even select *Position* on the logo layer and choose *Save Animation Preset* in the menu in the Effects & Presets panel, so you can use it as an effect preset. This is a great way to store your favorite expressions.



I have saved a lot of things I use regularly in this way. If you find yourself creating many of the same sliders, checkboxes, and color controls every time you design a new MOGRT, why not save all those as an effect preset so you can add them to your null object in one go? They will have the correct names, sliders will be set to useful ranges, and so on.

Bounce & Overshoot and Inertial Bounce

Probably one of the most used (maybe overused?) code snippets from the web is [Dan Ebberts' Keyframe Overshoot expression](#) from MotionScript.com. It can be used on almost any keyframed parameter. Animate Position, Scale, Rotation, and so on with linear keyframes, and add this expression to the parameter.

```
freq = 3;
decay = 5;

n = 0;
if (numKeys > 0){
  n = nearestKey(time).index;
  if (key(n).time > time) n--;
}
if (n > 0){
  t = time - key(n).time;
  amp = velocityAtTime(key(n).time - .001);
  w = freq*Math.PI*2;
  value + amp*(Math.sin(t*w)/Math.exp(decay*t)/w);
}else
  Value
```

Your animation no longer stops in a dull and uninteresting way, but bounces a bit before it stops. A more elaborate, and more customizable, version exists, but it's mostly located in forums, so it's harder to find. But I like this one better. Anyway, here it is.

This is also Dan Ebberts' code.

```
amp = .05;
freq = 4.0;
decay = 8.0;

n = 0;
if (numKeys > 0){
  n = nearestKey(time).index;
  if (key(n).time > time){
    n--;
  }
}
if (n == 0){
  t = 0;
}else{
  t = time - key(n).time;
}

if (n > 0 && t < 1){
  v = velocityAtTime(key(n).time - thisComp.frameDuration/10);
  value + v*amp*Math.sin(freq*t*2*Math.PI)/Math.exp(decay*t);
}else{
  value;
}
```

Amp is amplitude (think intensity), **freq** is bounces per second, and **decay** is the speed of decay. A low value for decay makes it bounce for a long time; a high value makes it stop faster.

Custom ease expressions

There are many online tutorials on writing code for easing. This one from Robert Penner is a well-known one: <http://www.gizma.com/easing/>

This one from Jari Komppa is also referred to often: <http://sol.gfxile.net/interpolation/>

[Nick Howell](#) makes unique and user-friendly MOGRTs, and he has a passion for easing. He's also a programmer, so he decided to write his own expression for easing. He has kindly agreed to share it in this book, so here we go. Use the following code on the Position parameter of your layer.

```
// Written by Nickolas Howell

function easeExpo(beginValue,endValue,duration,offset) {
  t=time-offset;
  if (t < 0) return beginValue
  if (t > duration) return endValue;
  if (beginValue.constructor == Array) {
    x=0;
    y=0;
    z=0;
    if (beginValue.length == 2) {
      x=easeExpo(beginValue[0],endValue[0],duration,offset);
      y=easeExpo(beginValue[1],endValue[1],duration,offset);return[x,y]
    } else {
      x=easeExpo(beginValue[0],endValue[0],duration,offset);
      y=easeExpo(beginValue[1],endValue[1],duration,offset);
      z=easeExpo(beginValue[2],endValue[2],duration,offset);
      return[x,y,z]
    }
  }

  c= endValue - beginValue;
  t /= duration/2;
  if (t < 1) return c/2*Math.pow(2,10*(t-1)) + beginValue;
  t--;
  return c/2*(-Math.pow(2,-10*t)+2)+beginValue
};

if (time <= 1) { easeExpo([160, 400], [1760, 400], 1, 0); } else { ease-
Expo([1760, 400], [160, 400], 1, 1); }
```

Change the numbers in the last line, and see how the movement changes. You can link all these to sliders, of course, so they're easier to use.

parseFloat

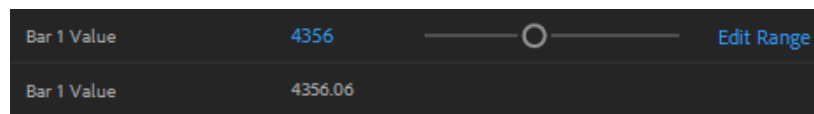
This expression is used for extracting numbers from text strings and will return a floating-point number. So, if the text in a layer named *Bar 1 value* is a number, this expression will convert that text to a floating-point number.

```
myText = thisComp.layer("Bar 1 Value").text.sourceText;
parseFloat(myText)
```

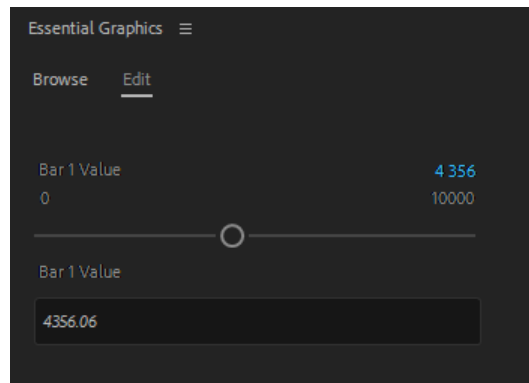
If the user types 4356.06 in the text field, the output of this code will be the number 4356.06. This means you can let the user enter numbers in text fields in a MOGRT, instead of using sliders.

We use this method because sliders are not very accurate when their range is wide. It's also more intuitive for most people to enter a number as text than to click a blue number on a slider. Plus, a text field will show all the decimals in the Essential Graphics panel, while a slider will not.

Slider and text field
in After Effects.



Slider and text field in Premiere Pro. Note that the slider value is not showing the decimals, while the text field is.



There are a few caveats with this approach, though. It fails if the user doesn't type a number, but something else instead, and it fails if the user is from a country where a comma is used as a decimal separator instead of a period, like in Europe outside of the UK.

We will take a look at error catching and foolproofing things like this later.

Rounding numbers

Sometimes you'll need to round numbers to make your MOGRTs work. Say you want to use a Slider Control to let the user choose between four different logos. This can be done by using an expression that sets Opacity to 0 or 100 on the four logo layers according to the value of the slider.

You could use a collection of *larger than* and *lower than* operators, but it's easier to just use the *equals* operator.

```
mySlider = thisComp.layer("Controls").effect("LogoChoice")("Slider");  
if (mySlider == 1) value else 0
```

The code would be the same for all four logo layers, except that *mySlider == 1* would change to *mySlider == 2*, *mySlider == 3*, and *mySlider == 4*.

Rounding values to whole numbers

The problem with this code is that it will only work if the user manages to drag the slider to exactly 1, 2, 3, and 4. Since the sliders are not very accurate, that's not very likely. So, we need to round the values. For this, we'll use the **Math.round** expression.

```
mySlider = Math.round (thisComp.layer("Controls").effect("LogoChoice")  
("Slider"));  
if (mySlider == 1) value else 0
```

This rounds the value to the nearest whole number, and the code will work beautifully, no matter where the user drags the slider.

No matter where the slider is positioned, the value will be rounded to the nearest whole number.



As you can see, after typing *Math.round*, we put the value that needs to be rounded inside parentheses. You can also use the *Math.floor* and *Math.ceil* expressions. These will round down and up to the nearest whole number.

Controlling the number of decimals

Math.round works very well when we need whole numbers, but sometimes we need a certain number of decimals. One way to do this is to use **toFixed** combined with **Number**.

```
mySlider = thisComp.layer("Controls").effect("Bar 1")("Slider");
Number(mySlider).toFixed(2);
```

The **toFixed(2)** code will limit the number of output decimals to a maximum of two. The word **Number** at the start of the last line tells After Effects to convert the string from the **toFixed** expression to a number.

There's a problem with this, though. The **toFixed** expression does not always round correctly! For example, values that end with .005 will be rounded down instead of up.

```
mySlider = 2.005;
Number(mySlider).toFixed(2); // outputs 2.00, should be 2.01
```

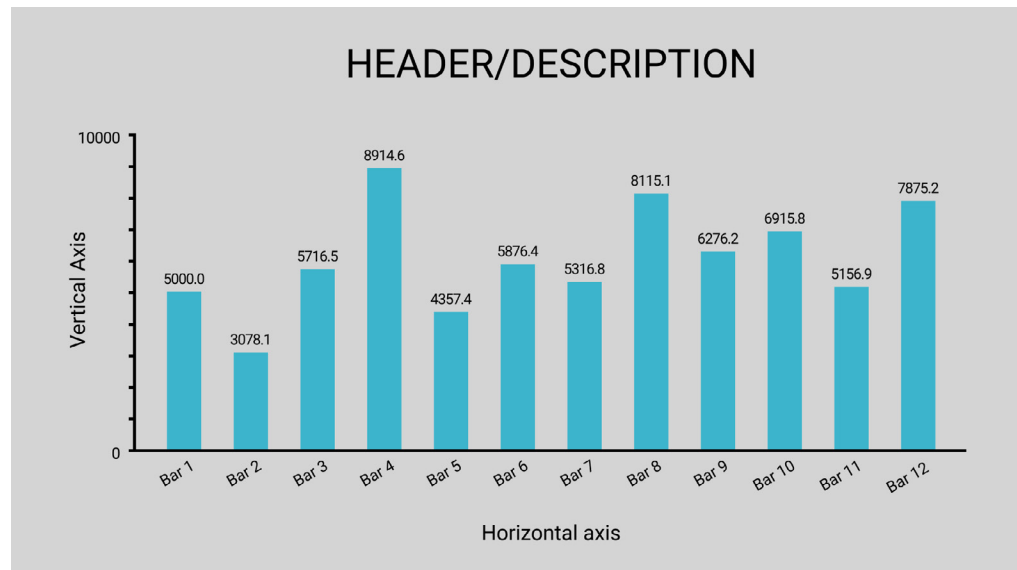
A solution is to use exponentials to multiply the floating-point number by some power of 10 in order to leverage **Math.round**.

I used this approach in a MOGRT with an animated bar chart, except that I used **parseFloat** (I could have used **Number**). In this MOGRT, the bar heights and the numbers over the bars are driven by one slider, and the number of decimals is driven by another slider. Here's the code I used to control the number of decimals. This particular one is for Bar 12.

```
dm = Math.round(thisComp.layer("Controls").effect("Decimals")("Slider"));
num = thisComp.layer("Controls").effect("Value 12")("Slider");
a = Math.round(parseFloat((num * Math.pow(10, dm)).toFixed(dm))) / Math.pow(10, dm);
parseFloat(a).toFixed(dm)
```

The values and bar heights are driven by one slider, and the number of decimals by another. This makes the MOGRT flexible and still very user-friendly.

With so many numbers to type, the MOGRT would be even more user-friendly if we let users use a spreadsheet to enter the data. You'll learn about that later.



Jack Moore explains how to avoid the rounding problem by using numbers represented in exponential notation in [this post on his website](#). His method creates leaner code than mine.

Split

The JavaScript **split** method is used for splitting text strings into smaller arrays. It outputs an array containing the split values. The syntax looks like this:

```
string_name.split (separator, times_to_split )
```

Let's say your client has a list of addresses written with commas separating the name, street, city, and ZIP code. It would look like this: *John Doe, 1450 That Street, San Jose, CA 12345*. You want to split any address that the MOGRT user pastes into a text field in the Essential Graphics panel into separate lines.

You can achieve this by adding the following expression to the Source Text parameter of a separate text layer. It will get its source text from the other layer, split the text whenever it sees a comma, and output the person field followed by a line break, then the street field, another line break, then the city field, then a comma, and lastly the zipcode field.

```
string = thisComp.layer("Paste Address Here").text.sourceText;
fields = string.split(",");
person = fields[0];
street = fields[1];
city = fields[2];
zipcode = fields[3];
person + "\n" + street + "\n" + city + ", " + zipcode
```

The result looks like this:

```
John Doe
1450 That Street
San Jose, CA 12345
```

Note that I could not use the word *name* for the variable in this code. The word *name* would return the name of the layer. That's why I used *person*.

The *times_to_split* value (aka *limiter*) is not used very often in MOGRTs since you don't always know how long the user text will be.

for loop with split

It's quite common to use something called a *for loop* that loops through a block of code a number of times. This is especially useful when you have lots of arrays. Here's the normal way of dealing with arrays. You can try it on the *Source Text* parameter of a text layer.

```
string = thisComp.layer("Our Cities").text.sourceText;
result = string.split(", ");
result[0] + "\r" +
result[1] + "\n" +
result[2] + "\n" +
result[3] + "\n" +
result[4] + "\n" +
result[5]
```

The more cities we have, the more typing we have to do. Using a for loop, we could use the following code no matter how many cities we have.

```
string = thisComp.layer("Our Cities").text.sourceText;
result = string.split(", ");
myText = "";
var i;
for (i = 0; i < result.length; i++) {
    myText += result[i] + "\n";
}
myText
```

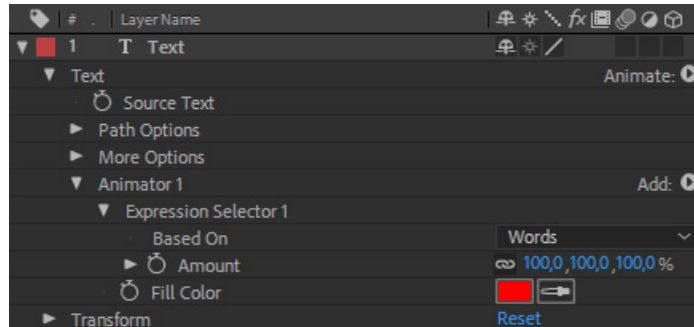
Yes, this code uses a few things we haven't covered here, but the good news is that you don't have to understand it to use it. Just paste the code into the *Source Text* parameter of your text layer and change the name of the source layer in the first line from *Our Cities* to whatever your layer is called, and it will work.

Don't worry—this is what everyone does. Copy and paste is by far the fastest way to write expressions. The split method can also be used to find and replace characters. You could find commas and replace them with periods, or find line breaks and replace them with spaces.

Highlight chosen words in text

Let's use split and a for loop for something more meaningful to MOGRT designers. Use two text layers: one named *Text* and one named *Highlighted Words*. Add an expression selector to the *Text* layer by choosing **Animation > Add Text Selector > Expression**. Then add a Color animator by choosing **Animation > Animate Text > Fill Color > RGB**. Change *Based On* to *Words*, and you should see something like this.

Text layer with Expression Selector and Fill Color animator added.

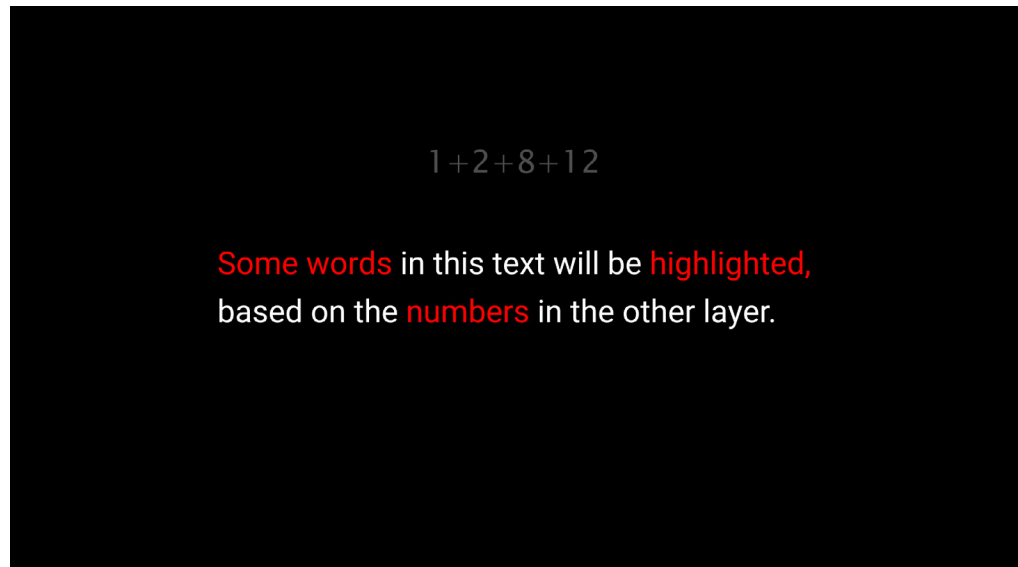


Alt-click the stopwatch for *Amount* and paste the following code, which is based on code that I found on [the Creative Cow forum](#).

```
string = thisComp.layer("Highlighted Words").text.sourceText.split("+");
result = 0;
for (i = 0; i < string.length; i++){
    if (textIndex == parseInt(string[i],10)){
        result = 100;
        break;
    }
}
result
```

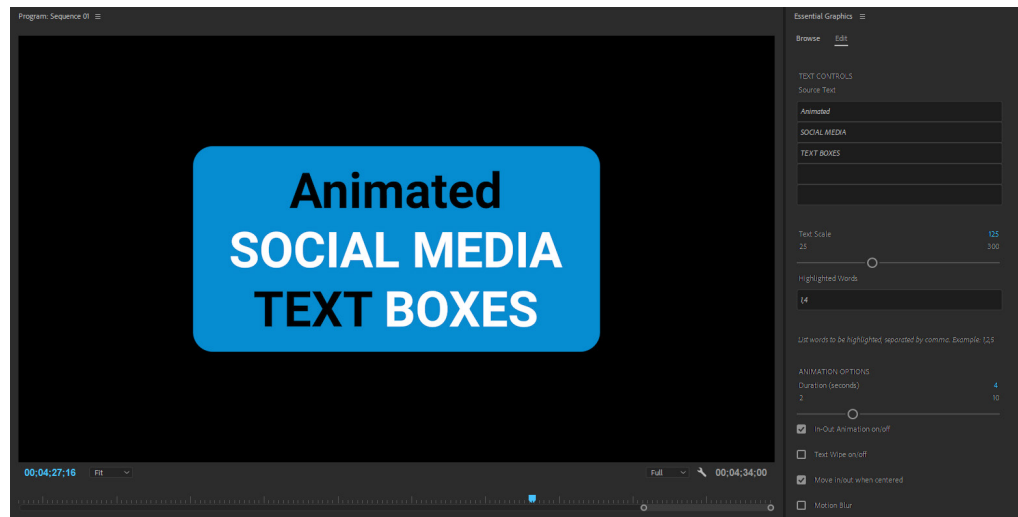
Now start typing numbers separated by plus signs in the *Highlighted Words* text layer. The words at those numbers in the main *Text* layer will change color.

The numbers, separated by + signs, in the *Highlighted Words* layer control which words are highlighted in the main text.



It's a good idea to make the *Highlighted Words* layer a guide layer (**Layer > Guide Layer**) to make sure it's never rendered. A guide layer will be visible while you work in After Effects but will never show in the output.

I used this method in my [Animated Social Media Text Boxes template](#), available on Adobe Stock. In that MOGRT, I used commas as separators instead of + signs, though.



Read more about the split method here: https://www.w3schools.com/jsref/jsref_split.asp.

Change the timing of keyframes with expressions

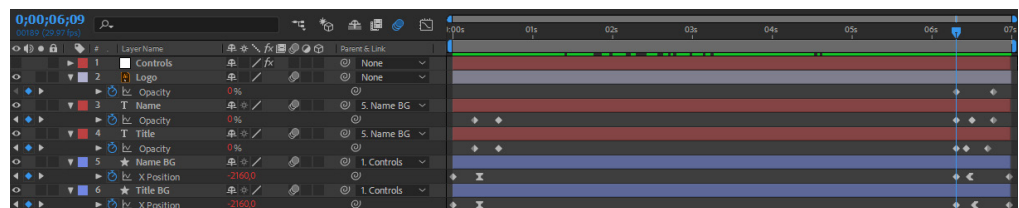
Some keyframed animations can be hard to re-create with expressions. Wouldn't it be great if we could change the timing of keyframes and move the out-animation with a slider? Well, thanks to the **key(index)** expression and some Dan Ebberts code magic, we can.

This code assumes that the first keyframe we want to move is keyframe 3 of the property with the expression. The keyframe number can be changed in the first line of code.

```
keyOut = 3;
tOut = thisComp.layer("Controls").effect("OutroStart")("Slider");
if (time < inPoint + tOut){
    value;
}else{
    t = time - (inPoint + tOut);
    valueAtTime(key(keyOut).time + t);
}
```

This code works perfectly as long as the keyframed out-animation doesn't happen earlier than the time specified by the duration slider. I made my *OutroStart* slider go from 2 to 6 seconds and placed all my outro keyframes after 6 seconds to make sure there was no conflict.

Note the extra keyframes on some layers. Without these, all the layers would animate out at once.



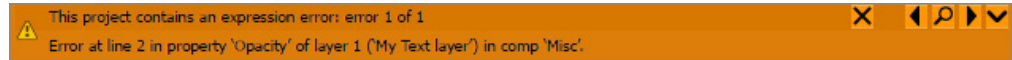
All the layers have the same expression on their keyframed properties. I wanted to keep the staggered out-animation, so I added extra keyframes. The code looks for keyframe number 3, so I added some extra keyframes on some layers to make sure the staggered out-animation was intact. I did not want intro keyframes on the logo layer, so I changed the first line of code on that layer to **keyOut = 1**.

Note that the expression will throw an error message if there is no keyframe number 3, so make sure you either change the number or have at least three keyframes. I guess this code will be used less now since the introduction of Responsive Design — Time in After Effects CC 16.0. But it's still a good way to move keyframes around with expressions.

Expression error banner

The expression error banner (the orange bar that appears in the Composition panel when an expression fails to evaluate) has some buttons. The *Hide Expression Error Banner* button (X) will hide the banner and disable the *Show Warning Banner When Project Contains Expression Errors* option in **Preferences > General**. To show the expression error banner again, go to the preferences and re-enable it.

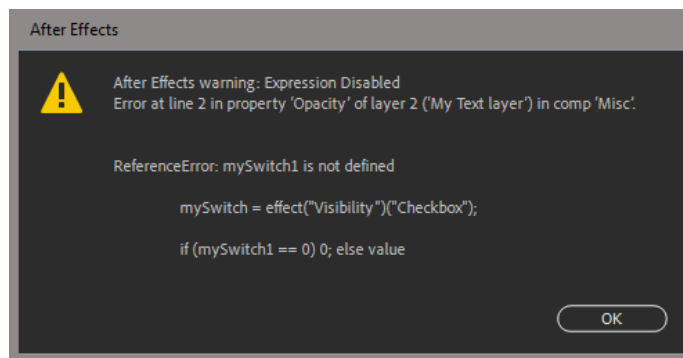
The expression error banner tells you what the error is and where to find it.



The two arrows will show the previous or next error message, if you have more than one. The arrow to the far right will show the messages or collapse the banner to a single line. The magnifying glass takes you to the comp where the error is and highlights the problematic property.

If you click the warning triangle to the left of the expression in the layer, you get more info about the error.

The spelling error in this expression is easy to spot because the message says *mySwitch1 is not defined*.



Making MOGRTs user-friendly

Unfortunately, many MOGRTs out there are hard to use. This chapter offers tips that make the MOGRT experience better for the user, which will put more money in your pocket. If your MOGRTs are hard to use, your customers will not be satisfied, and you will earn less money.

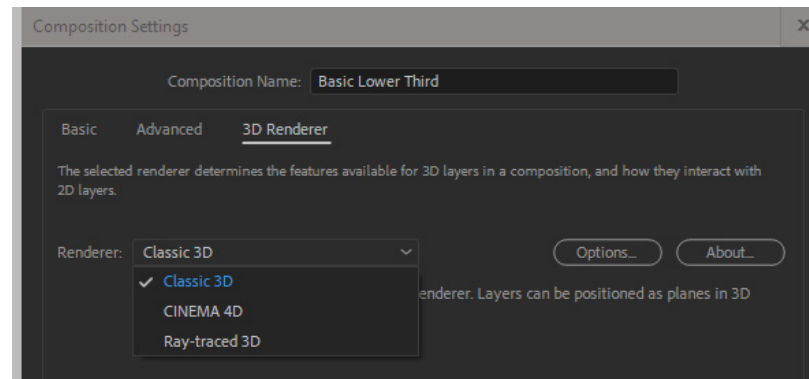
“The user experience of the majority of mogrts is fairly complicated and not intuitive, so if you're able to make your controls simple and easy to use you've already separated yourself from most other developers.”

— Nick Howell

Decide if it's okay if After Effects must be installed

Since the October 2017 release of Premiere Pro, users no longer need to have After Effects installed on the same computer to render MOGRTs. There are some features and effects that will require After Effects to be installed, though.

Only the Classic 3D renderer is supported in MOGRTs when After Effects is not installed.



Effects and features that will not work with Premiere Pro alone

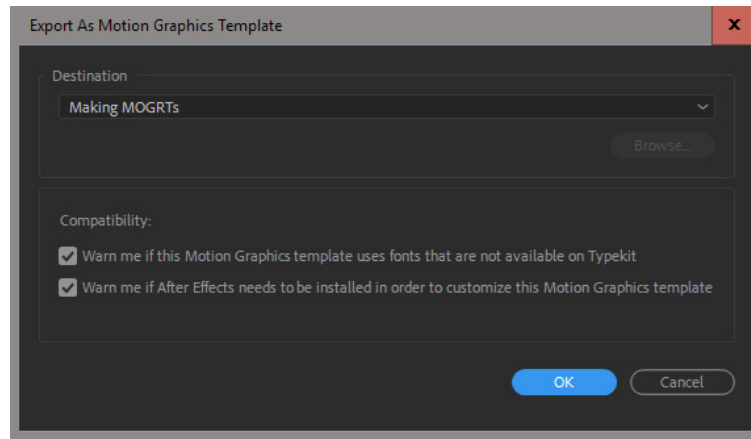
These effects in After Effects and features are only supported in MOGRTs via Dynamic Link. If you use them, the end user needs to have After Effects installed.

If you're selling MOGRTs through Adobe Stock or other sites, you should not use these effects and features, but if it's a custom MOGRT for a client and all the users have After Effects installed, then it's kay.

- **CINEMA 4D and Ray-traced 3D renderers** are not supported. Only the Classic 3D composition renderer is supported.
- **The following effects** are not supported.
 - Camera-Shake Deblur
 - Lumetri Color
 - Puppet
 - CINEWARE
 - Immersive video (VR) category of effects
 - Dynamic Link footage like Premiere Pro sequences and Character Animator scenes
 - FLV, C4D, and SWF footage
 - Third-party plug-ins

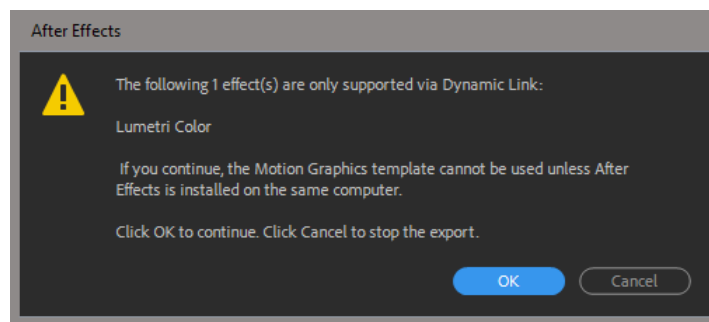
Some websites where you can sell MOGRTs will not accept MOGRTs that depend on After Effects being installed. And users will not be very happy if they find out they need to install another app to use the template they've bought. Because of this, there's a checkbox in the Export As Motion Graphics Template dialog box for getting a warning if the template will not work unless After Effects is installed.

You can choose to be warned if you export a MOGRT that requires After Effects to be installed.



If you export a MOGRT where one of these effects is used and *Warn me if After Effects needs to be installed...* is selected, you will get one or more reminders like this one.

The warning tells you which effects are not supported without After Effects installed.



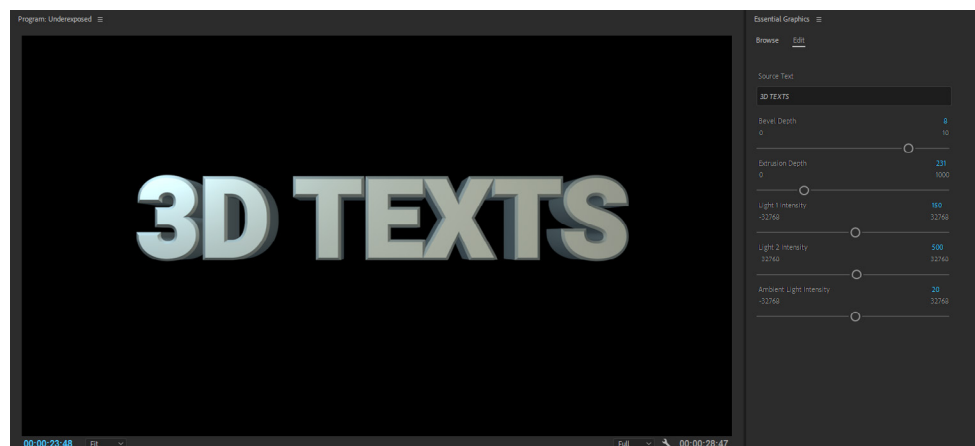
Making MOGRTS for users who have After Effects installed

The fact that certain effects and features are only supported via Dynamic Link doesn't mean that you can't use them when you're making MOGRTs, though. It's just that the user needs to have After Effects installed along with Premiere Pro to use them. If your clients have After Effects installed, go ahead and use these options.

The most used feature in this kind of template is probably the CINEMA 4D 3D renderer. Using this, you can make extruded 3D text, shapes, logos, and so on.

If you know that the end user has After Effects installed, there's nothing wrong with using the effects and features in the "not supported" list.

Add some real 3D to your MOGRTs with the CINEMA 4D 3D renderer.



If you use these features, make sure you inform your clients that for the MOGRT to work in Premiere Pro, they need After Effects too.

Pre-render non-supported effects if possible

If you really like a third-party effect (or one of the other non-supported ones) and would love to use it in a MOGRT you want to sell via Adobe Stock, there may still be hope. If the effect does not need to respond to whatever the user adjusts in the Essential Graphics panel in Premiere Pro but is only used on a background layer, you may be able to pre-render the layer with the third-party plug-in and use the rendered video file in the MOGRT instead of the source layer with the effect.

If the client has a special font

that everyone in their company has installed, you can, of course, use that font and ignore the warning, but not if you make MOGRTs for sale on Adobe Stock or other sites.

Use fonts that are available to all Adobe Creative Cloud users

There's a checkbox in the Export As Motion Graphics Template dialog box for getting a warning if the MOGRT uses fonts that are not available on Adobe Fonts. When enabled, this option will warn you if the exported MOGRT uses fonts that were not synced from Adobe Fonts.

But what if you've paid for that font on Adobe Fonts, and you're not sure if the users will have it installed? Then they would have to buy the font to be able to use the MOGRT, which will most likely make them angry. If you want to be sure that everyone can sync the font for free, use only the fonts that are available in the trial version of Premiere Pro. These are all free to sync from Adobe Fonts. Here's a link that will show you only these fonts.

There are some nice fonts that everyone can sync for free from Adobe Fonts.



<https://premierepro.net/free-fonts-for-mogrts/>

If you see a font and want to use a similar font you can use the [Adobe Capture app](#) to take a picture of the text and let Capture find a similar font from Adobe Fonts.

Use sliders and rounding to give the user some options

We've already used this a few times, but it's worthy of its own mention. When you use rounding combined with a slider and some *if/else* statements, the user can very quickly and easily change one or more options. The following example moves a logo layer to 100 pixels from any corner in a 1920x1080 frame, or to the center, based on the value of the slider called *Logo Placement*.

```

logoPos = Math.round ( thisComp.layer("My Controls").effect("Logo Placement")
("Slider") );
if (logoPos == 1) [100 ,100]
else if (logoPos == 2) [1820, 100]
else if (logoPos == 3) [100, 980]
else if (logoPos == 4) [1820, 980]
else [960, 540]

```

Don't give the user too many controls. It makes the MOGRT harder to use.

Control several properties with one slider or checkbox

The more properties you can link to one slider, the tidier and more user-friendly the MOGRT will be in Premiere Pro. In the bar graph example here, the value above a bar and the bar height are of course taken from the same slider. I've seen MOGRTs where the user had to type the same value in two different fields, which is stupid.

Also, when the user controls the number of marks on vertical axis, this will automatically adjust the distance between them, so the top stays in the same place. Having two sliders for this would be less user-friendly. Always think of ways you can cut down on the number of controls in your MOGRTs.

Making several text layers and shape layers move, scale and respond to as few sliders as possible is a very good skill to have as a MOGRT designer.

"I feel that making templates is a balance between creating something great and being able to allow a user to define a template in a way you didn't intend or expect. It's a much different experience than designing for a client, that's for sure."

— Theresa Rostek, motion graphics production artist, Adobe

Give the user a Duration slider in earlier versions

Contrary to what many designers think, it is totally possible to control duration and other timing elements of a MOGRT with sliders. As you've seen in the section on expressions, we can use the linear and ease expressions to control animations with a slider, and it's also possible to move keyframes with expressions.

These methods are very useful if you make MOGRTs for older versions of Premiere Pro. If you're making them for customers with Premiere Pro 13.0 (the October 2018 release), you can use Responsive Design — Time instead.

Allow for multiple lines of text

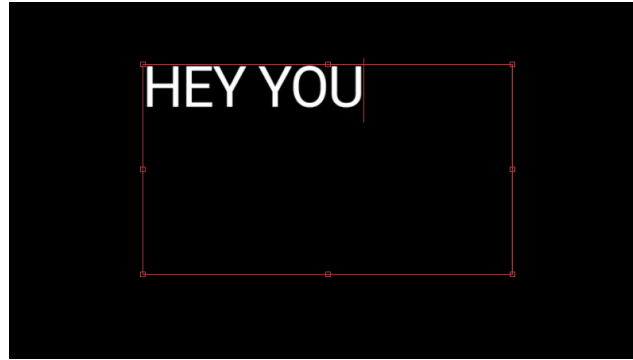
There many approaches to creating multiline text blocks in MOGRTs, and they all have their strengths and weaknesses. The three most common are the following.

- Use several text layers
- Use area/paragraph text (text box)
- Use point text (standard text) and add line breaks

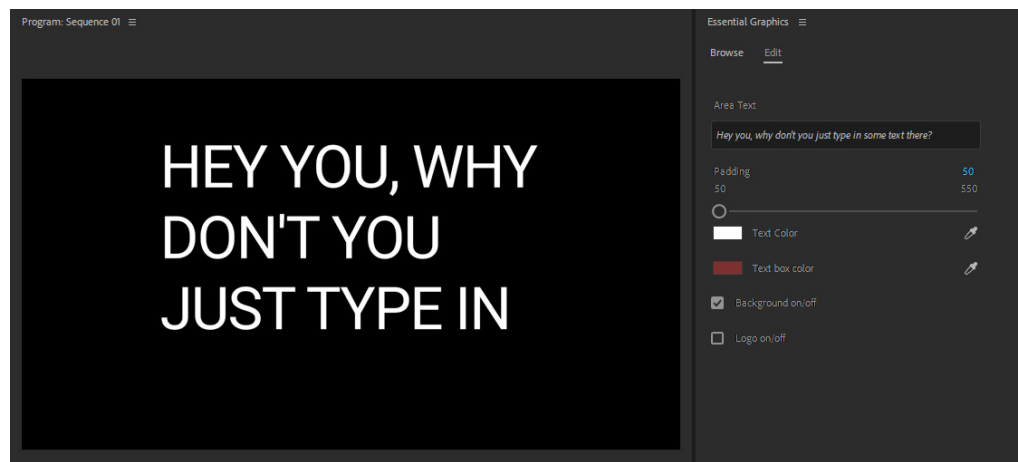
Using four text layers for four lines of text is okay if you want to animate the lines independently, but you can also do that by using a text animation preset on area text, and make sure it's based on lines, not words or characters. Multiple text layers also make the expressions for text placement, scaling, shape layer size, and so on more complicated, as they need to add the height of several layers.

The next option, area text, is fine if you want to restrict the "legal" text area to a certain area of the frame. But in Premiere Pro, the user will have only one line in the MOGRT to edit text in, and if the text is long, it will be cut off.

Click and drag with the Type tool to create an area of text in After Effects.



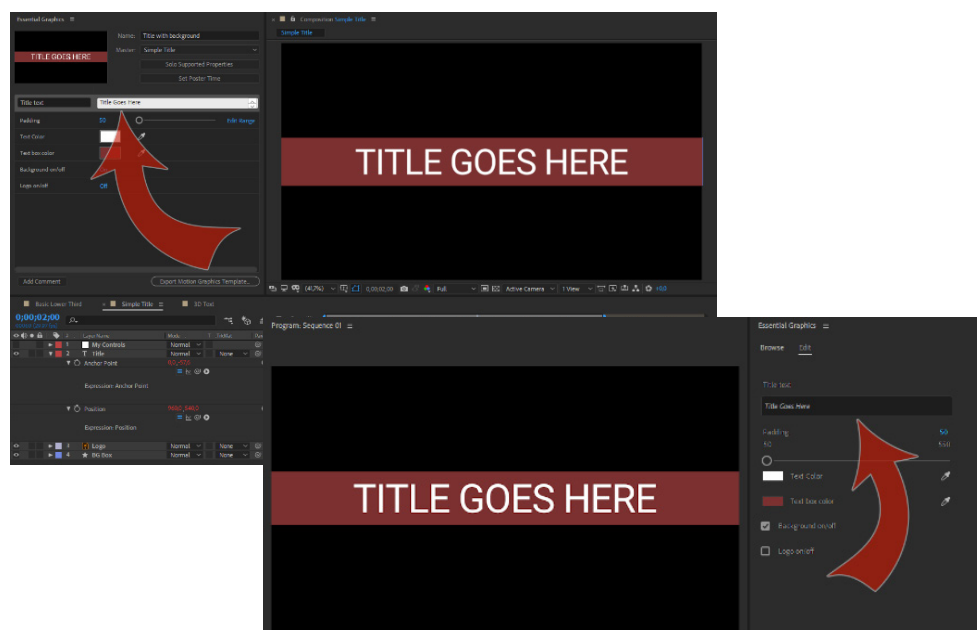
When you use area text, the text may get cut off if it's too long.



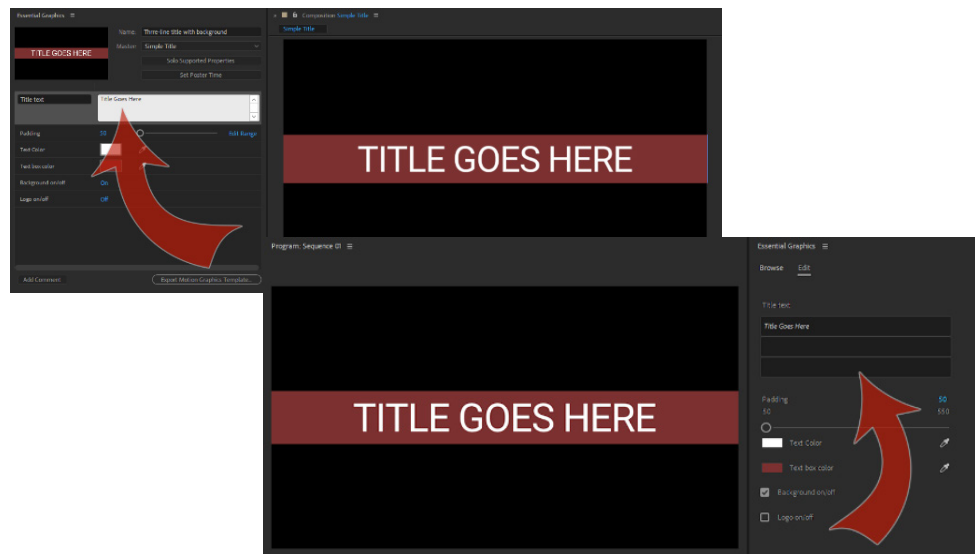
The best solution in most cases is to simply use normal text (point text) and add line breaks in the Essential Graphics panel. These will show as empty lines in the Essential Graphics panel in Premiere Pro. The user can also decide where to place the line breaks, and it's very intuitive to use. The user can insert more text at will, and *sourceRectAtTime* expressions will still work as usual.

There's one caveat though: This breaks the new feature in Premiere Pro 12.1.0 with which you can click with the Type tool on the text in the Program Monitor, even with a MOGRT made in After Effects, and enter your text. The point where you can click ends up in illogical places. It seems this feature doesn't understand this concept of multiple lines, but that may change in future versions.

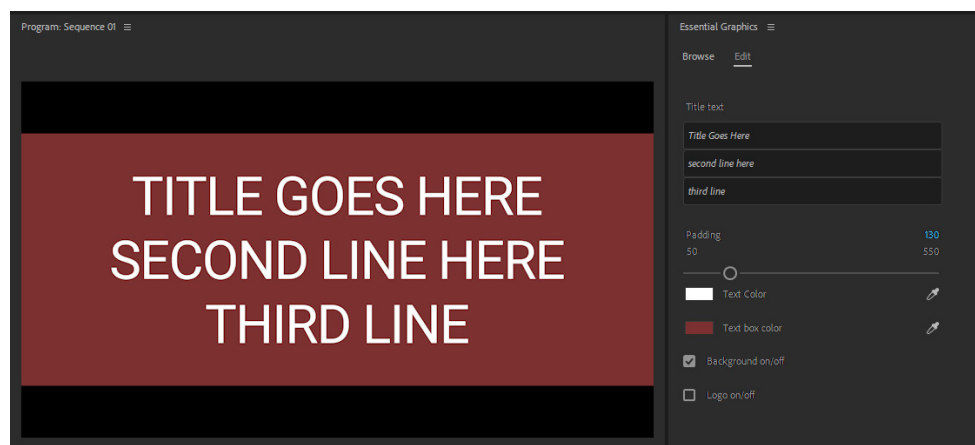
With only one line of text in the Essential Graphics panel in After Effects, the user can only type one line of text in Premiere Pro.



The MOGRT gets as many lines in the Essential Graphics panel in Premiere Pro as you have entered in After Effects. Allowing for three to five lines is quite common.



If the user enters text in the empty fields, the expressions make the shape layer expand to accommodate the extra lines.



Here's the expression on the Size parameter of the BG Box shape layer that makes the rectangle grow vertically when there are extra lines in the text.

```
y = thisComp.layer("Title").sourceRectAtTime().height;
padding = thisComp.layer("My Controls").effect("Padding")("Slider");
[1920, y+(padding*2)]
```

Here's the expression on the anchor point of the text layer that makes sure that the text is vertically centered no matter what the height is.

```
txtHeight = thisLayer.sourceRectAtTime().height;
txtTop = thisLayer.sourceRectAtTime().top;
y = txtTop + txtHeight/2;
[0, y]
```

Allow for different text designs

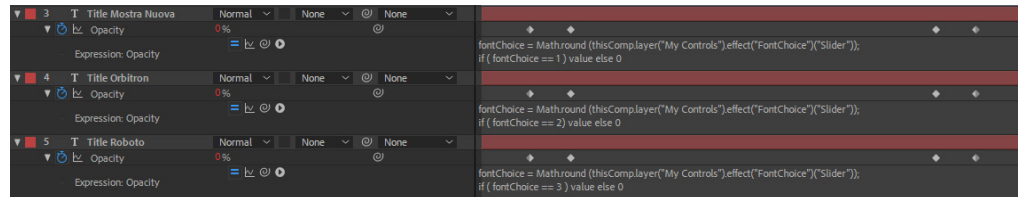
Since the properties in the Character and Paragraph panels are not controllable with expressions, they can't be changed in a MOGRT. There are some workarounds though, so the user can change the text color, font, tracking, and so on.

Font choices

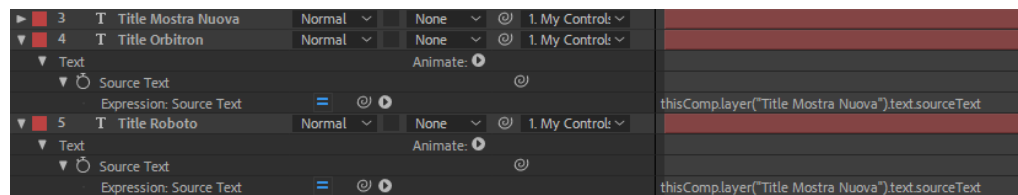
To provide restricted font options, you can create different layers for each font and use *if/else* expressions to set the Opacity to zero for all but one, depending on slider value. You also need to link the Source Text parameter on the extra layers to the first one.

```
fontChoice = Math.round (thisComp.layer("My Controls").effect("FontChoice")
("Slider"));
if ( fontChoice == 1 ) value; else 0
```

These three text layers with different fonts get their Opacity set to 0 according to the Font Choice slider value.



Make sure you link the source text of the extra layers to the source text of the first layer, so the text will change on all layers when the user enters new text.



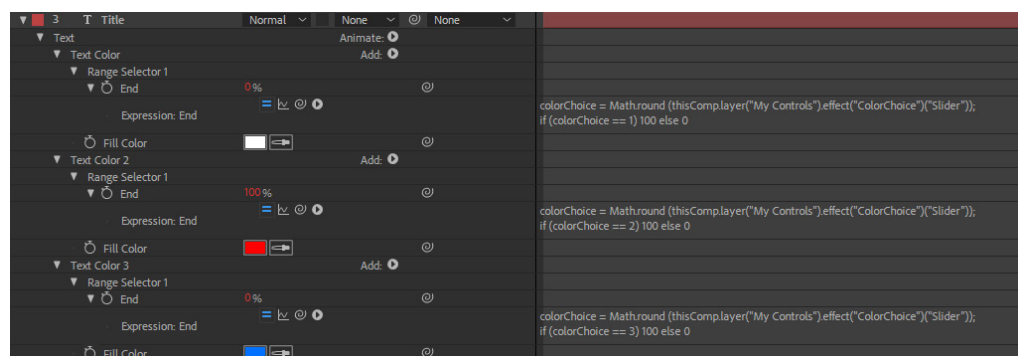
Text color choices

If you add a *Text Color* animator, you can put it in the Essential Graphics panel and let the user select colors freely. You can also use expressions to change the text color this way. When I make templates for a client that has a design guide with only a few approved colors, I use several of these Text Color animators and let the user choose between a limited set of colors with a slider.

Here's how. Click the Animate button on the text layer and choose **Fill Color > RGB**. Duplicate it as many times as you need and change the fill colors. Then paste the following code to the *End* parameter on all the Text Color range selectors and change the number after *colorChoice ==* for the layers so they correspond to the slider values.

```
colorChoice = Math.round (thisComp.layer("My Controls").effect("ColorChoice")
("Slider"));
if (colorChoice == 1) 100; else 0
```

These three Text Color animators will let the user choose between four colors: the original text color plus the three colors from the animators.



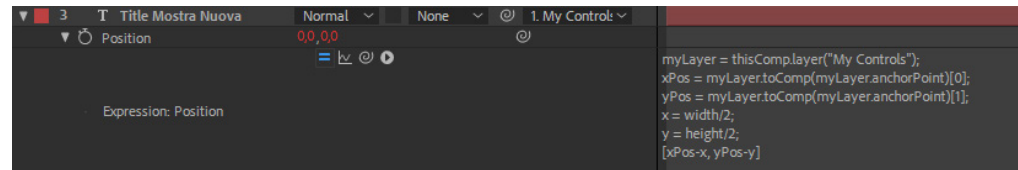
If you want to use this trick on multiple text layers, make sure you link the colors and opacity of the same properties on all layers to one of them. This makes it easier to do changes to the code, as you only need to change it in one place.

Text scaling

It's easy to enable font size adjustment in the Source Text properties in the Essential Graphics panel. But I don't like to give users full control over the text scaling. I like to give them a slider and let that slider scale the text within certain limits.

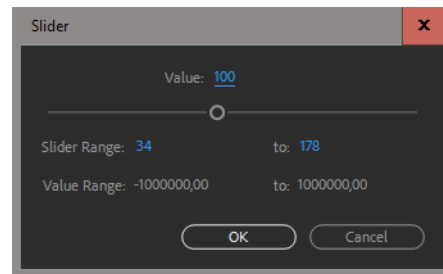
One easy way to make the text and the background shape layer scale proportionally is to parent them both to a null object and scale that layer. I often parent everything to the Controls layer. Just make sure that you use the *toComp* expression trick that you learned in the chapter on useful expressions to fix any positioning problems you get when parenting.

Use the *toComp* expression to fix positioning problems after parenting, if needed.



You could just add the Scale property of the Controls layer to the Essential Graphics panel, but then the user could potentially set the value to very high and very low, even negative values. You will probably want to use a Slider Control with set limits to adjust scaling instead. Use the Property Pick Whip to link the Scale of the Controls layer to the Scale slider. This works better, since you can set the limits on the slider.

Using a Slider Control for scaling lets you set the range limits.



Auto-scale text

Since you never know how much text users will put in a MOGRT, it can be hard to decide where to set the limits for the range. If you allow them to scale the text so that the word *WOW* fills the frame, the words *BANANA BOAT* would be way outside the boundaries. You could let users scale to a large size and let them handle the problem themselves by readjusting the size.

Another approach is to *not* let them scale it, but dynamically change the size when more text is added, so it's always the same width. Measure the text size. When it gets longer or taller than you want, scale it down. Here I've set the max height to 960 pixels and the max width to 1800 pixels.

```
txtWidth = thisLayer.sourceRectAtTime().width;
txtHeight = thisLayer.sourceRectAtTime().height;
if (txtWidth > txtHeight) newSize = (100 * 1800/txtWidth);
else newSize = (100 * 960/txtHeight);
[newSize, newSize]
```

The text automatically scales to fit the size limits I've set.



This works beautifully if it's only one line of text, and the anchor point is in the middle of the layer: **Layer, Transform > Center Anchor Point in Layer Content**. If you want to give them more control

over the size with a slider, things get more complicated because you're in a way adjusting the size you're measuring. You'll have to do some anchor point math magic to keep the text in the right position when it's scaled.

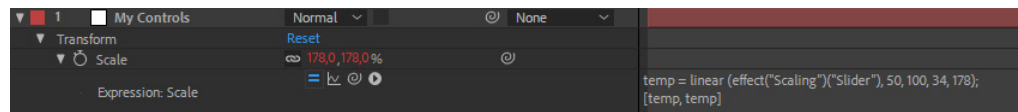
Make sliders with logical ranges

What if you find that the MOGRT looks best when the scale is between 34 and 178? Those numbers are not logical to the user, and they might think it's strange that the slider stops at 34 and 178.

This is where the *linear* expression comes in handy. If you change the temp value that the Property Link Pick Whip created to the following, the scale will go from 34 to 178 as the slider moves from 50 to 100, which is a much more logical range for the user.

```
temp = linear (effect("Scaling")("Slider"), 50, 100, 34, 178);  
[temp, temp]
```

The linear expression
lets you customize the
slider values to a logical
range.



Let the user control the line length

Once again, we'll stand on the shoulders of a giant, and lean on code from expressions guru Dan Ebberts. I based the following incredibly useful code on some code from Dan in [a post on the Creative Cow forum](#), where he's very active.

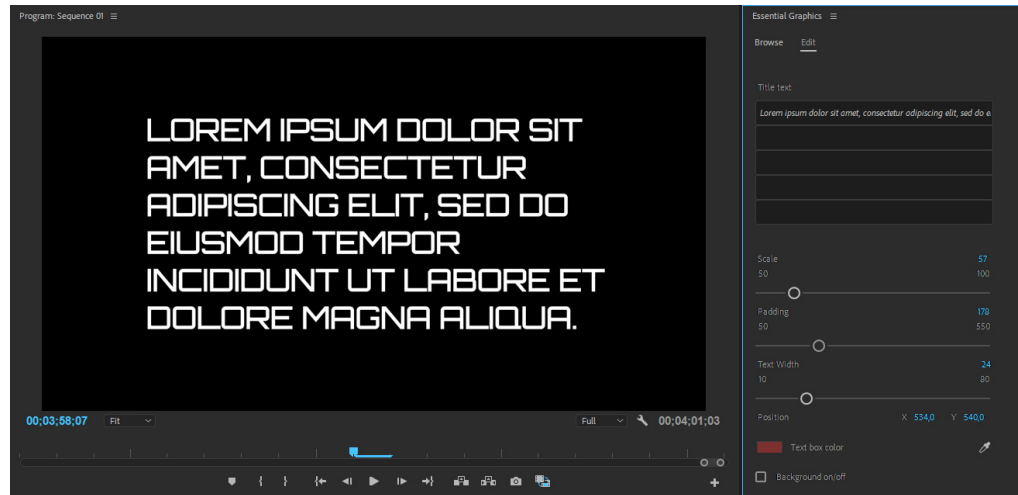
```
txt = thisComp.layer("Text Source").text.sourceText;  
n = thisComp.layer("My Controls").effect("Text width")("Slider");  
outStr = "";  
newLine = ""  
splt = txt.split(" ")  
for (i = 0; i < splt.length; i++){  
  if ((newLine + " " + splt[i]).length > n){  
    if (outStr != "") outStr += "\r";  
    outStr += newLine;  
    newLine = splt[i];  
  }else{  
    if (newLine != "") newLine += " ";  
    newLine += splt[i];  
  }  
}  
if (newLine != ""){  
  if (outStr != "") outStr += "\r";  
  outStr += newLine;  
}  
outStr;
```

Isn't it great that we don't need to understand the code to use it? Copy and paste for the win!

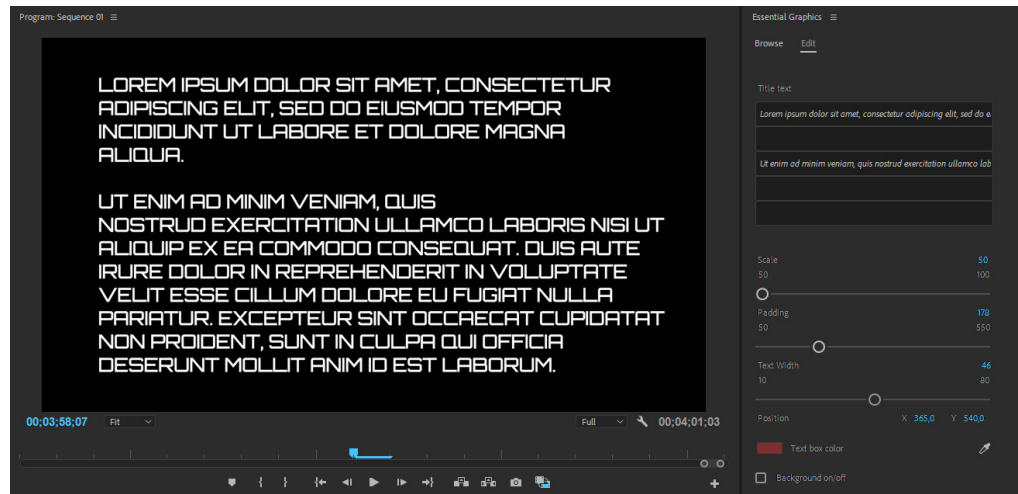
The code adds line breaks after a certain number of characters, but not in the middle of a word. The number of characters can be controlled by a slider.

Using this code, I made a MOGRT where the user can paste a bunch of text into the text field in the Essential Graphics panel in Premiere Pro and easily control the scale and width of the text with sliders.

This MOGRT gives the user total control over the line breaks. Breaks can be added automatically after a certain width (number of characters) based on the slider value or manually.



If you add extra line breaks before exporting the MOGRT, users can insert line breaks where they want by splitting the text into multiple text fields in Premiere Pro.



In the example here, I haven't taken the scale into account, so the line length (measured in number of characters) stays the same whatever the scale is. When the scale is adjusted, the text width onscreen changes. It would be possible to use an expression with some math to make the line length depend on both the *Scale* slider and the *Text Width* slider, so it automatically makes the lines shorter when the text gets bigger. I'll leave this up to you.

Nick Howell's timing guide preview

When it comes to user-friendliness, it's hard to beat the best ones from [Nick Howell](#). Most of his MOGRTs have a *Timing Guide Preview* that you can turn on to temporarily get a visual representation of the timing. A playhead shows you the current position, and color-coded and labeled sections show the intro duration, the hold duration, and the outro duration.

The user gets a visual representation of the intro, hold, and outro durations and a playhead that shows the current time. Pretty slick!



Responsive Design — Time

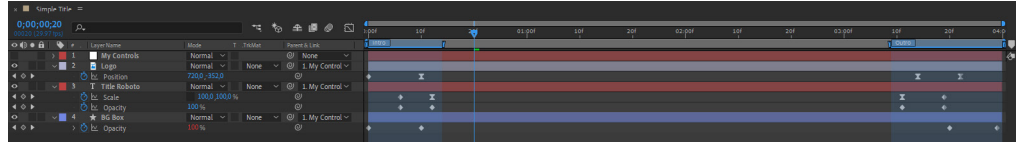
The Responsive Design features introduced in After Effects 16 (October 2018 release) let you create time-adaptive motion graphics. *Responsive Design — Time* enables adaptive time-stretching when a composition is nested or exported as a MOGRT. This is a great way to make your MOGRTs user-friendly!

The duration of your comp will be the default duration of the exported MOGRT when it's dropped into a Premiere Pro timeline. If you build it with Responsive Design — Time, the user can easily adjust the duration by dragging the out point.

Dragging the in and out points of the clip in Premiere Pro will time-stretch the composition instead of trimming the in or out point. The duration of protected regions will be preserved. This is a fantastic feature for MOGRT designers and makes the MOGRTs a lot easier to use in Premiere Pro.

Note: Adding audio to a MOGRT will disable the time-stretch functionality in Premiere Pro.

These intro and outro regions are protected with Responsive Design — Time.



Before we got this feature, we used *Duration* sliders in MOGRTs, moved keyframes with expressions and other voodoo techniques. And still, the duration of the clip in the Premiere Pro timeline had to be adjusted manually. *Responsive Design — Time* has changed all this, and has made both designing and using such MOGRTs much easier.

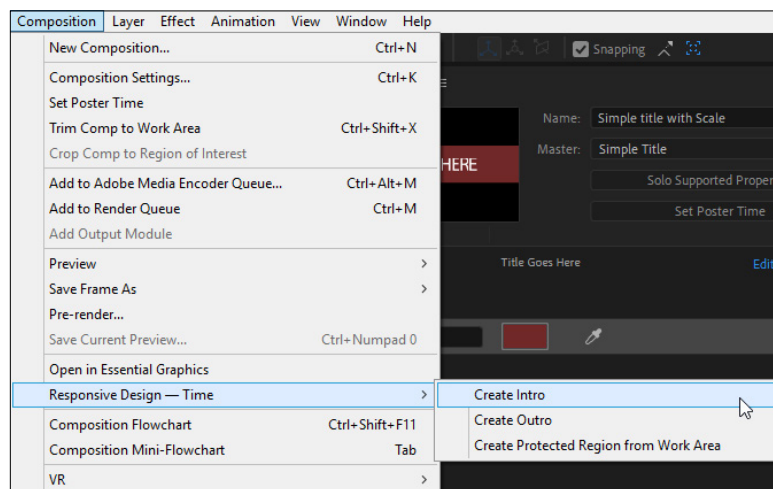
There are several ways to create a Responsive Design — Time protected region in After Effects. One is to choose one of the commands from the **Composition > Responsive Design — Time** submenu.

Create Intro creates a protected region for 15% of the composition duration, starting on the first frame of the composition.

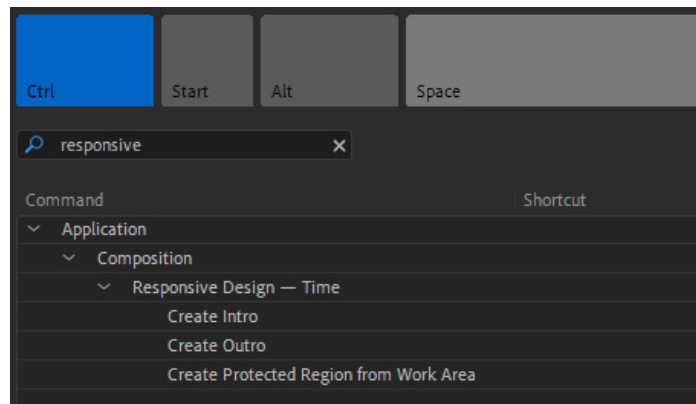
Create Outro creates a protected region for 15% of the composition duration, ending on the last frame of the composition.

Create Protected Region from Work Area creates a protected region with start and end points that match the work area bar.

The Composition > Responsive Design — Time submenu.

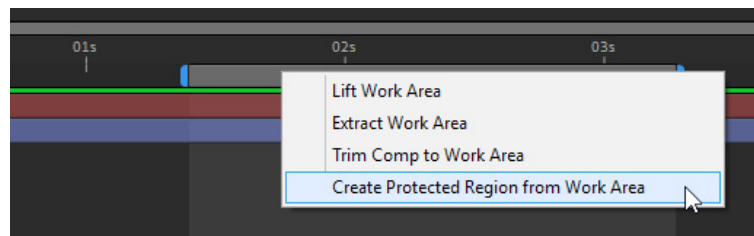


You can also make your own keyboard shortcuts for these commands.



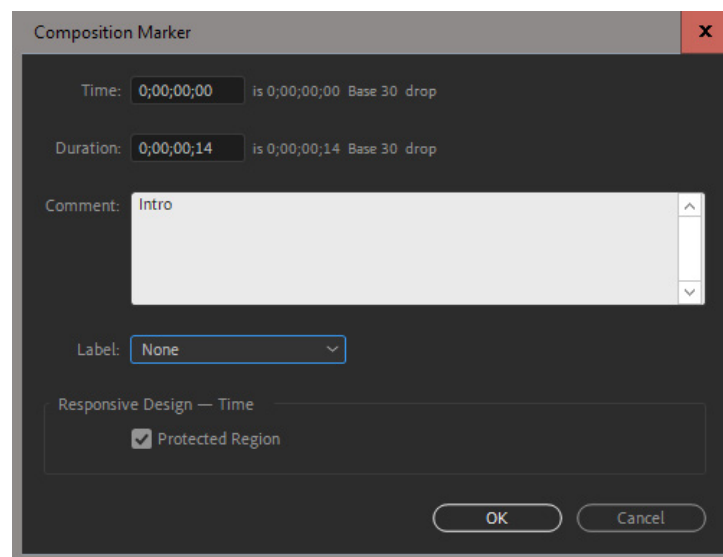
Another way to add a protected region is to right-click on the work area bar, and then choose *Create Protected Region from Work Area*.

You can create a Protected Region from the Work Area.



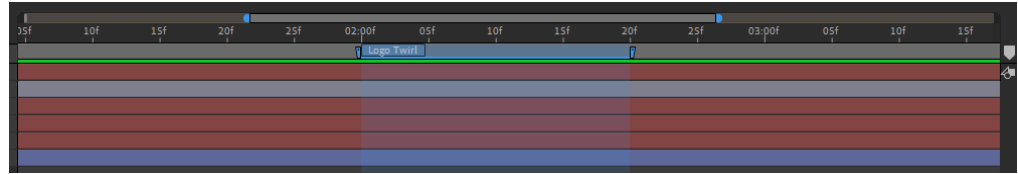
You can also create a composition marker first, using your favorite method, and then convert it to a protected region. Choose **Layer > Markers > Add Marker** or press the asterisk (*) on the numeric keyboard with no layers selected. Double-clicking an existing marker will open the Composition Marker dialog box. To create a protected region, set a duration, enable the *Protected Region* option, and click OK.

Set a duration and enable the Protected Region option in the Composition Marker dialog box.



You can also make protected regions in the middle of the animation. This is useful if you want to add some glitches or maybe twirl a logo around its own axis midway through the MOGRT and want the duration of this to always be the same.

Protected region in the middle of the animation.



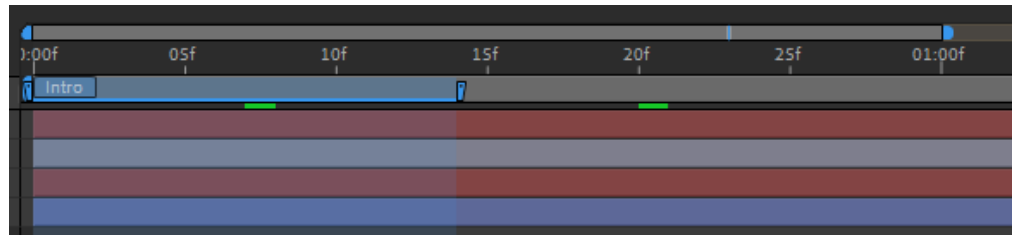
When you nest a composition with protected regions, dragging the in and out points of the layer will do a time-stretch, preserving the duration of protected region. Alt-drag the in/out points to do a standard trim.

Protected region markers behave the same as other composition markers, except that when you nest a composition with protected regions, those markers cannot be modified. Instead, the layer bar for the nested composition is shaded to indicate the protected regions.

If you modify a protected region after nesting a composition, the protected region marker on the nested composition layer is not automatically updated. To update markers on a nested composition layer, select the layer and choose **Layer > Markers > Update Markers from Source**.

It's a good idea to name the markers. I often use simple names like *Intro*, *Outro* and *Pause*.

Here I've named the marker *Intro*, so I can remember what it does, even after a few months, when I need to update the MOGRT.



Why still use a duration slider?

Responsive Design — Time replaces most of the needs for a *duration* slider. But it does not stretch the protected regions; instead it applies a single time-stretch rate to the entire clip. Using a *duration* slider to move keyframes or control linear or ease expressions, you can stretch the sections between protected blocks differently.

Foolproofing the MOGRT

If you want to get serious about foolproofing, learn to use *try* and *catch(err)* to fix user errors and situations where the code may result in dividing by zero and other illegal math operations.

Show warning banners

Remember when we used the `parseFloat` expression to convert numbers in the text to a floating-point number? Here's the code we used.

```
myText = thisComp.layer("Bar 1 Value").text.sourceText;
parseFloat(myText)
```

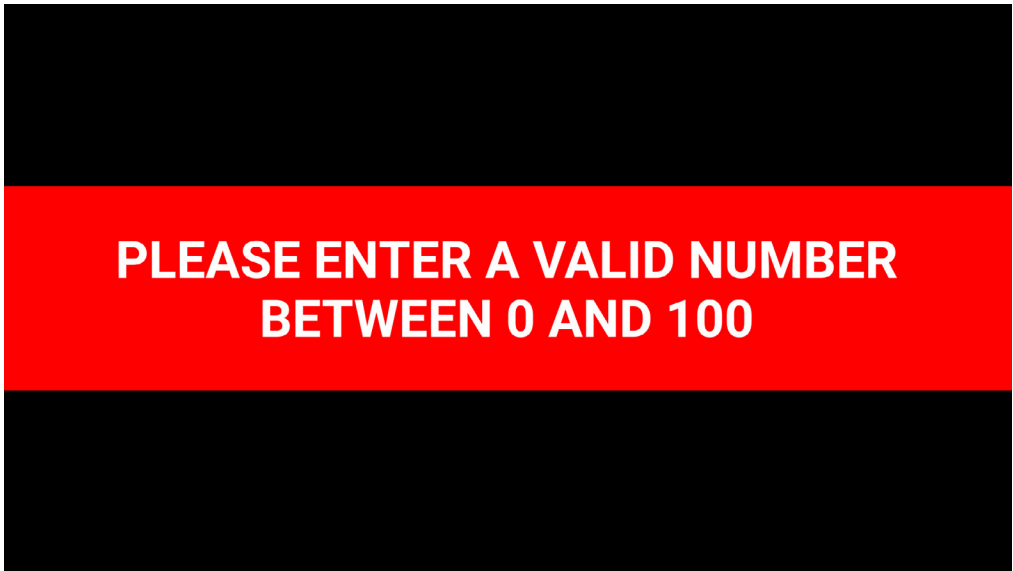
This will work fine as long as users enter a valid number. But what if they accidentally write some text in that field instead? The expression will break, and the user will not know why. It would be much better if we could detect the problem and let users know that they need to enter a valid number.

If you nest a comp with a red solid and a warning text, you can use the following code on the Opacity for that layer.

```
txt = parseFloat(thisComp.layer("Bar 1 Value").text.sourceText);
if (isNaN(txt)) 100; else 0;
```

The `isNaN()` part is code for *is Not a Number*. This will throw a big warning if the user enters text that's not a number.

When the user sees this warning, it's clear what the problem is.



PLEASE ENTER A VALID NUMBER
BETWEEN 0 AND 100

Use `catch(err)` and `break`

There may be other problems that are more serious than the previous one. Some expressions may create conditions where the code is trying to divide by a value that's zero, and the expression will be disabled.

This is what *try* and *catch(err)* can be used for. Here's the syntax structure.

```
try {  
    Block of code to try  
}  
catch(err) {  
    Block of code to handle errors  
}
```

Learning how to use error catching code enables you to build MOGRTs that never break.

We can use this kind of code on the Opacity of the layers that link to the source text of the layer named *Bar 1 Value*. The warning banner layer could have the following expression, so it shows when the entered text is not a number and doesn't break if something else is wrong.

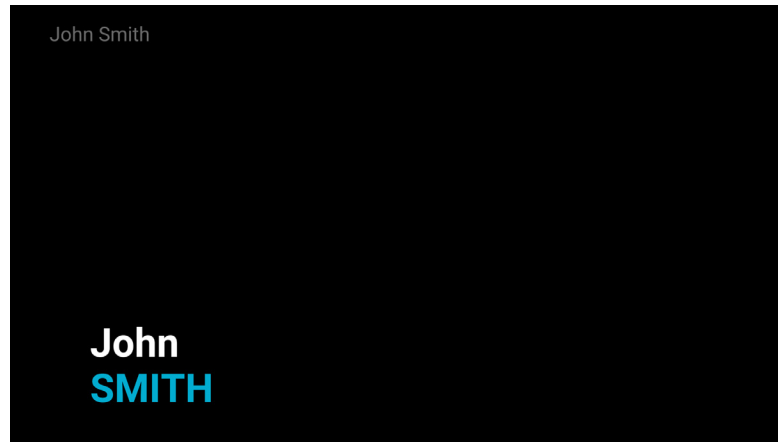
```
n = parseFloat(thisComp.layer("Bar 1 Value").text.sourceText);  
try {  
    if ( isNaN(n) ) 100; else 0;  
}  
catch (err) {  
    100;  
};
```

Here's another attempt at explaining the structure of the *try* and *catch(err)* syntax. Say you use this code on the Source Text parameter of a text layer.

```
nameArray = thisComp.layer("NameSource").text.sourceText.split(" ");  
try {  
    nameArray[0] + "\r" + nameArray[1].toUpperCase();  
}  
catch(err) {  
    nameArray[0];  
}
```

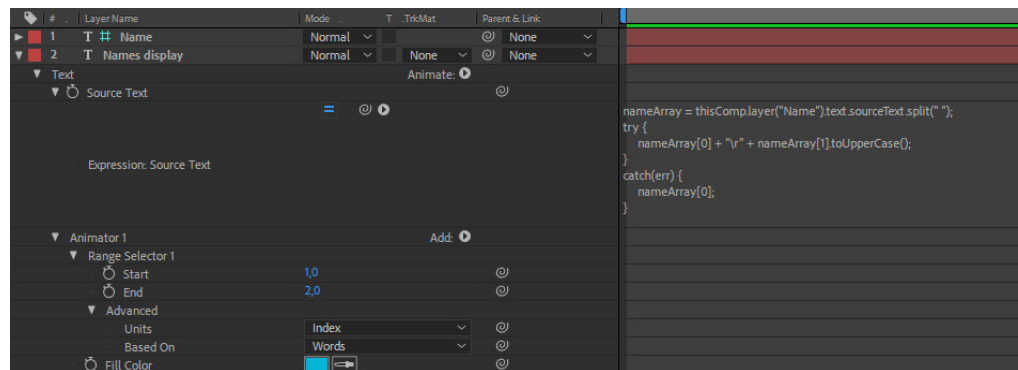
If the *NameSource* layer has both a first name and a last name, the code under *try* will be executed. It detects spaces, splits the text into arrays, outputs the first name, adds a line break, and then outputs the last name, converted to uppercase text.

The first name and last name are automatically formatted.



If the *NameSource* layer doesn't have a last name, the first code, under *try*, will fail. Instead of throwing an error and disabling the expression, the second code, under *catch(err)*, will be used, and display only the first name. In this case, I also added a Text Color animator that adds color to word number 2.

These are the settings that control uppercase text and text color.



Data-driven MOGRTs

Starting with the October 2018 release of After Effects, you can add controls for CSV and TSV (comma-separated values and tab-separated values) spreadsheet data to the Essential Graphics panel. MOGRTs with data controls allow the data to be modified or replaced in Premiere Pro.

This means that the MOGRT user can create a spreadsheet and link the MOGRT to it, and all values are updated automatically. This is a huge time-saver, and also makes the MOGRTs a lot easier to use.

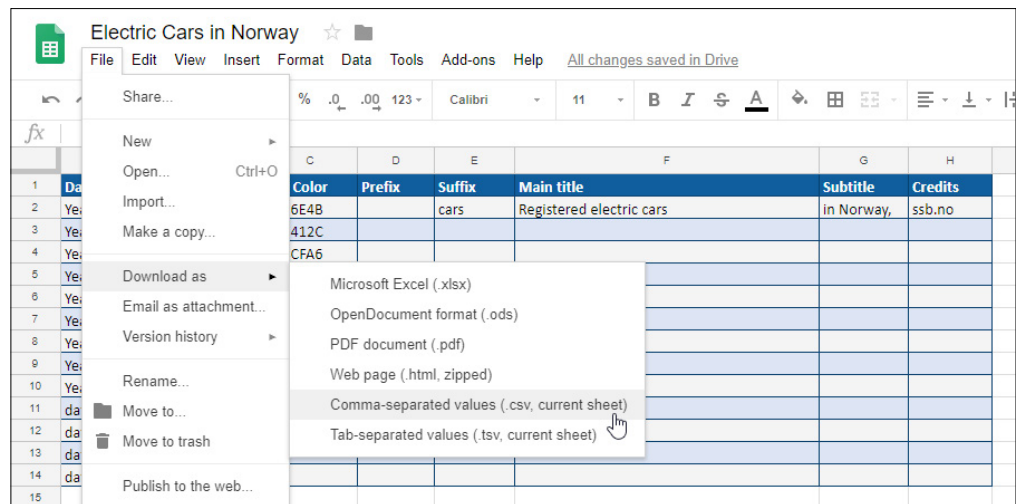
Updating your existing infographics MOGRTs to use the new data controls is easy. If you already have a well-rigged MOGRT, adapting it to the new spreadsheet controls is not too much work. All you have to do is connect the different properties to the corresponding properties in the subsections in the Data section of the CSV or TSV layer.

While you're building your MOGRT in After Effects, you can use data controls in the Essential Graphics panel as a data browser and the editor for CSV or TSV data layers in your composition.

Beware of CSVs made in Excel

I have tried using Excel to create my CSVs, but it seems to export noncompliant CSVs (missing line breaks) by default. Google Docs exports CSVs with the correct specs, so I have been uploading my Excel spreadsheets to Google Sheets and exporting CSVs from there.

Export CSV file from
Google Sheets.



If you don't want to use Google Sheets, you can convert the Excel file to a compliant CSV using one of the online file converters, like CloudConvert (<https://cloudconvert.com/xlsx-to-csv>).

A	B	C	D	E	F	G	H
Data Label	Data value	Bar Color	Prefix	Suffix	Main title	Subtitle	Credits
Year 2010	3000	#226E4B		cars	Registered electric cars	in Norway,	ssb.no
Year 2011	6000	#15412C					
Year 2012	10000	#94CFA6					
Year 2013	18000	#6DBF92					
Year 2014	30000	#F47B779					
Year 2015	45000	#226E4B					
Year 2016	62000	#15412C					
Year 2017	98000	#94CFA6					
Year 2018	142000	#6DBF92					

Note that this example has cells for *Main title*, *Subtitle*, and *Credits* in the first row. For a MOGRT you want to sell on Adobe Stock, you probably shouldn't require any titles to be in the spreadsheet. I'd use normal source text instead. But for corporate customers putting them in a spreadsheet is a good thing, since everything in the MOGRT then can be controlled by the spreadsheet data.

Controlling colors with a spreadsheet

The `hexToRgb()` expression converts hexadecimal color values to RGBA color values. This can be useful when you want to control color with hex strings in data sources.

Any of the following will return [1.0, 0.0, 1.0, 1.0]:

```
hexToRgb("FF00FF")
hexToRgb("#FF00FF")
hexToRgb("0xFF00FF")
hexToRgb("0XFF00FFFF")
```

In the last example, the last two digits set alpha to 1.0. If you don't specify an alpha value, it defaults to 1.0. This expression is not available in After Effects 15 and earlier versions.

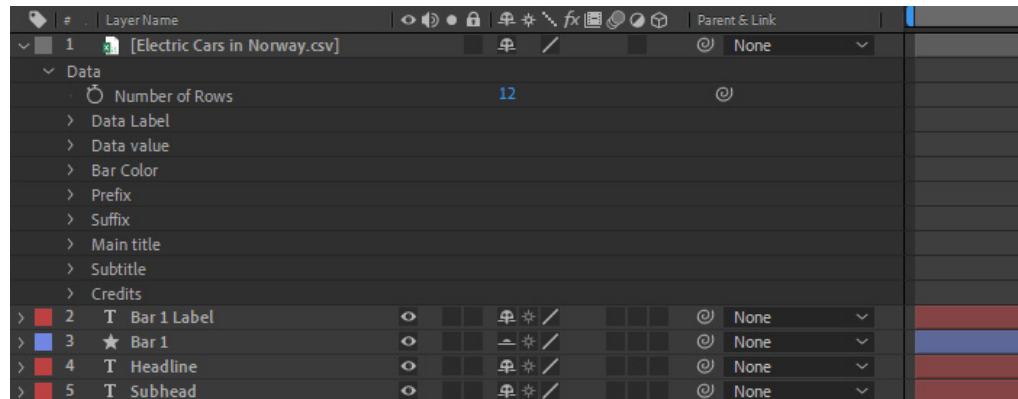
Hex color values

Color values must be represented in the spreadsheet by hexadecimal values starting with a hashtag sign. The hashtag sign is required, for example, #000000 for black or #15412C for a darkish green. Premiere Pro will show a color picker and a hex number for spreadsheet cells that have their Data property type set to Color.

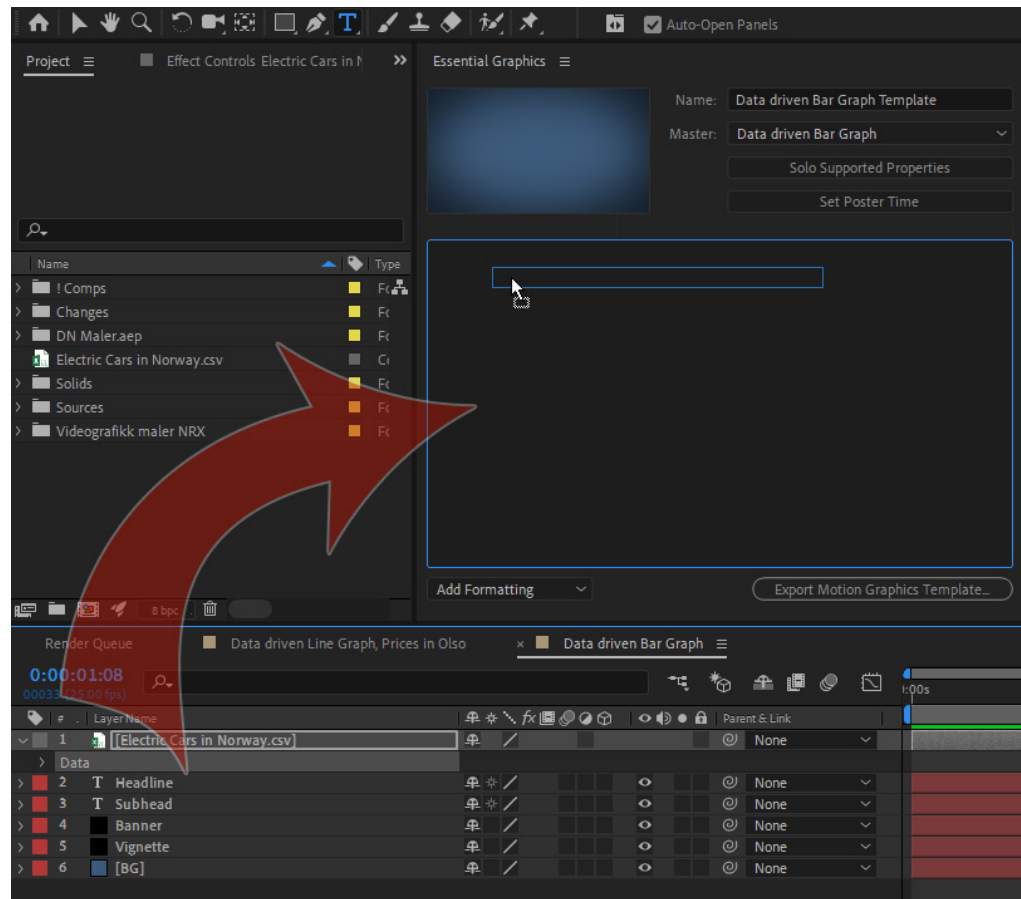
If there is no hex number in a cell, the color defaults to black.

To add spreadsheet data to your comp, import a CSV or TSV spreadsheet file, drag it to the timeline, and then drag the *Data* group up to the Essential Graphics panel. This will display the column names and the entries from row 0 in the spreadsheet.

This is how a CSV file
looks in the timeline.



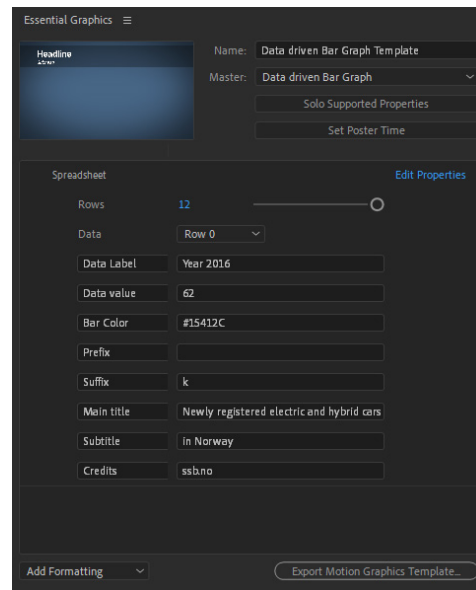
Drag the Data group up
to the Essential Graphics
panel.



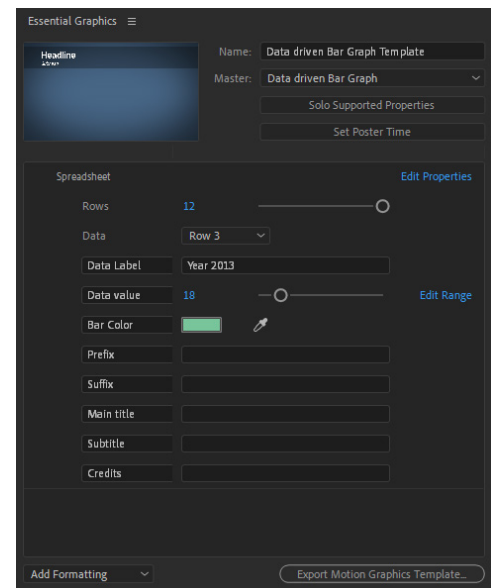
Note that adding a data layer's Data property group to the Essential Graphics panel will remove the expressions linking the layer's data properties to the source data file and change them to static values.

However, if you modify the source data file outside of After Effects, the static data values will still automatically be updated.

You can choose which row you want to view and edit. To edit the values for row 3, choose row 3 in the Data Row drop-down menu. Just remember that the counting starts with 0, so row 3 will be the fourth row in the spreadsheet. Yeah, I know. Programmer's logic.

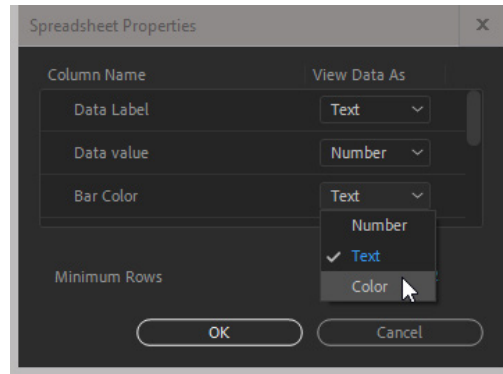


Choose which row you
want to view and edit.



To edit the Spreadsheet Properties, click the *Edit Properties* button to the right of the spreadsheet data group to access the following dialog box.

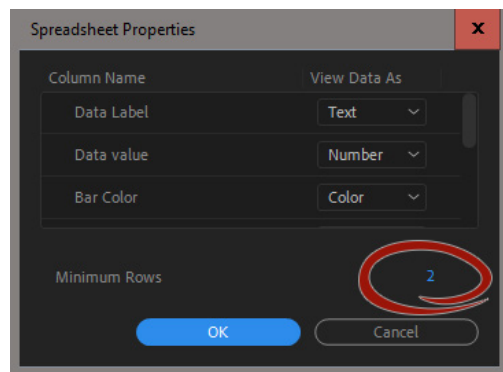
Choose data type for each column.



Here you can set the data type for each column. Three data types are supported: text, color, and numbers. In Premiere Pro this will translate to either a text string, a hot text slider for numbers, or a hex value (for example #C98E44) with a color picker.

The Rows slider lets you control how many rows from a spreadsheet the MOGRT should use. The maximum number of columns you can set with this slider is limited by the spreadsheet you use to create the template. The minimum number of rows can be set in the Spreadsheet Properties.

Set the minimum
number of rows in the
Spreadsheet Properties.



To link to the hex values in the data and convert them to RGB values, you must first convert the data to a string with the `.toString()` expression, and then use the `hexToRgb` expression to output the RGB values.

Here's an expression I used on the *Fill Color* property of a bar (a shape layer) to link it to a cell with hex values in the spreadsheet named *Electric Cars in Norway.csv*.

```
barColor = thisComp.layer("Electric Cars in Norway.csv")("Data")("Outline")
("Bar Color")("Bar Color 0").toString();
hexToRgb(barColor);
```

Use `parseFloat()` expression to fix number problems

CSV values are sometimes saved as just text, even though some values are actually numbers. This will cause some problems. Also, since the maximum value of a number control is 32768, values higher than this are simply cut off at 32768, which is obviously very bad.

In these cases, the `parseFloat()` expression comes in handy. It converts the text to numbers, so you can reach them with expressions.

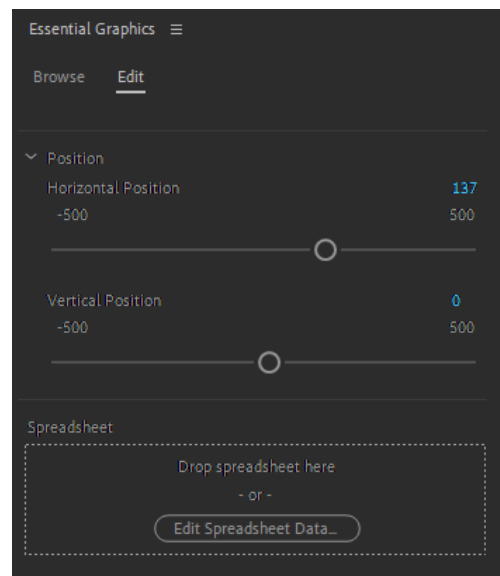
Here's an expression I used on the Size value of a bar (a shape layer) to make it scale according to the values that were interpreted as text strings.

```
x = 70;
barMaxHeight = 600;
maxValue = parseFloat(thisComp.layer("Electric Cars in Norway.csv")("Data")
("Outline")("Max value")("Max value 0"));
barValue = parseFloat(thisComp.layer("Electric Cars in Norway.csv")("Data")
("Outline")("Data value")("Data value 0"));
barHeight = barMaxHeight * barValue / maxValue;
[x, barHeight]
```

Using the data-driven MOGRT in Premiere Pro

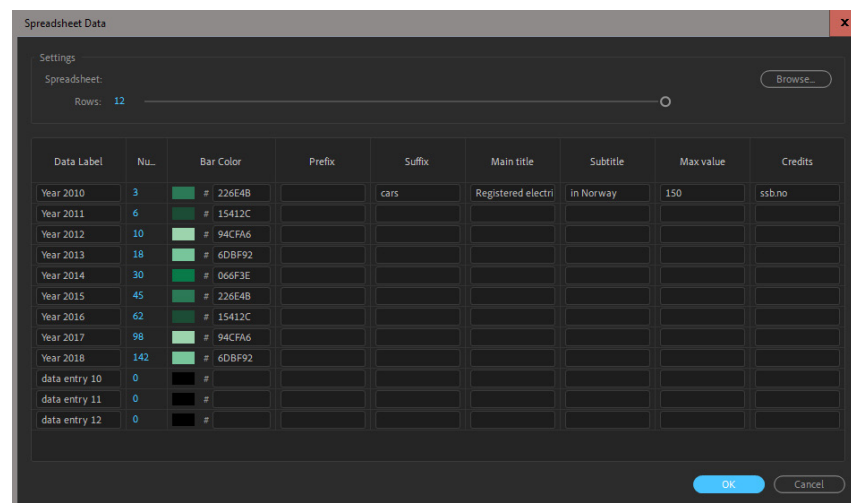
When you export a composition with data controls as a MOGRT and add that template to a sequence in Premiere Pro, you can modify the data values manually or replace the data with a different source file. With the MOGRT clip selected, you will see a drop zone in the Essential Graphics panel where it says: "Drop spreadsheet here or Edit Spreadsheet Data."

The spreadsheet drop zone in the Essential Graphics panel.



You can drag a CSV file from Finder/Explorer or from the Project panel in Premiere Pro. This opens the spreadsheet settings, where you can view and manipulate the mapping in a spreadsheet-like view.

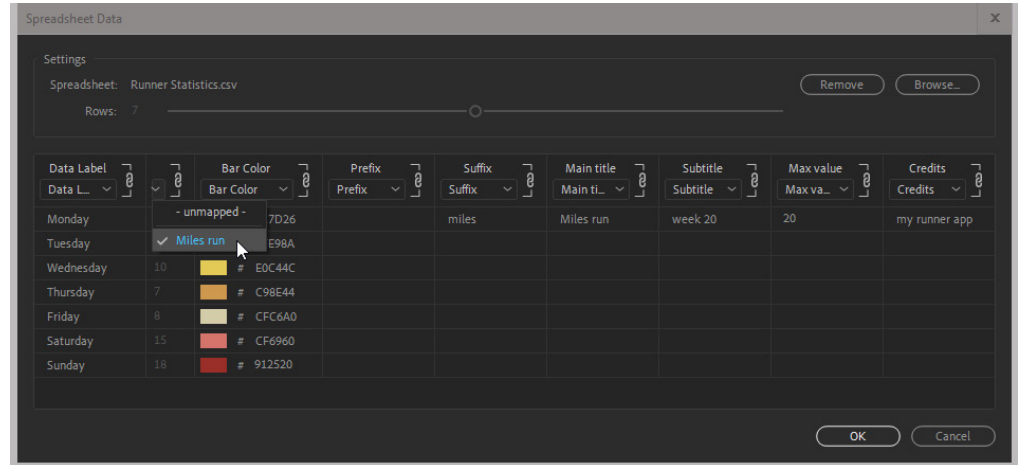
The Spreadsheet Data settings panel looks very much like a spreadsheet.



Premiere Pro will only auto-map the data of a spreadsheet to a MOGRT correctly when the column headings match. If the headings do not match, you can map them manually.

You can map columns manually if the headings don't match.

When replacing the data file, if the number of columns and column names in the new data file do not match, Premiere Pro will allow you to use the new or changed rows or to leave them unassigned.

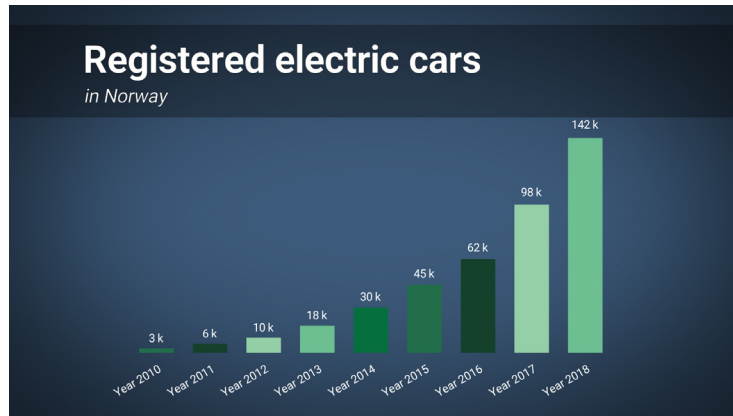


The mapping can be done on a column-for-column-basis; that is, one column can read the data from the spreadsheet and therefore be locked for manual changes, while another column can be unmapped to enable manual changes.

You can also click *Browse* to replace the data with a different source file. When the spreadsheet is updated, the mapped columns in the MOGRT will also be updated. Note that if the spreadsheet has more columns than the template allows, only the first rows will be used by the MOGRT.

You can get to the spreadsheet controls whenever you want. Just click the *Edit Spreadsheet Data* button.

This is the same MOGRT with two different CSVs assigned. Everything is controlled by the spreadsheet.



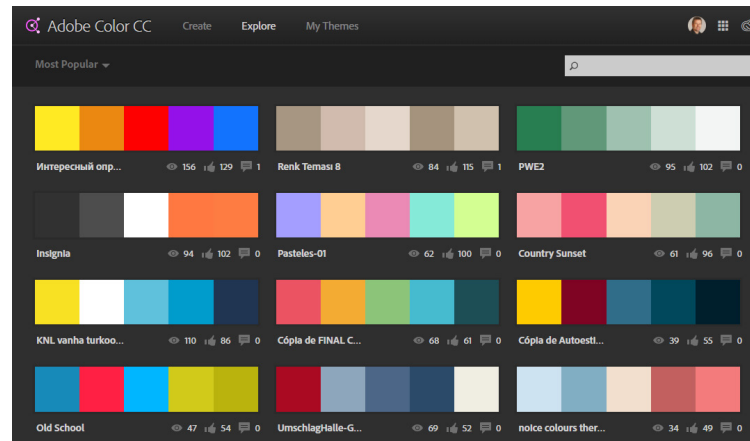
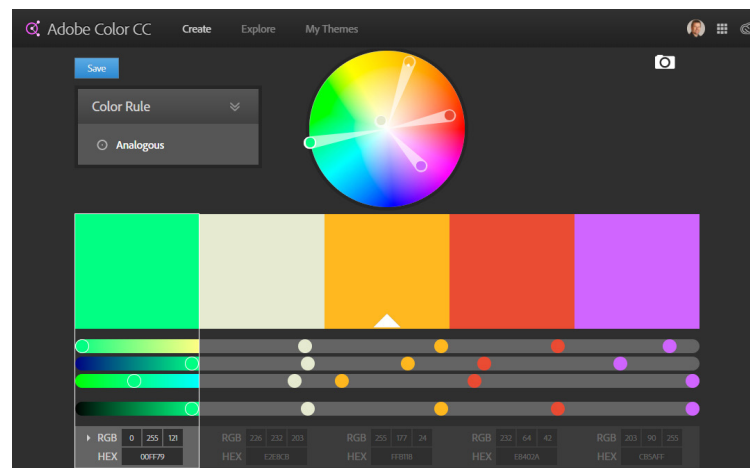
MOGRT design tips

This chapter is a collection of methods and workflows that lets you design your MOGRTs more quickly. If you're an experienced After Effects user, you probably know most of them. But I'm often surprised that seasoned After Effects artists sometimes don't know about some of the most basic features.

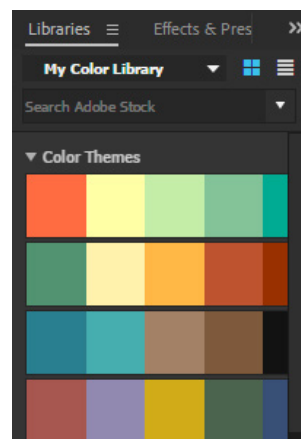
Use Adobe Color

It can be hard to find good color combinations for your MOGRTs. My favorite place to get inspired and to create my own color themes is Adobe Color at <https://color.adobe.com/>. You can browse the themes that others have created or use their color themes builder to create your own theme.

Adobe Color CC lets you create, explore and save color themes.



You'll find your color themes in the Color Library.



When you save a theme on the web page, it syncs to your Color Library in After Effects, so you can use the color picker to pick those colors when you create shape layers and text and when you add Color Controls.



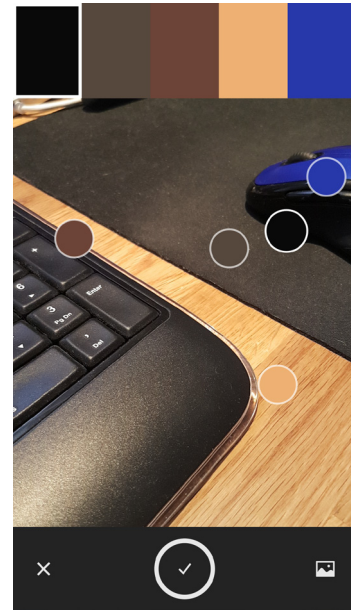
The Adobe Capture app

You can also use the Adobe Capture smartphone app to create color palettes and themes from images. If you see a painting you like, or you feel that the autumn colors in your favorite forest are worth saving, just point the smartphone camera to it and capture the colors. The themes you capture with the app are automatically synced to your library.

The app can also capture shapes and vectorize them, save textures, and other cool stuff.

<https://www.adobe.com/products/capture.html>

The Adobe Capture app can turn even the most boring image into a color theme, and you can easily adjust the colors before you save them.

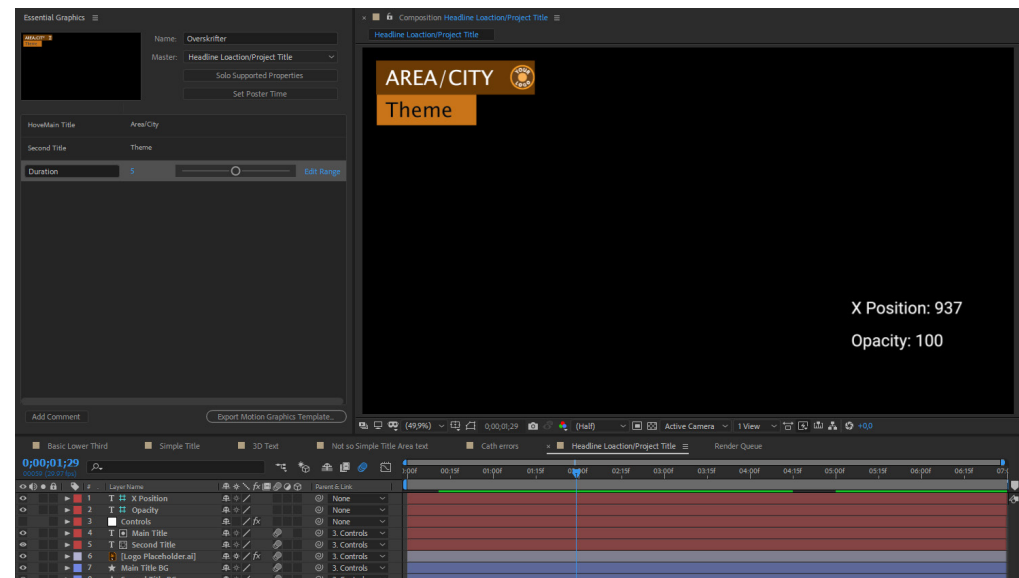


Use Property Reference text layers

A nice new feature in the spring 2018 version of After Effects is the Property Reference text layers. Just drag any property to the Composition panel, and After Effects will create a text layer that displays the property's current value. The text layer is automatically set as a guide layer, so it will never be output in the MOGRT if you should forget to delete it before you export a MOGRT.

For example, dragging a layer's Opacity property to the Composition panel creates the text, "Opacity: 100" or whatever the value is at the current frame. The value will reflect any expressions that affect the property.

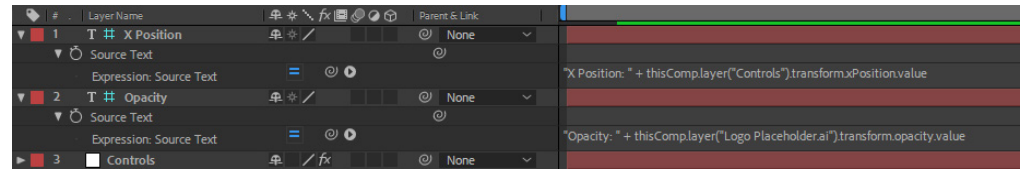
Here I've dragged Position from the Controls layer and Opacity from the logo layer to the Program Monitor.



This is great when you're testing and troubleshooting and temporarily need a readout of the resulting value of your expressions. Text, numerical values, 2D and 3D point and scale, angle, checkboxes, and pop-up menus are supported. Color properties are not supported.

There's no info about which layer the property comes from, but all it does is link the source text to the chosen property with an expression. So, you can easily change the text inside the quotation marks from, say, "Opacity: " to "Logo Opacity: " or something similar.

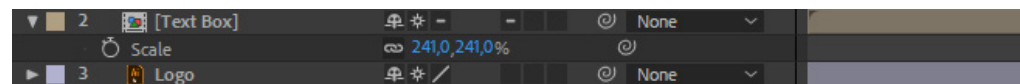
The feature adds expressions to the new text layers and makes them guide layers.



Use Collapse Transformations

When you're nesting a comp and want to scale it, make sure the *Collapse Transformations* button is on for the layer, or else it will get blurry when you're scaling up. It's the button that looks like a sun, to the right of the shy guy. This button is also important for keeping vector layers sharp when scaling, but on vector layers it's called *Continuously Rasterize*. Beware of a [known issue with Master Properties and Collapse Transformations in the 15.1 version](#), though.

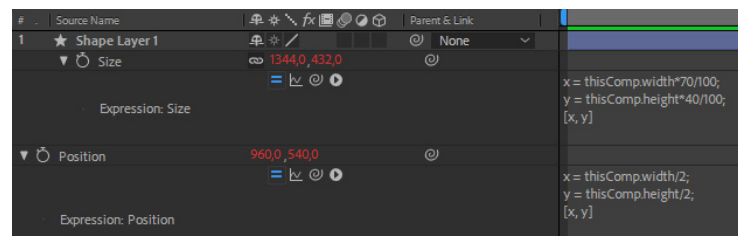
To make vector art and nested comps render sharply, make sure the "sun" button is on.



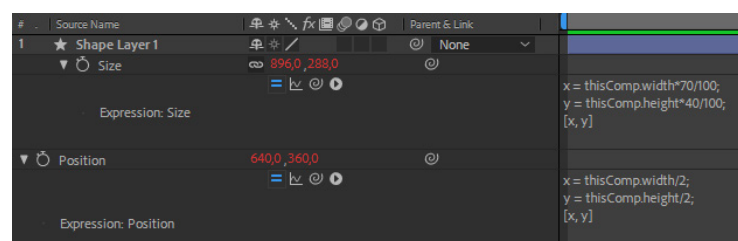
Make the MOGRT resolution independent

If you place your layers relative to the frame size instead of entering absolute pixel values, they will automatically adjust to different frame sizes. We use comp width and height to make values frame size agnostic. Copying code that centers or scales a layer using comp size will always work, no matter what size it is. 1920x1080, 1280x720, 1080x1080, 4k x 4k, and so on.

The size and position of this shape layer adapts to the comp size. This comp is 1920x1080.



The same shape layer in a 1280x720 comp. Note how the position and size have changed.



The shape layer fills the same percentage of the screen in comps with different comp sizes.



Here's the code for the shape layer Size property.

```
x = thisComp.width*70/100;  
y = thisComp.height*40/100;  
[x, y]
```

And here's the code for the shape layer Position property.

```
x = thisComp.width/2;  
y = thisComp.height/2;  
[x, y]
```

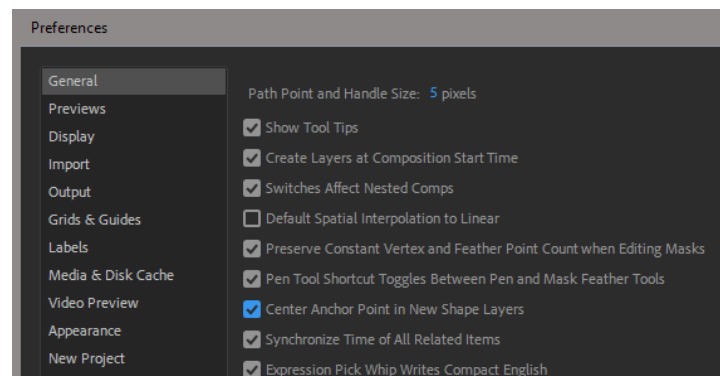
Center anchor point in new shape layers

The 15.1.0 (CC 2018.1.0) version of After Effects introduced a preference to center the anchor point of a shape layer in the first shape you create on that layer. This is very useful when you want to scale, rotate and move the shape layer.

The anchor point is the point that a layer will scale from, rotate around, and so on. When you're adjusting the Position parameter of a layer, it's the position of the anchor point you're controlling.

Choose **Preferences > General > Center Anchor Point in New Shape Layers** to enable this option. It's disabled by default.

This preference will save you the extra step of moving the anchor point the next time you draw a rectangle or a circle.



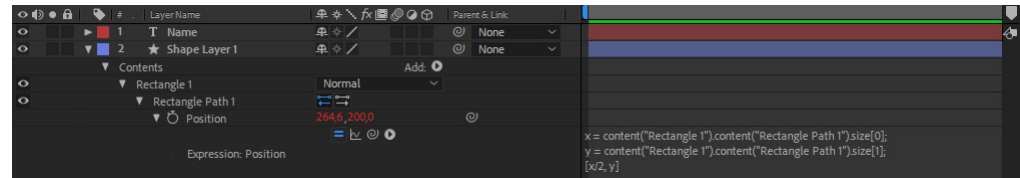
Place anchor points where you need them

Okay, this is a bit misleading because when you make a new shape layer rectangle, each rectangle doesn't have an anchor point. They only have Size, Position, and Roundness settings for each shape. So, instead of moving the anchor point, it's pretty common to move the position of the shape, relative to the size. This code on the Position parameter of Rectangle Path 1 makes it scale from the left.

```
x = content("Rectangle 1").content("Rectangle Path 1").size[0];  
y = content("Rectangle 1").content("Rectangle Path 1").size[1];  
[x/2, y]
```

Since this Position parameter is now controlled by the expression, you must use the normal Position parameter to place the layer where it should be.

The code should be on the Position parameter inside Rectangle Path 1.



The layer now scales from the left edge.



If you want it to scale from the right, add a minus sign before the x in the last line of code.

```
x = content("Rectangle 1").content("Rectangle Path 1").size[0];  
y = content("Rectangle 1").content("Rectangle Path 1").size[1];  
[-x/2, y]
```

The rectangle now scales from the right.



The following code makes the rectangle scale from the bottom:

```
x = content("Rectangle 1").content("Rectangle Path 1").size[0];  
y = content("Rectangle 1").content("Rectangle Path 1").size[1];  
[x, -y/2]
```

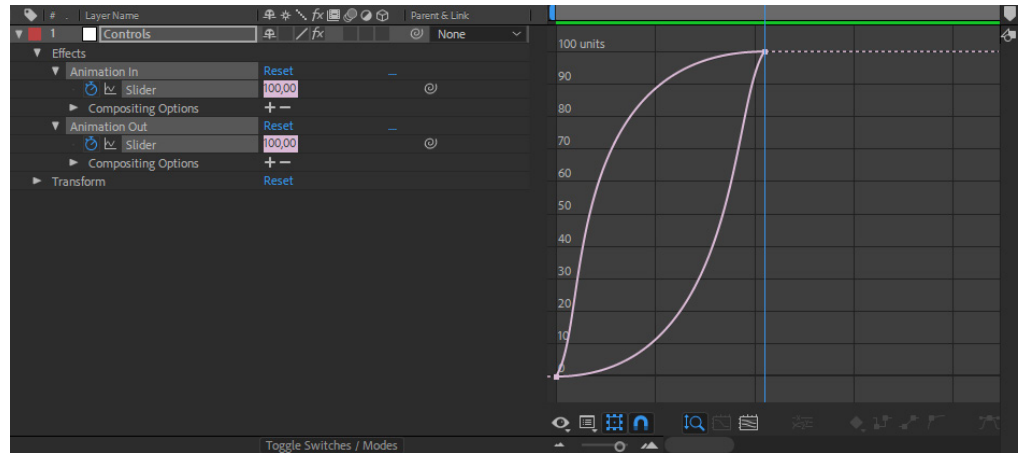
And to make it scale from the top:

```
x = content("Rectangle 1").content("Rectangle Path 1").size[0];  
y = content("Rectangle 1").content("Rectangle Path 1").size[1];  
[x, y/2]
```

Use one custom animation in many layers

All the animations can be controlled with the same two keyframed animations, animating your layers in and out. Just make two sliders, set keyframes at 0 and 100, and adjust the animation curve with the Graph Editor.

These two animations can control all the others.



I often add two more sliders for setting the in-animation time and the out-animation time and use the following code to control everything.

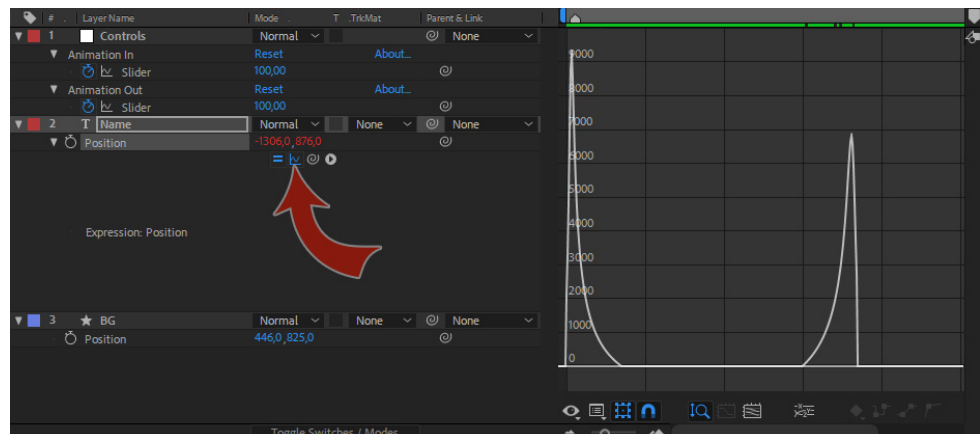
```
tIn = thisComp.layer("Controls").effect("In-animation start time")("Slider");
tOut = thisComp.layer("Controls").effect("Out-animation start time")("Slider");
animIn = thisComp.layer("Controls").effect("Animation In")("Slider").valueAtTime(time+tIn);
animOut = thisComp.layer("Controls").effect("Animation Out")("Slider").valueAtTime(time-tOut);
x1 = linear(animIn, 0, 100, value[0]-1500, value[0]);
x2 = linear(animOut, 0, 100, value[0], value[0]-1500);
if (time < 1) x = x1 else x = x2;
y = value[1];
[x, y]
```

Yeah, that's a lot of code. But now the same custom ease animation can be used on all the layers, and it's super easy to change the timing by adding or subtracting time and by dragging the sliders. Sometimes the numbers (-1500) would also be replaced by a slider value, so all layers can move the same way.

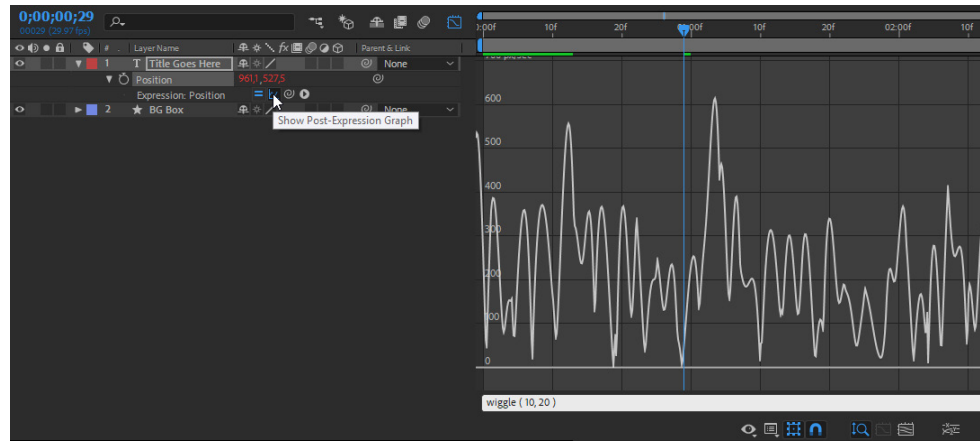
Get a visual representation of the animation

A button called the Post Expression Graph button shows the result of expressions in the Graph Editor. It looks like a little line graph. This is extremely helpful when you're troubleshooting and want to see the timing of the animation with the expressions.

The Post Expression Graph for the custom animation.



The visual representation of a wiggle expression.

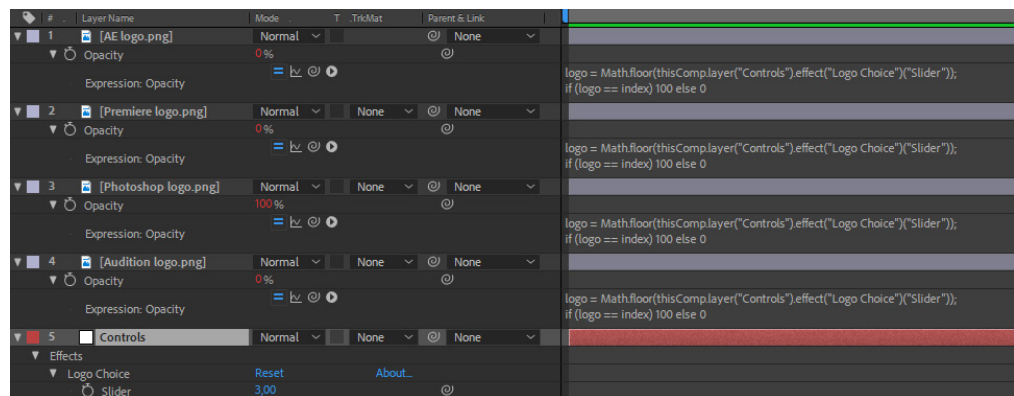


Reuse code with index

You can use the *index* expression to control visibility for all layers with the same code. Let's use a logo choice as an example. Make a *Logo Choice* slider and paste the following code to the *Opacity* expression on each layer. The word *index* means the layer number. Since each layer has a different layer number, we can use this in our expression, and even though the expression is the same, the behavior changes for each layer. This is a great time-saver.

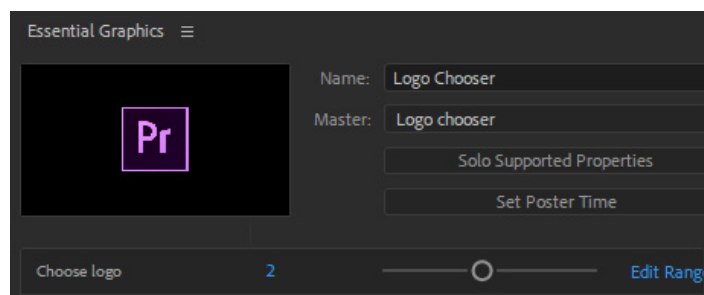
```
logo = Math.floor(thisComp.layer("Controls").effect("Logo Choice")("Slider"));
if (logo == index) 100 else 0
```

All the layers have the same code.



Note that I placed the Controls layer at the bottom. I could have placed it at the top and used *index-1* instead of *index* in the code. This method can be used on any type of layer, not just logos, and on many other properties.

In this MOGRT, the user can choose which logo to show by dragging the slider.



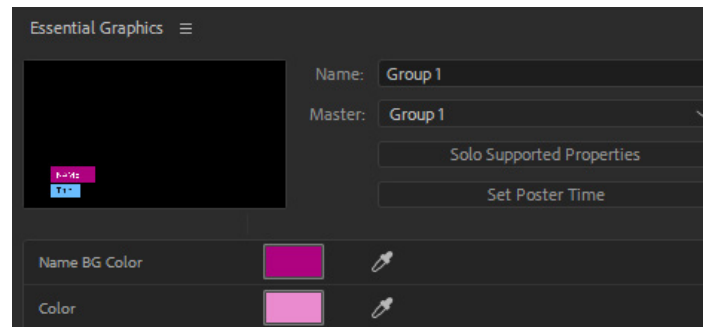
Use Master Properties

The 15.1.0 (CC 2018.1) version of After Effects introduced Master Properties. Master Properties allow you to access properties from a comp when it is nested in another comp. This saves a lot of time because you don't have to dive in and out of comps to access their properties.

Creating Master Properties

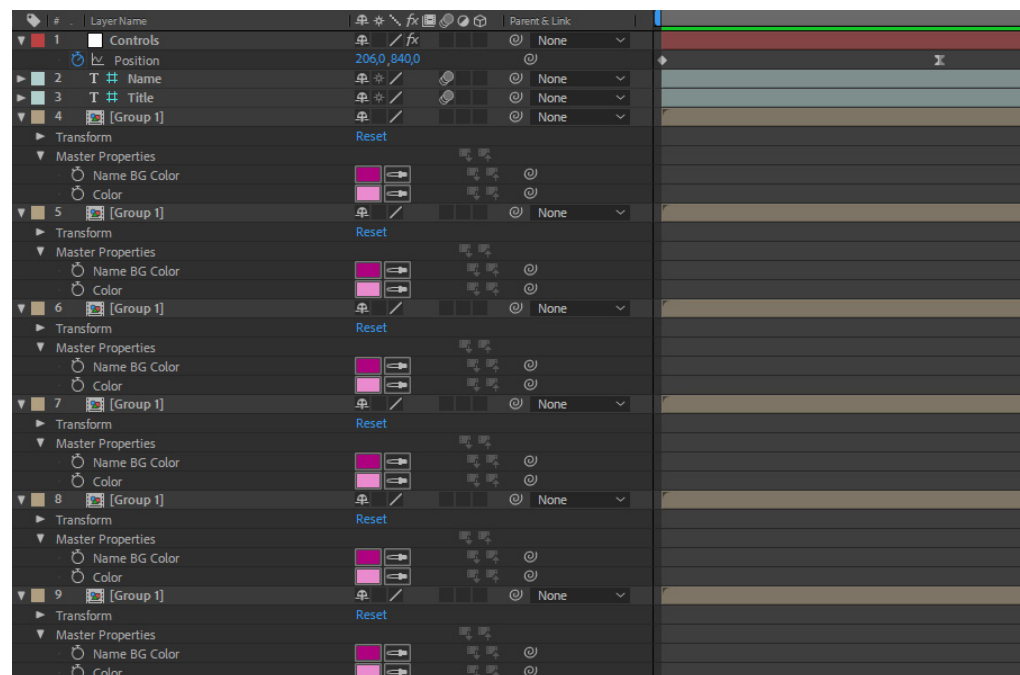
You create Master Properties by adding properties from a nested composition in the Essential Graphics panel, or by nesting the composition after you add the properties ([Layer > Precompose](#)).

For the Group 1 comp,
I added the Fill Color of
the two rectangles to the
Essential Graphics panel.



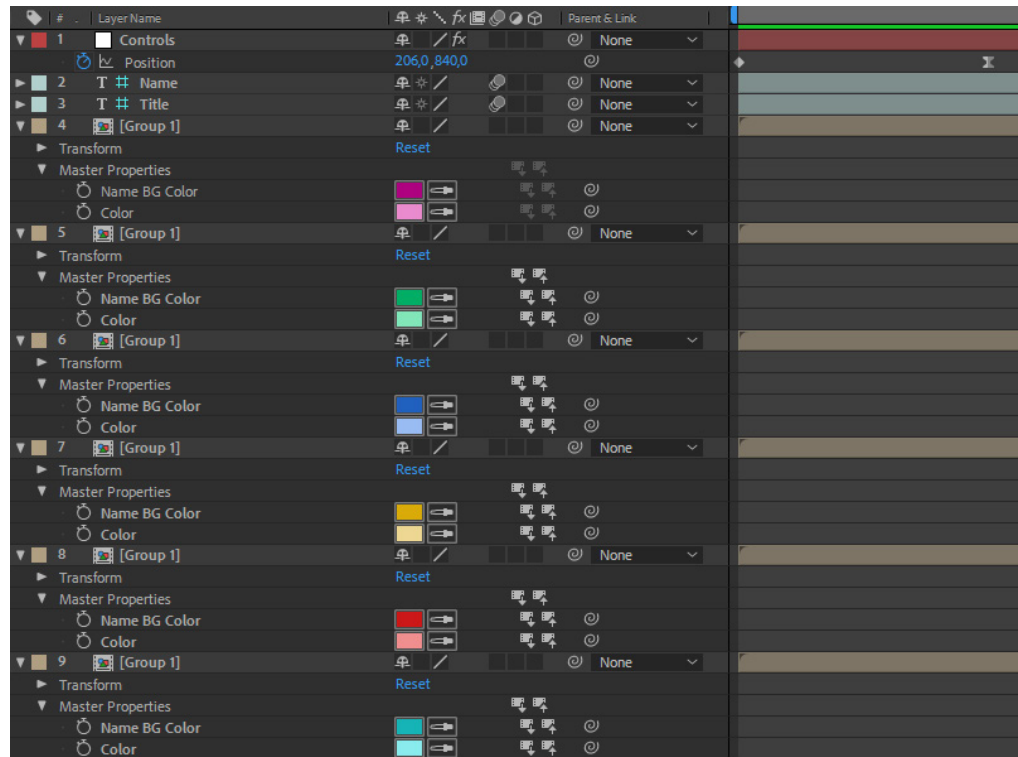
When you nest the comp, the Timeline panel displays the *Master Properties* group of the nested comp.

When I nested and
duplicated the comp, the
Fill Color choices appeared
under Master Properties in
the main comp.



When you change a master property, it will override the original values and keyframes, which allows you to reuse a comp multiple times with different values. The nested comp is now a unique instance of the master comp, and only this instance renders with the modified value or keyframes. The source comp remains untouched. This means that you no longer have to create six duplicate comps when you need six different instances of it.

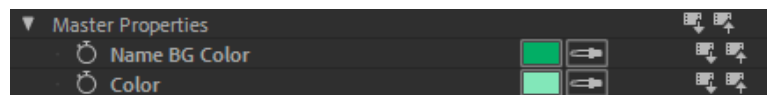
Using Master Properties,
I could use six instances of
the same comp with differ-
ent color choices.



Pull and Push buttons

When you modify a master property, the Pull and Push buttons in the Timeline panel are enabled, indicating that a property is unlinked. The Pull and Push buttons relink a master property with the source property, which means that after you click either the Pull or Push button, master properties again link to their source properties and disable the Pull and Push buttons.

When you modify
a master property, it
overrides the value or
keyframes of the source
property, and the Push
and Pull buttons on the
right are highlighted.



Pull discards the modified value, keyframes, and expression of a master property.

Push applies the modified value, keyframes, and expression of a master property to the source property.

So, in my case, with the color choices, I could change the color of my instance back to the original color by clicking the *Pull from Master Comp* button, or push the new color to the master by clicking the *Push to Master Comp* button.

There are Pull and Push buttons for each master property and for the entire Master Properties group. The way the buttons for the entire group work depends on what's selected before you click them. If no properties are selected, all the properties in the group will be affected. If one or more properties are selected, only those properties will be affected by the group buttons.

Known issues with Master Properties

Master Properties are great and will change the way you work in After Effects, but there are a few issues with this new feature that you should be aware of. Here's a list of known issues with Master Properties in the 15.1.0 (CC 2018.1.0) version of After Effects from Adobe's knowledge base.

<https://helpx.adobe.com/after-effects/kb/known-issues-after-effects-cc.html>

- After Effects does not allow you to copy Master Properties and keyframes to other properties or layers.
- After Effects does not allow you to add properties to the Essential Graphics panel from the Master Composition group of a nested composition.
- In a composition with multiple copies of the same nested composition with master properties, if you enable the Collapse Transformation switch for any of the copies, the other copies can render incorrectly. The copies render with the modified master property values, unless you modify the properties individually.
- When you modify the value of a Time Remap master property, After Effects is unable to return the correct time value in the master composition.
- When you modify a master property for an audio effect or property such as Tone, the audio playback does not reflect the changes.
- If your project contains multiple master properties by the same name, the expressions that refer to the master properties evaluate incorrectly.

Even with these issues, I strongly recommend that you slim down your projects with Master Properties whenever you have a chance. And my guess is that most of them will be fixed in future updates. Check the link in a few weeks or months, and they may have been addressed already.

Use standard procedures, work faster

There are some features in After Effects that have "always been there," but they're still very important when you need to work fast.

Use folders to always keep your Project panel tidy.

Name everything, or you'll go crazy.

Use Guides, Rulers, Grid, and the Align panel to place layers precisely.

Use labels to bring order to your 100-layer timeline.

Shy the layers you don't need to see.

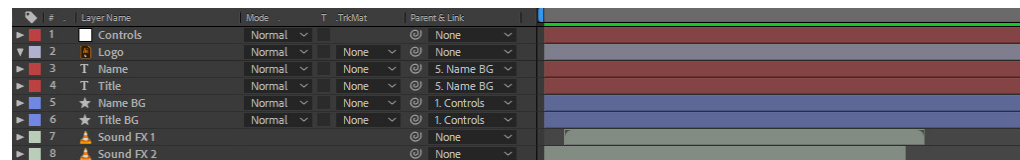
Parent layers and animate only one (but use *toComp expressions*).

Nest comps (and activate Collapse Transformations).

Add audio to your MOGRTs

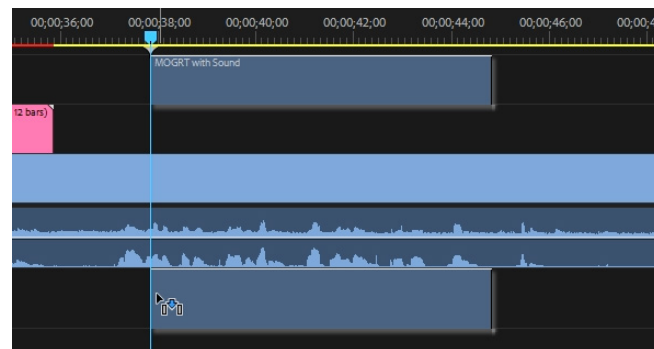
You can add music, Sound FX and any other audio to your MOGRTs. Sound FX often adds to the experience of the animation, so it can be worth trying a few whooshes, data-beeps, or other sounds in your MOGRTs. But be aware that at this time, Adobe Stock does not allow audio in MOGRTs for sale.

Use Sound FX in your MOGRTs.



To use MOGRTs with audio in Premiere Pro, drag the MOGRT from the Essential Graphics panel to the timeline, but don't let go yet. When the video part of the MOGRT is at the right time in the right track, press **Shift+Ctrl** (**Shift+Cmd**) and move the cursor to the audio track you want the audio clip on. The video will stay in its track, and only the audio will move.

Press Shift+Ctrl (**Shift+Cmd**) to lock the video to one track, and then move the audio to the desired track.



Just be aware that sound will not respond to expressions that control its timing. So even if the following code on *time-remapping* works fine in After Effects for moving the outro sound effect so it starts when the out-animation starts, the timing of the audio in the MOGRT will not work in Premiere Pro.

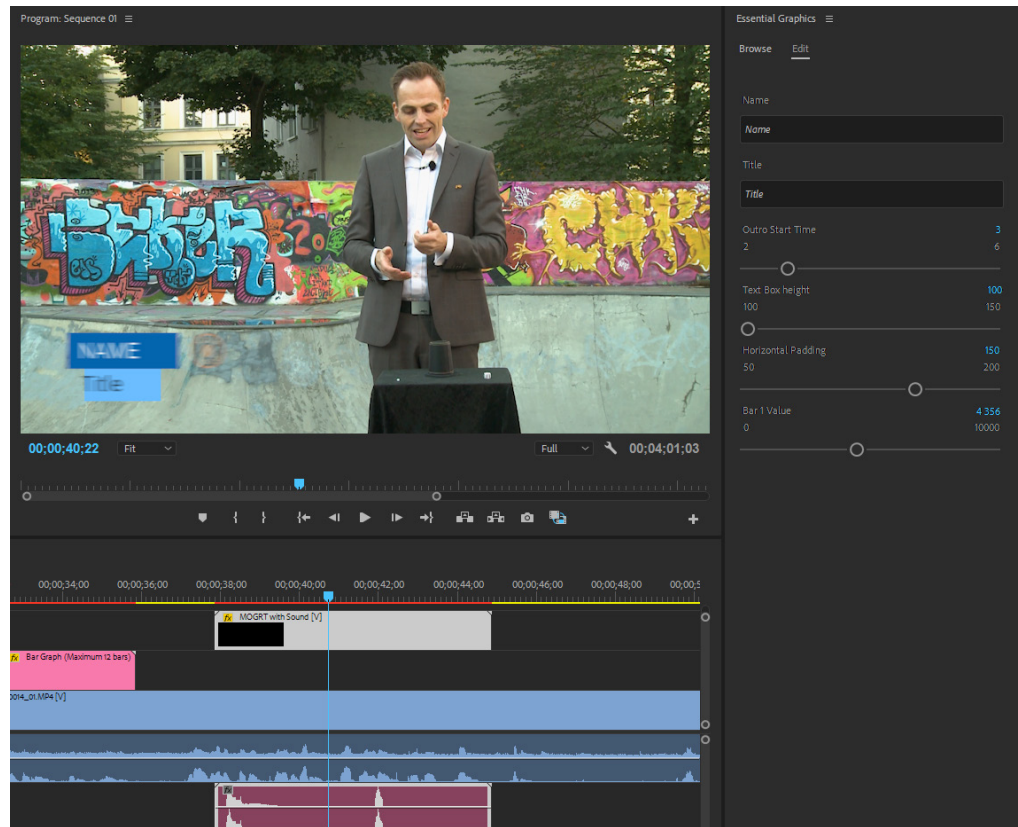
Audio in a MOGRT is not responsive. It will always have the default timing that it had in the After Effects timeline, regardless of what the settings are in Premiere Pro.

```
keyOut = 1;
tOut = thisComp.layer("Controls").effect("OutroStart")("Slider");
t = time - (inPoint + tOut);
valueAtTime(key(keyOut).time + t)
```

Take a look at the MOGRT in the following figure. The default outro time is 4 seconds, but in Premiere Pro, I changed it to around 3 seconds. You'd think that the audio would now be recalculated, so the sound FX comes at the time the layers move out. After all, this is the way it works in the After Effects timeline. Alas, the audio does not respond to the Time Remapping expression in Premiere Pro.

Moving the Outro Start Time slider in After Effects works as expected, but in Premiere Pro, the audio does not respond to the slider.

Here, I moved the Outro Start Time slider to around 3 seconds, so the out-animation starts there. The playhead is parked where the out-animation starts after dragging the slider. The audio waveform shows the delayed sound FX.



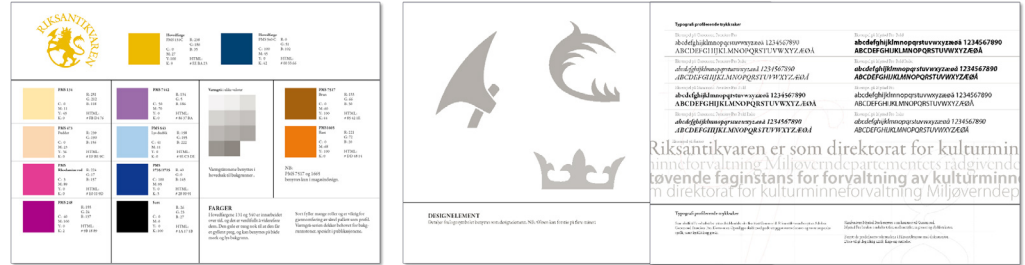
Making MOGRTs for your customers and colleagues

There's a big difference between making MOGRTs for sale on Adobe Stock and other sites and making them for a customer. MOGRTs for sale must be flexible and customizable. MOGRTs for a specific company must limit the choices to approved fonts, specific logo element sizes, company colors, and other style elements.

Their design manual is the bible

If it's a large company or organization, they will most likely have a design manual. Stick to this, or their design police will not approve your MOGRTs.

Sample pages from the design manual for the Norwegian Directorate for Cultural Heritage



Sample pages from the design manual for the Norwegian Public Roads Administration.

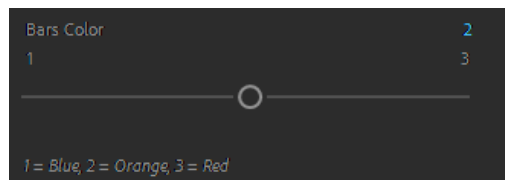


Choice of approved color combinations

Don't give the user Color Controls in a MOGRT where sticking to company colors is important. Instead, give the user limited color choices with Slider Controls, rounding, and if/else statements.

```
colorChoice = Math.round (thisComp.layer("Controls").effect("colorChoice")
("Slider"));
if (colorChoice == 3) 100 else 0
```

In this MOGRT, it's impossible to choose "illegal" colors.



One slider can also control colors on several layers. If the text is white, the background can't be white too, so it makes sense to automatically change it to black or dark gray and so on. The more layers one slider can control, the easier the MOGRT will be to use.

Choice of approved fonts

The same method can be used if the company has two or more approved fonts.

```
colorChoice = Math.round (thisComp.layer("Controls").effect("fontChoice")
("Slider"));
if (fontChoice == 2) 100 else 0
```

If the company font is available from both Adobe Fonts and other sources, use the one from Adobe Fonts. This ensures that everyone can automatically sync the font if they don't have it installed.

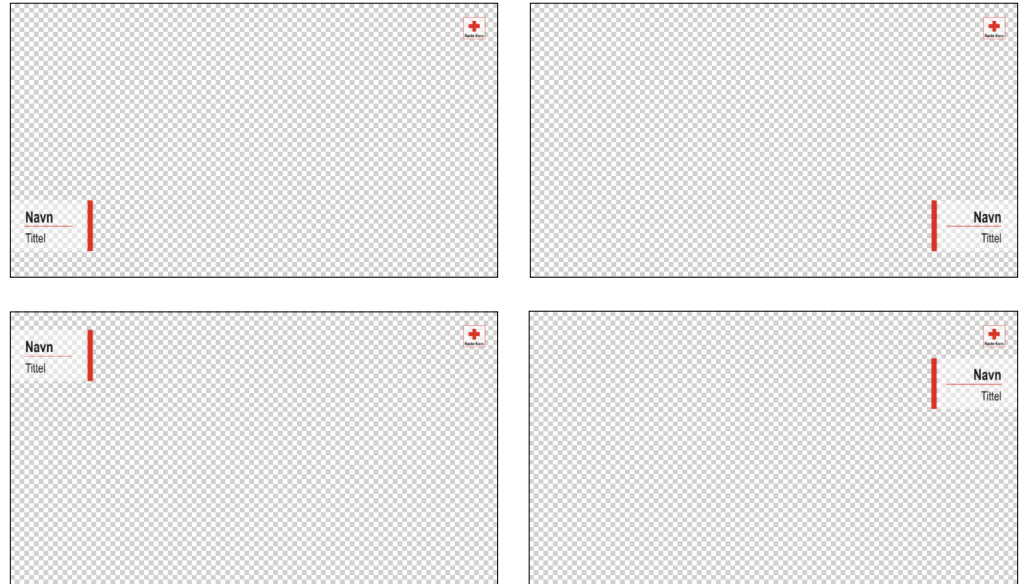
Make MOGRTs with built-in safety nets

Sometimes, it's nice to give the MOGRTs features that users didn't think they needed. The Norwegian Red Cross always puts their logo in the upper right corner, but the editor doesn't necessarily see this while editing, since the logo is the last thing they add.

We decided to build their template with simple restraints on the placement when the text is right-aligned. All the right-aligned layers were parented to the same null object, and the following simple *linear* expression on the Y-Position parameter of the null object was all it took to make sure the logo was never covered.

```
vPos = thisComp.layer("Controls").effect("Vertical Position")("Slider");
linear(vPos, -200, 500, -90, 500)
```

The name and title go further up when left-aligned. The positioning is restricted so they never cover the logo when right-aligned.



Beware of openers

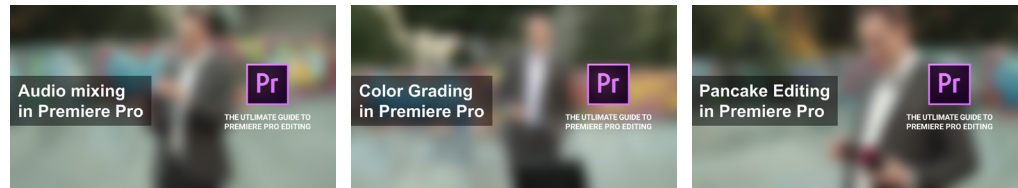
One of my customers wanted a standard opener, where their logo was revealed with a nice animation and then showed for about 2 seconds. They really liked the opener, but as I expected, they stopped using it after a very short time. Why? Because all their videos suddenly had the exact same cover image on YouTube and social media.

If your videos start with an opener with only your branding, all the video thumbnails on YouTube and social media will look the same.



They would have to manually choose a cover image on YouTube and social media, and many of their editors forgot, or didn't bother. So they stopped using it. A much better approach is to make an intro where the title shows clearly. Now every thumbnail is different, but still has the branding.

Using the title in the opener is one way to ensure that all the thumbnails are different without any manual interference.



If the client insists on showing their logo nice and big, try to convince them to use it in the outro instead, as part of the end credits, or as a standalone quick animation before fade to black.

Get access to their library

When you're making MOGRTs for a client, it makes for a very smooth workflow if they give you temporary access to their library. You will need both read and write access, but only until the MOGRTs are tested and approved. Then they can stop your access.

I've worked this way with many customers, and they find it very slick. I ping them and ask them to try the new version, and they immediately have access to it. If they want some changes, I do some adjustments and export a new MOGRT, which they again have full access to.

A good rule is to never work directly in the library they use for their daily editing. Create a new test library. Then when everything is approved, they can move the MOGRTs over to their standard library. In their Test library (in the Library panel in Premiere Pro), they can just right-click and choose **Move to > MOGRT Library**.

Optimizing MOGRT performance

Keeping your MOGRTs fast and responsive is a balancing act between features and flexibility, layer count, and the number of expressions. Since expressions are calculated for every frame, expressions that are calculated based on values from earlier frames will get slower for each frame. These expressions can slow a MOGRT down so much that it gets almost unusable. Following are some tricks to make the MOGRT perform better.

Keep the layer count down

The more layers you have, the slower the rendering will be, especially if you add Motion Blur and effects that take a lot of computing power. If you can use expressions to make one layer behave differently instead of using several duplicates with different animations, that will make the MOGRT faster.

Move layer in from left, right, top, or bottom

Say your client wants a logo to fly in from four different directions. Instead of using four layers that are animated differently and opacity expressions to turn their visibility on and off, you can use if/else expressions on one layer and make the logo move from different directions based on a Direction slider. Here's some code from Dan Ebberts:

```
horiz=0;
vert=0;
d = Math.round(thisComp.layer("Controls").effect("Logo Direction")("Slider").value);
if (d==1) horiz = 3050
else if (d==3) horiz = -3050
else if (d==2) vert = 2550
else if (d==4) vert = -2550;
offset = easeOut( time, inPoint, inPoint+1, [horiz,vert], [0,0] );
value + offset
```

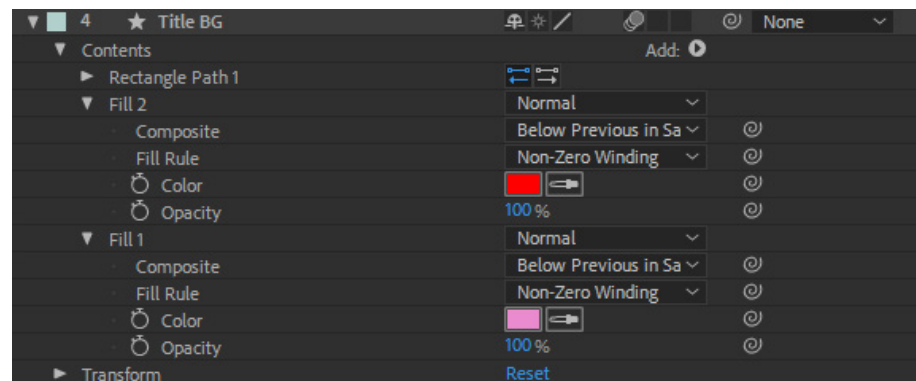
This code makes the layer ease into the layer's preset position from outside the frame in 1 second. The direction depends on a slider value.

Use multiple fill color instances

Instead of having five different colored shape layers and text layers in a comp, you can have one text layer and one shape layer, with five fill color instances. This will reduce the layer count from 10 to 2. Then add an expression to a parameter that shows or hides the color.

On a shape layer, click the Add button to the right of *Contents*; **Add > Fill**. This adds an extra Fill where you can choose the color. With expressions linked to a Color Choice slider, you can control the Opacity of each fill color.

Add several fill instances with different colors, and use expressions on the fill Opacity.

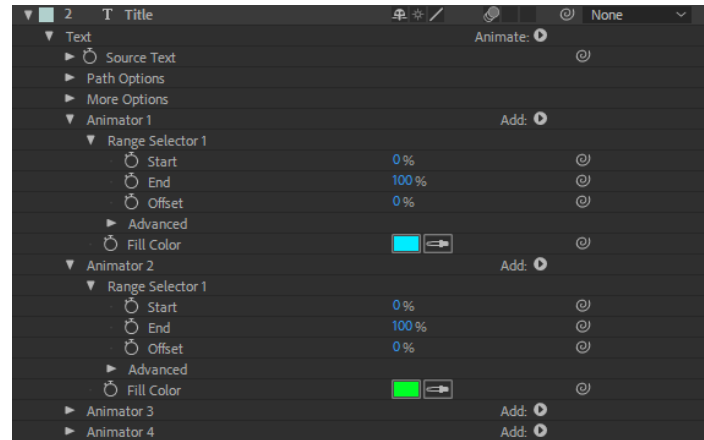


We can use our standard expression for controlling Opacity with a slider again.

```
colorChoice = Math.round (thisComp.layer("Controls").effect("colorChoice")
("Slider"));
if (colorChoice == 2) 100; else 0
```

For text layers, the approach is similar, but instead of adding a fill, we choose to animate the fill color. Click the *Animate* button to the right of *Text* and add an RGB fill; **Animate > Fill Color > RGB**. We could add Opacity to the animator, but we don't have to. We can control the *End* parameter on the Range Selector instead. At 100, all the text is colored, and at 0, no text is colored.

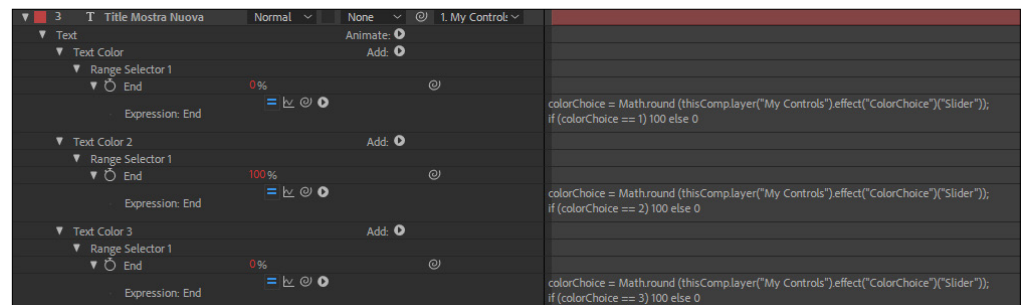
Duplicate the animator to get several colors to choose from.



We can use the exact same expression for the End property of the Range Selector as we did on Opacity on the text layer to link to the Color Choice slider.

```
colorChoice = Math.round (thisComp.layer("Controls").effect("colorChoice")
("Slider"));
if (colorChoice == 2) 100; else 0
```

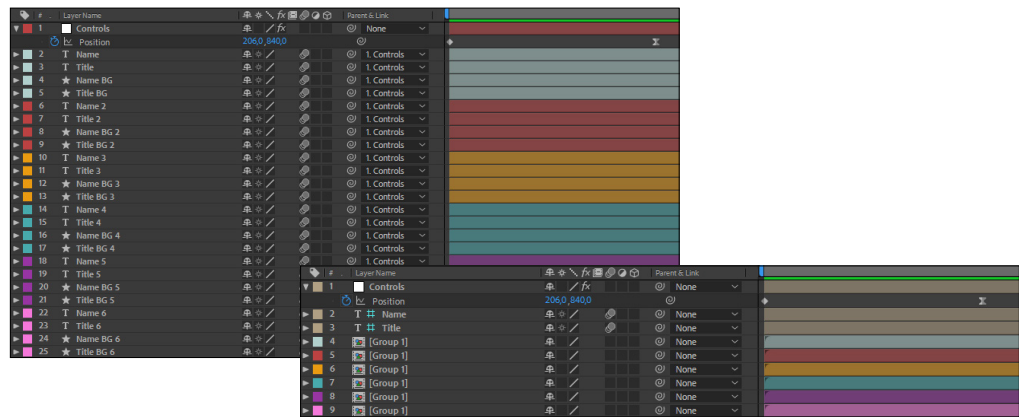
The End parameter of the Range Selector in each animator is now controlled by the Color Choice slider.



Use nested comps

Comps with lots of layers tend to render slowly. With the new Master Properties feature, we can often reduce the layer count drastically with nesting.

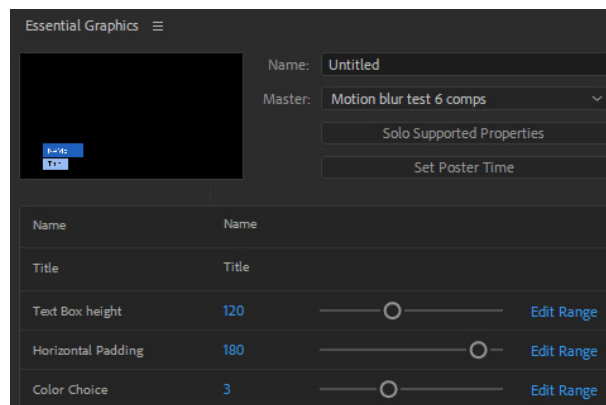
The comp on the left, with 24 layers of text and shapes, will render many times slower than the one on the right, where the layers are grouped into just one nested comp with four layers, used six times.



The comp on the left in the previous figure has 24 layers of shapes and text. This was done to be able to turn the layers on and off with expressions to change the text and shape layer colors. Each group of layers with the same colored labels belongs together. This is a highly ineffective way of doing this.

The comp on the right in the figure achieves exactly the same thing, but by reusing the same precomp six times, with Master Properties. The Essential Graphics panel looks the same for both, so the only difference for the user is that the slim one renders much faster in Premiere Pro.

Other than faster rendering, there's no change for the user.



We can use the same expression on Opacity for the nested comp layers as we'd use for all the layers in the comp with no nesting.

```
colorChoice = Math.round( thisComp.layer("Controls").effect("Color choice")
("Slider"));
if ( colorChoice == 1) 100; else 0
```

An expression on six comp layers instead of the same expression on 24 text and shape layers will calculate a lot faster.

The 15.1.0 version of After Effects has a known issue with Collapse Transformations and Master Properties that you should be aware of when you're nesting comps.

<https://helpx.adobe.com/after-effects/kb/known-issues-after-effects-cc.html>

Nest comps to give the user animation options

The same approach of nesting comps can be used for choosing from different animation options. Each nested comp would have a different animation, and you let the user choose the animation type with a slider that controls the Opacity of each nested comp.

There are more ways to keep the layer count down, of course, and these examples should get you on the right track. Another way to shorten the render time is to pre-render.

Reduce the render time with pre-renders

If you're creating backgrounds and textures with effects like Fractal Noise and Turbulent Displace, these effects will increase the complexity and affect the render time. There may be cases when they don't have to be adjustable in the MOGRT, and if that's the case, you should definitely consider pre-rendering them to a video file and using that in the MOGRT instead.

This will increase the size of the MOGRT file, because it will have a video file embedded in it, but at least for shorter MOGRTs, that's not a problem. The speed improvement makes it worth using a few extra megabytes of disk space.

Exporting and distributing MOGRTs

Before exporting your MOGRT, take the time to do some basic preflight checks to prevent problems with controls that don't work and links to missing properties.

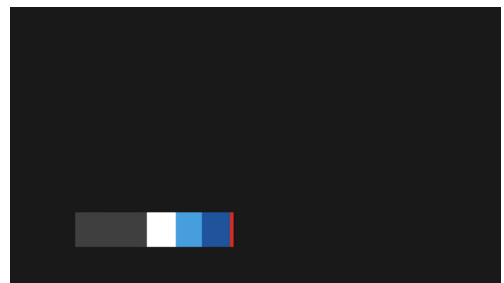
Use nesting to avoid problems when you export MOGRTs

There's a rule for what's included from other comps when you export a MOGRT:

Only compositions in the hierarchy of the master composition are included in the template.

So, a MOGRT that works perfectly well in After Effects may fail in Premiere Pro if it uses, say, a slider or a Color Control from a comp that is not nested in your main MOGRT comp. You can also get a completely different look after the MOGRT is exported. If the colors of four shape layers were taken from four Color Controls in an unrelated comp, they will all fall back to the default fill color, which is red.

This MOGRT used animated colored shapes to reveal the text. It looked good in After Effects, but when the MOGRT is used in Premiere Pro, all the fill colors go red. This is because the Fill Colors were linked to a Color Control in a non-related comp.



Before exporting a master composition as a MOGRT, you should nest all compositions containing unrelated properties into the master composition's hierarchy.

If you have a habit of creating a controls comp, make sure you nest it into the main comp and turn off its video switch before you export your MOGRT. This will avoid including controls that don't do anything in the template. If necessary, disable the video and/or audio switches for the nested composition layers.

Controls for unrelated properties will not have any effect when the composition is exported as a MOGRT.

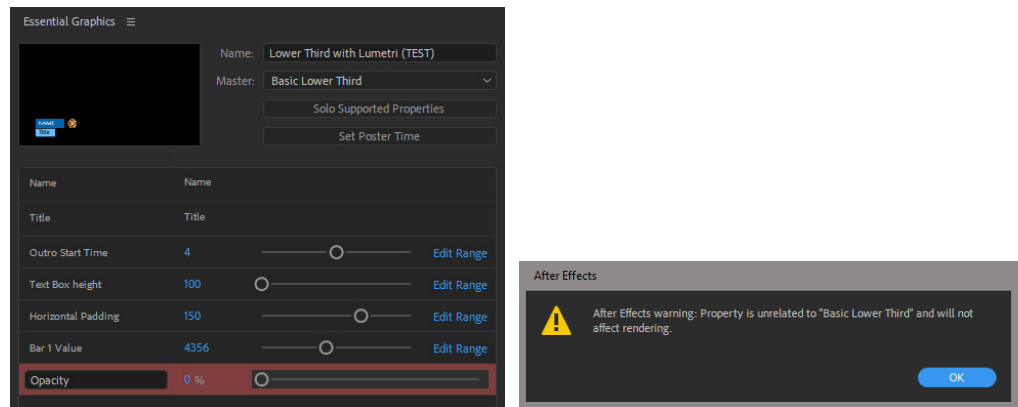
Use nesting to fix problems with unrelated properties

You can add properties to the Essential Graphics panel from the master composition or any child composition in its hierarchy. These are considered *related* properties. Properties from compositions that are *not* in the master composition's hierarchy are considered *unrelated*.

Properties that are linked by expressions to the master composition's hierarchy, but that are not in the master composition's hierarchy themselves, are also considered unrelated.

If you add an unrelated property to the Essential Graphics panel, you will receive a warning. The control will be added, but it will be highlighted in red.

You get a very clear warning when you add a unrelated property to the Essential Graphics panel.



The red highlight will disappear if you nest the unrelated property's composition into the master composition or its hierarchy, which relates the property to the master composition.

Universalizing the MOGRT

After Effects comes in many different languages, and effects and properties have different names in all the languages.

For English AE version: `effect("Duration")("Slider")`
 For German AE version: `effect("Duration")("Schiebereglér")`

If you want your MOGRTs to work in all language versions of After Effects, you must use universal expressions. For example, the universal expression for linking to the slider is:

Universal expression: `effect("Duration")(1)`

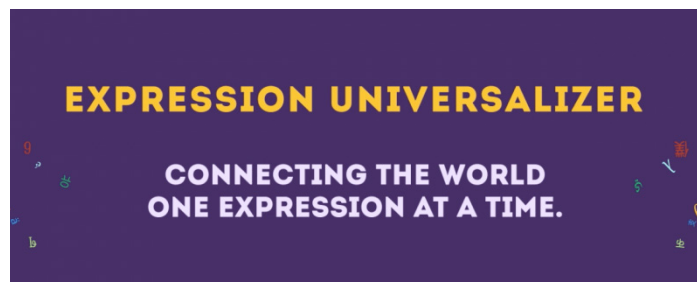
It's impossible to learn how all the different properties should be referred to in a universal expression, but you don't need to. Just press **Alt (Opt)** while pointing the *Expression Pick Whip* to a property, and After Effects will write a universal expression.

Expression Universalizer script

If you don't mind spending a few dollars, you should consider buying the *Expression Universalizer script* created by Lloyd Alvarez. It can universalize all the expressions in the whole project before you export the MOGRT.

<https://aescrpts.com/expressionuniversalizer/>

The Expression Universalizer script is a huge time-saver.



Make sure you save a copy of the project before you universalize it, so you can make changes in a project with normal expressions. This way, you can read in clear English language which properties they link to.

Universalized expressions are harder to read, though, so keep a non-universalized copy of your project.

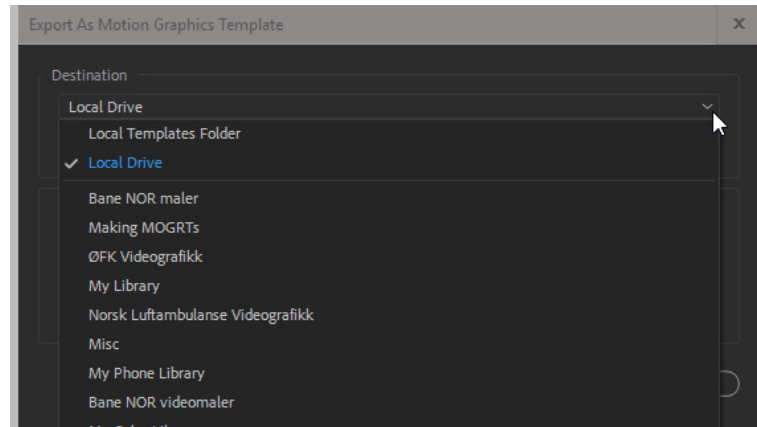
Exporting MOGRTs for use on your own system

When you export a MOGRT, you get three choices: Export to a folder on a local drive, a library, or the local Templates folder. If you're the only one who's going to use the template, you can choose the local Templates folder, and the template will be available in the Essential Graphics panel in Premiere Pro. The actual folder you're exporting to depends on your OS.

```
Windows:  Users\<username>\AppData\Roaming\Adobe\Common\Motion Graphics
Templates\
MacOS:    Users/<username>/Library/Application Support/Adobe/Motion Graphics
Templates/
```

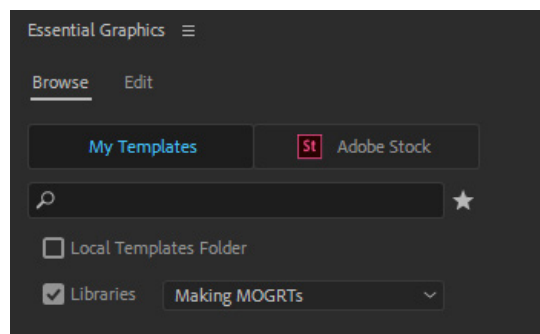
If you find a folder at **Program Files > Adobe > Adobe Premiere Pro CC 2018 > Essential Graphics**, this is a backup of the default templates, in case you should accidentally delete them.

You can export to a local drive, the local Templates folder, or to a library.



Exporting to the local Templates folder works okay, but in my opinion, it's better to export to a library. Libraries are synced to Adobe's server via Creative Cloud and will be available to you wherever you are on the planet on a system with Premiere Pro installed. If you export to a library, you can choose which library to show MOGRTs from in the Essential Graphics panel. No such luck for the local Templates folder yet.

You have more search and filtering control in libraries than in the local Templates folder.



Unless your IT department is paranoid about Internet security and considers everything that's "cloud" to be dangerous, there are no downsides to using libraries.

Sending MOGRT files to customers and colleagues

When you build MOGRTs for a customer, there are two ways you can share them. The most labor-intensive is to save to a local file and send that file to everyone who needs the MOGRT. Then they must install the MOGRT manually on their system in Premiere Pro by clicking the *Install Motion Graphics Template* button in the Essential Graphics panel, or via **Graphics > Install Motion Graphics Template**.

If you've ever tried to get a group of people to do something, you know that there are always a few who will forget to do it, ignore the request, or fail to do it for other reasons. And if you send them an updated version of the file and ask them to delete the old one and install the new one, how many percent will actually do it? Again, libraries are a better choice.

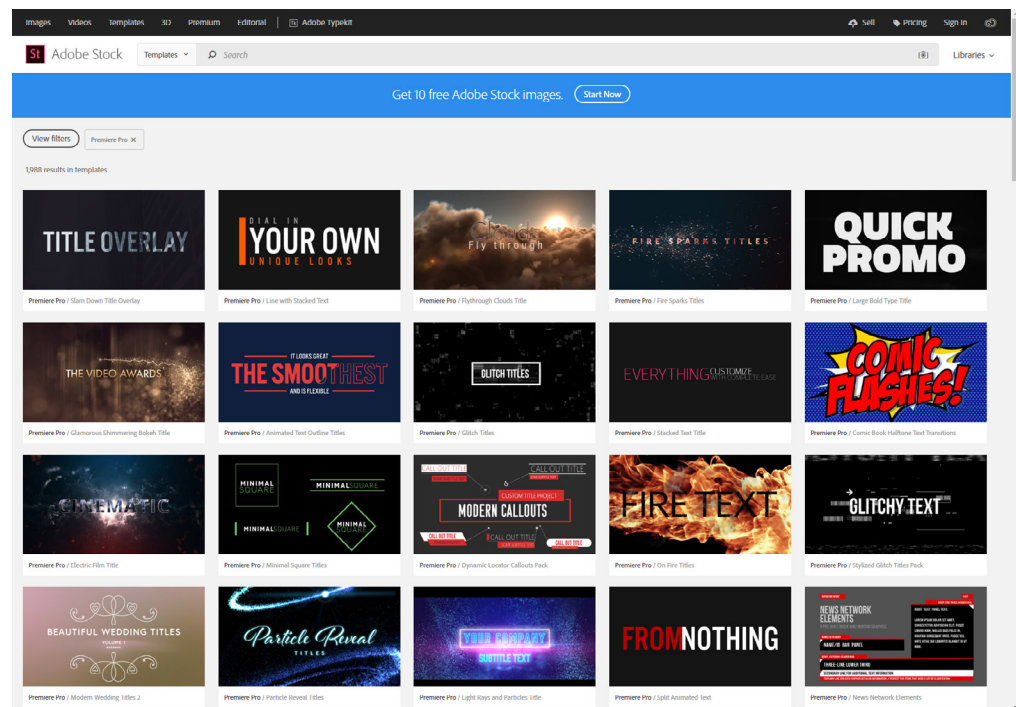
Using libraries, you can auto-distribute MOGRTs to customers and colleagues. If you get write access to their library or, preferably, a test library, you can push the templates to everyone who subscribes to that library in one go, directly from After Effects. And if you delete the old version yourself from the Libraries panel in Premiere Pro, there's absolutely no chance that someone can use the old version.

If the client's IT department will not allow the use of cloud services, your only choice is to send them .mogrt files and hope that they install them.

Selling MOGRTs

If you plan to sell a lot of MOGRTs, you should make sure that your MOGRTs are easy to use. All the guidelines in the "Making MOGRTs user-friendly" chapter apply. It's also very common to make extensive use of comments. We use them as headings as well as for explanations of how a slider works and so on.

**Sell your MOGRTs on
Adobe Stock.**



If you want to sell MOGRTs, the most obvious marketplace is Adobe Stock. The advantage of Adobe Stock is that every user of Premiere Pro can browse, search for, see a video preview, and license templates directly from inside their Essential Graphics panel.

There are, of course, other web pages that sell MOGRTs.

Adobe Stock guidelines

You'll find the Adobe Stock MOGRT requirements and guidelines here:

<https://spark.adobe.com/page/4vDyM6q1IGFXf/>

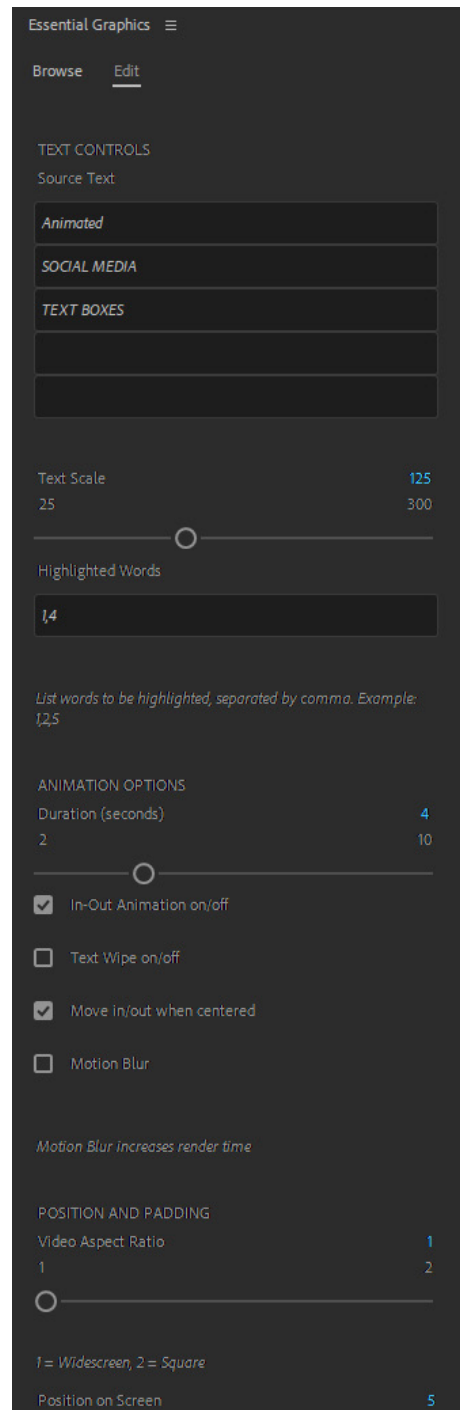
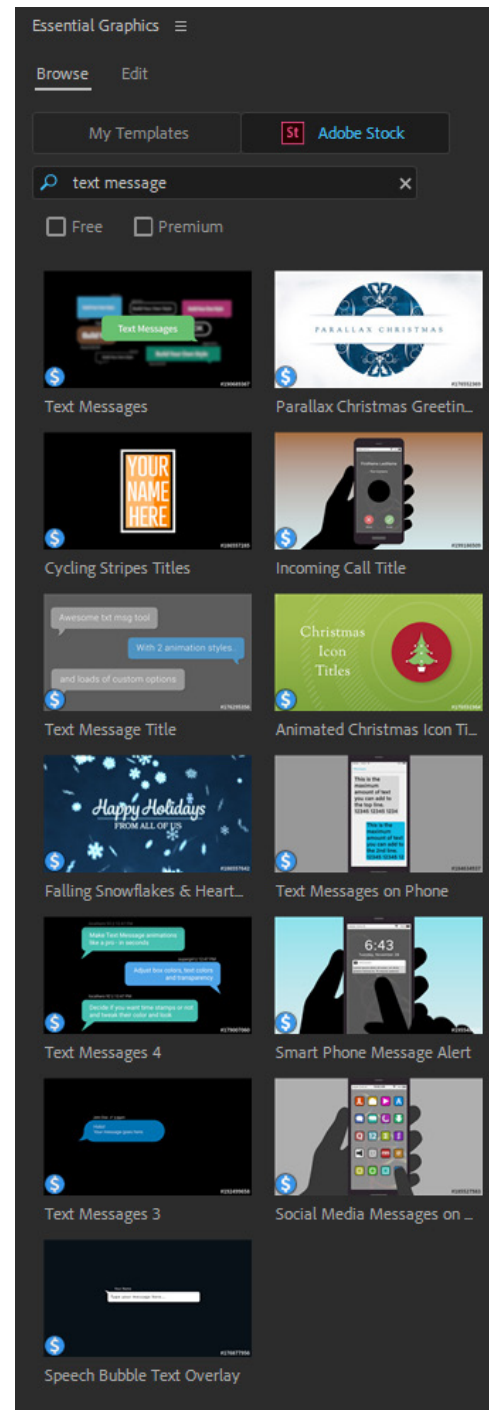
There's also a "best practices" web page here:

<https://spark.adobe.com/page/6sORsTLKTQVzA/>

For more information, reach out to the Adobe Stock staff at motion@adobe.com.

Use comments to explain how sliders and other controls work.

Premiere Pro users can browse Adobe Stock MOGRTs in the Essential Graphics panel.



For now, Adobe Stock requires that MOGRTs be made in After Effects version 15.1.2 or later.

Please note that at the time of this writing, contributors are still able to submit MOGRTs in 15.1.2 but the people at Adobe Stock prefer MOGRTs to be submitted from After Effects 16.

Making changes in existing MOGRTs

Version 15.1.0 (CC 2018.1.0) of After Effects added the option to open a MOGRT as a project in After Effects, retaining its structure and assets. You can then save this as an After Effects project file or export it as a new MOGRT with added functionality for use in Premiere Pro.

Read about the Premiere Pro side of changing and updating MOGRTs in the chapter called “What happens when you install a MOGRT?”

Open MOGRTs as a project

Only MOGRTs (.mogrt files) created by After Effects can be opened as a project file. Premiere Pro created .mogrt files can be changed in Premiere Pro and saved as new templates.

To open a MOGRT as a project, choose **File > Open Project** and select the .mogrt file. After Effects must extract the project file and assets to a folder, so when prompted, select the folder where you want to save the extracted files and click the *Extract* button.

This saves an .aep file (plus assets, if there are any) in a subfolder in the selected folder, and the project opens in After Effects. Here, you can modify comps or add or change controls in the Essential Graphics panel. It's just a normal After Effects project. Then export a new MOGRT from the Essential Graphics panel. You can choose to replace the original .mogrt file or to save the new one with a new name or in a new location.

If you don't have the .mogrt file, there's still hope if you've used the template in a project. The actual MOGRT clip in the timeline is not a .mogrt file. If you right-click on the clip in the timeline and select *Reveal in Finder/Explorer*, it will take you to a folder with a .aegraphic file. If you double-click it, or open it via **File > Open** in After Effects, you will get the same dialog box as when you open a .mogrt file in After Effects.

Even if you're on an older version of After Effects, you can still open a .mogrt file as a project—it's just more complicated: Rename the .mogrt file to .zip. Extract the contents to a new folder. Rename the .aegraphic file inside the folder to .zip, and extract to a new folder. Now you'll have access to the .aep file, and you can open it in After Effects.

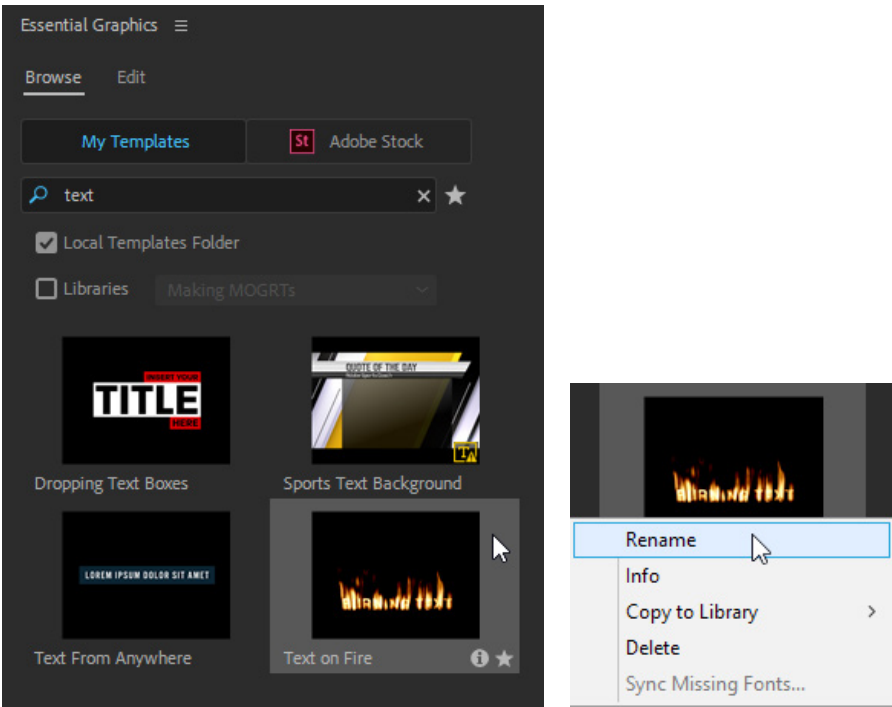
What happens when you install a MOGRT?

When a Premiere Pro user clicks the Install Motion Graphics Template button in the Essential Graphics panel or goes to **Graphics > Install Motion Graphics Template**, the .mogrt file for the chosen template is copied to the local Motion Graphics Templates folder. Where that folder is depends on your OS.

Windows:	Users\<username>\AppData\Roaming\Adobe\Common\Motion Graphics Templates\
MacOS:	Users/<username>/Library/Application Support/Adobe/Motion Graphics Templates/

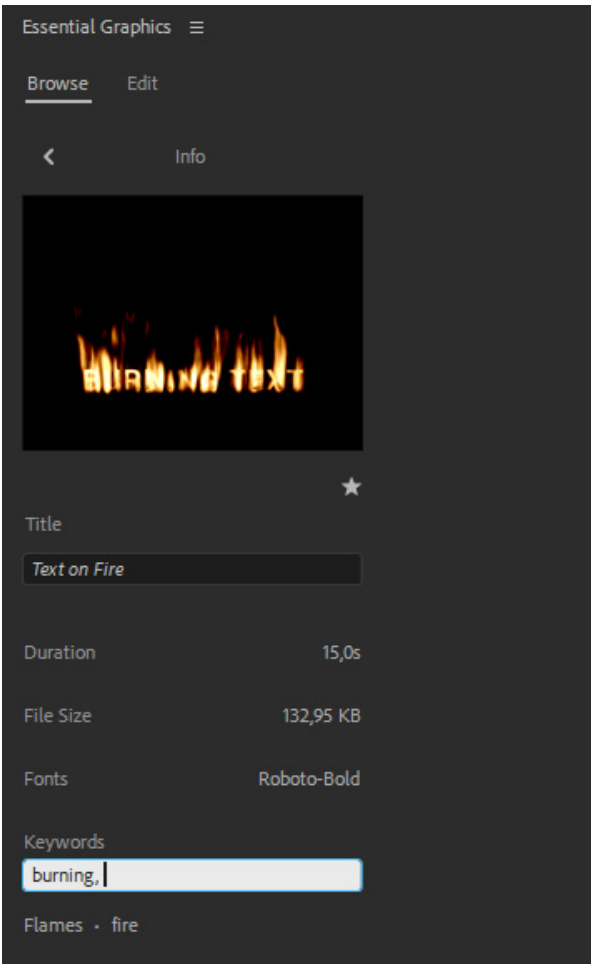
This makes the thumbnail for the MOGRT appear in the Essential Graphics panel when you select the local Templates folder. Here, you can make it a favorite (click the star icon), rename it, get more info, copy it to a library, delete it, or sync missing fonts. Explore the right-click menu to find these options.

The **MOGRT thumbnail** appears in the Essential Graphics panel.



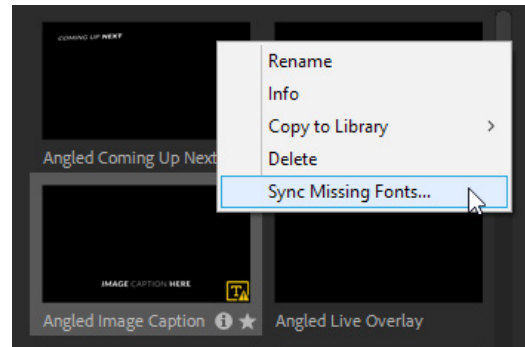
If you click the info icon under the MOGRT thumbnail, you can change the name and add keywords to make it easier to find later.

Change the MOGRT's name, if you want, and add search keywords.



If the template uses a font that you don't have installed, you will see a yellow Adobe Fonts logo in the lower right corner of the thumbnail. You can right-click to sync the missing fonts.

Right-click to sync fonts.



If the user gets the MOGRT from a library instead of installing it manually, it's not copied to the local Motion Graphics Templates folder. Instead, an .aegraphic file and a .png file are stored in the Libraries folder.

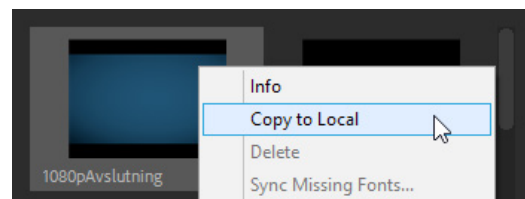
Windows: Users\<username>\AppData\Roaming\Adobe\Creative Cloud Libraries\LIBS\<random characters and numbers>_AdobeID\creative_cloud\dcx\<random characters and numbers>\components

MacOS: Users/<username>/Library/Application Support/Adobe/Creative Cloud Libraries/etc.

I do not recommend that you mess around in the Libraries folder. Instead, add the template to your timeline, right-click the clip, and select *Reveal in Finder/Explorer*. This will take you to a folder with an .aegraphic file. If you double-click it, you will get the same dialog box as when you open a .mogrt file in After Effects.

Since the MOGRT is in your library, you get an option in the right-click menu to copy it to your local Motion Graphics Templates folder.

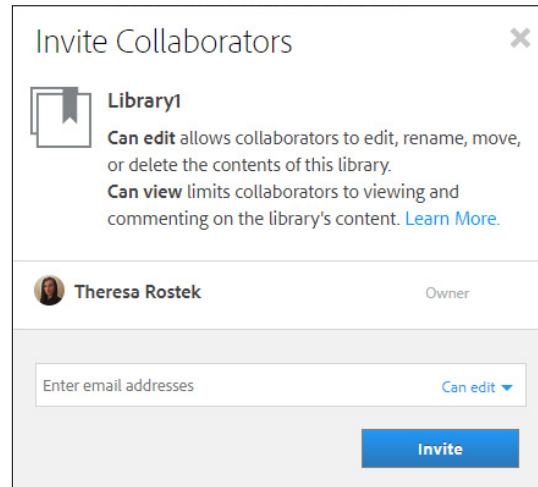
You can copy a template from a library to the local templates folder.



Share your MOGRTs with others using libraries

Right-click the template in the Libraries panel in Premiere Pro and choose *Copy to* or *Move to*, and you can move it to a shared library. If you want to share a library, select it in the Libraries panel, click the panel menu, and choose *Collaborate*. This opens the Invite Collaborators dialog box.

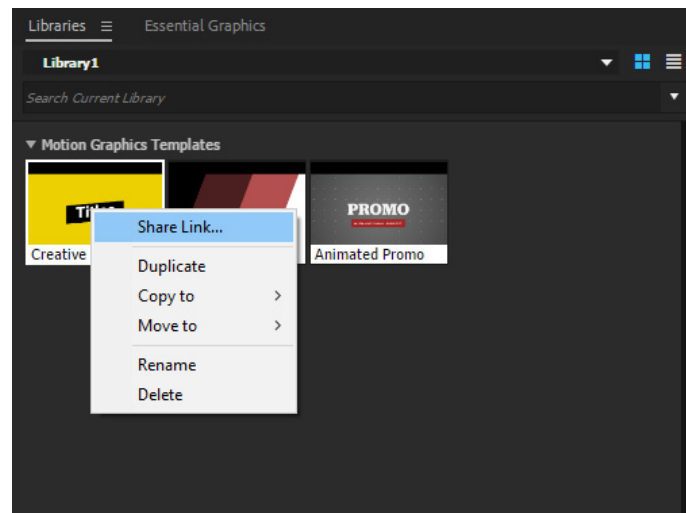
You can invite others to your library by entering their e-mail address (Adobe ID) and clicking Invite.



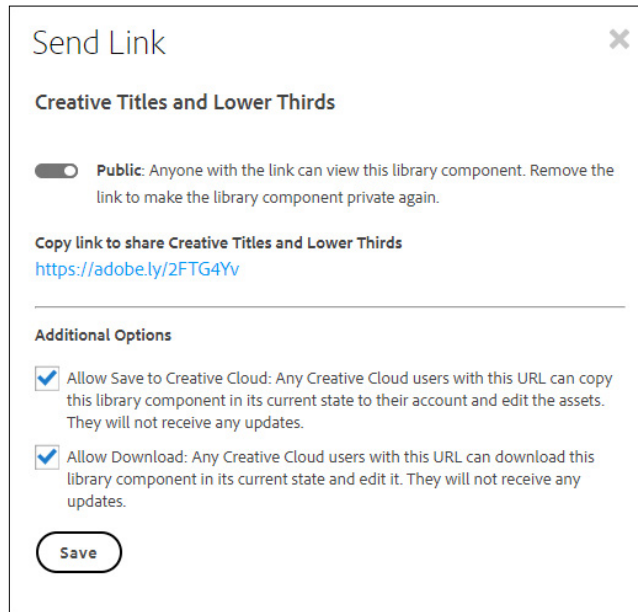
Remember to review their read/write permissions before you send the invite.

You can also choose *Share Link* in the right-click menu. This opens a browser window and takes you to your libraries in Creative Cloud.

Right-click a template in the Libraries panel to share it.



On Creative Cloud
web page, select what the
users should be allowed
to do: Save, download and
so on.



The 'Send Link' dialog box has a title bar with a close button. The main heading is 'Creative Titles and Lower Thirds'. It features a toggle switch for 'Public' which is currently turned on. Below the toggle is a description: 'Public: Anyone with the link can view this library component. Remove the link to make the library component private again.' There is a section for 'Copy link to share Creative Titles and Lower Thirds' with a blue hyperlink 'https://adobe.ly/2FTG4Yv'. A horizontal line separates this from the 'Additional Options' section. This section contains two checked checkboxes: 'Allow Save to Creative Cloud: Any Creative Cloud users with this URL can copy this library component in its current state to their account and edit the assets. They will not receive any updates.' and 'Allow Download: Any Creative Cloud users with this URL can download this library component in its current state and edit it. They will not receive any updates.' At the bottom is a 'Save' button.

Replace the MOGRT with a new version

If you use libraries, it's really simple to update to a new version of a MOGRT:

1. **Delete old version** using the Libraries panel in Premiere Pro.
2. **Export new version** from After Effects

Everyone who subscribes to the library will immediately have the new version.

If you don't use libraries, you need to spread the new MOGRT file to everyone via email or a file sharing service. Then they must uninstall the old one and install the new one. At least 2 out of 10 will not follow your instructions and end up using the old version.

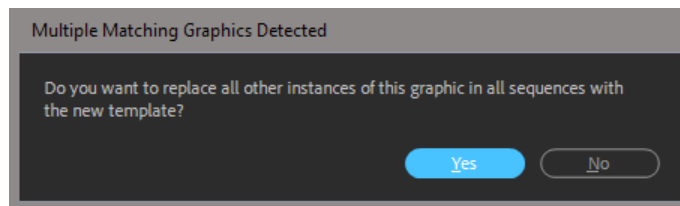
Okay, so that replaces the template in the Essential Graphics panel, so they can't add the old version to their project by accident. Great, but what about the MOGRTs they've already used in a timeline?

Smart Replace templates in Premiere Pro

There's an option in the spring 2018 version or later of Premiere Pro to replace a MOGRT already in use in your sequence with an updated version while preserving the customizations you've already made. In Premiere Pro, press the **Option** key on macOS or the **Alt** key on Windows, and then drag the new version of the template from the Browse tab of the Essential Graphics panel and drop it over the existing clip in the sequence.

If the template is used more than once in the project, Premiere Pro will ask if you want to replace just the selected instance or every instance in all sequences.

Premiere Pro asks if
you want to replace all
instances or just the
selected one.



The dialog box has a title bar that says 'Multiple Matching Graphics Detected'. The main text asks: 'Do you want to replace all other instances of this graphic in all sequences with the new template?'. At the bottom are two buttons: 'Yes' (highlighted in blue) and 'No'.

When you've replaced the clip, Premiere Pro will update the *Edit* tab of the Essential Graphics panel with the new and updated controls. Existing fields are updated when possible, new fields will have their default values, and deleted fields will no longer show.

How to solve font syncing issues

Obviously, fonts cannot be automatically resolved if they are not available on Adobe Fonts. But they can also be in an unresolvable state if there was a communication failure to Adobe Fonts. If you don't have the Creative Cloud application up and you're not logged in, you will run into this issue.

You can also check in After Effects or Premiere Pro under help that you are logged in. That should fix the unresolvable state.

That's it!

This concludes our deep dive into MOGRT creation in After Effects. I hope it has inspired you and enabled you to learn more and develop your skills further. If, by making MOGRTs, you can help people add stunning motion graphics to their films with ease, that's fantastic!

If you have suggestions for stuff we should cover in future updates to this book, like clever code, smart workflows and so on, please send an email to leirpoll@gmail.com.

Elverum, Norway, October 2018

Jarle Leirpoll

Appendix: Online Resources for MOGRT creators

A collection of useful info for MOGRT creators

Tutorials on Essential Graphics panel

How to Set Up an Essential Graphics Panel Step by Step Guide

https://www.youtube.com/watch?v=Vmy_JaIOSH8

How to connect Multiple Animations with Checkbox and Slider Controls

<https://www.youtube.com/watch?v=ab4KPhsmEdY>

After Effects Help

<https://helpx.adobe.com/after-effects/using/creating-motion-graphics-templates.html>

Tutorials on expressions

MotionScript by Dan Ebberts

<http://www.motionscript.com/>

JJ Gifford's Introduction to Expressions

<http://jjgifford.com/expressions/basics/>

After Effects Help

<https://helpx.adobe.com/after-effects/using/expression-language-reference.html>

<https://helpx.adobe.com/after-effects/using/expression-basics.html>

<https://helpx.adobe.com/after-effects/using/expression-examples.html>

After Effects Expressions forums

Creative Cow Expressions forum

https://forums.creativecow.net/adobe_after_effects_expressions

Adobe user-to-user expressions forum

https://forums.adobe.com/community/aftereffects_general_discussion/aftereffects_expressions

After Effects Enhancers forum

<http://www.aenhancers.com/viewforum.php?f=5>

Animoplex Expressions course

https://www.youtube.com/channel/UCbfz_keteqaKbpiRiS95Dqg

Ukramedia tutorials

<https://ukramedia.com/category/tutorials/>

JavaScript resources

After Effects expressions are JavaScript based, and a better knowledge of JavaScript will help you write better code faster and better understand the code you get from web pages. There are many places online where you can learn about JavaScript. Most of them are aimed at web programming, but you can still learn a lot from them. Here are just two examples.

Learn How to Code with Lifehacker.com

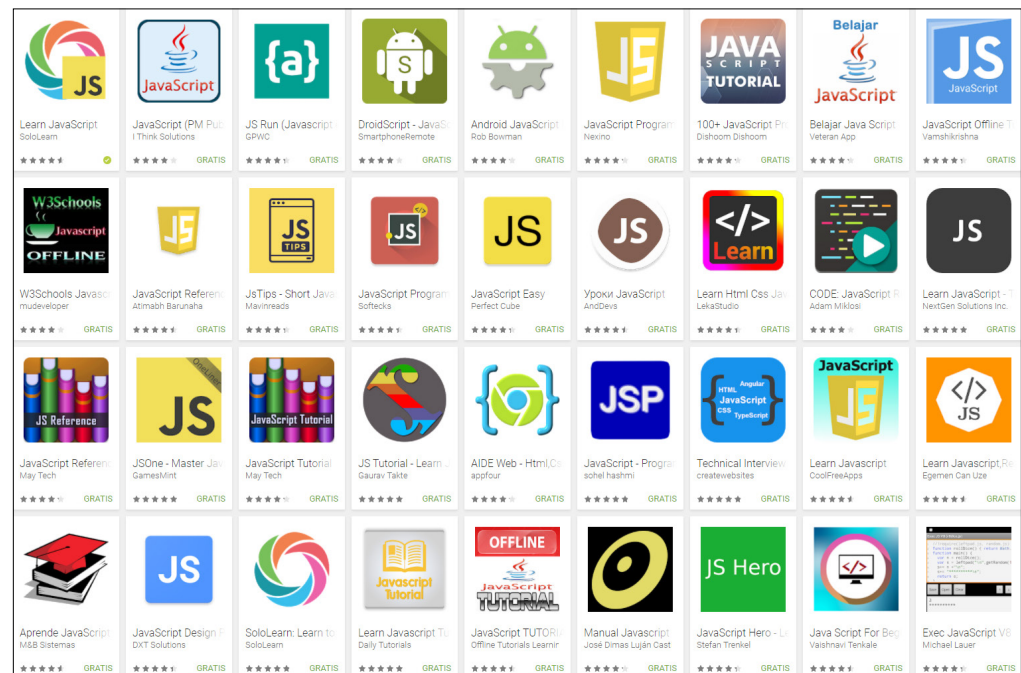
<https://lifehacker.com/5736011/learn-how-to-code-part-i-variables-and-basic-data-types>

Microsoft JavaScript Version Information

<https://docs.microsoft.com/en-us/scripting/javascript/javascript-language-reference>

There are also apps for your smartphone, both free and paid, that will help you learn JavaScript. Search for them on Google Play or the App Store.

Learn JavaScript with
your phone.



Useful scripts and extensions for MOGRT creators

Here's a short list of useful After Effects scripts that can speed up and simplify your work with expressions and MOGRTs. The descriptions are taken from the web pages.

Expressionist

Expressionist improves the expression workflow by giving you basic IDE features such as syntax highlighting, line numbers, and multiple cursors. It allows you to quickly add an expression to multiple properties and gives you inline error handling.

<https://aescrpts.com/expressionist/>

iExpressions

Harness the power of After Effects expressions without writing or reading a single line of code. More than 125 ready to use (and customize) iExpressions—and each of them comes with an easy and intuitive to use interface.

<https://aescrpts.com/iexpressions/>

Expression Universalizer

Converts the expressions in your projects so that they are compatible with After Effects running in any language. Now with batch processing.

<https://aescrpts.com/expressionuniversalizer/>

Find and Replace Script

It helps you to change layers names, effects names, and expressions in your project very quickly. Input what you want to replace and select where you want to change it. That's it! Now you can easily copy your rigged layers or change control layer or do whatever you want! The Find And Replace script can be used as a dockable palette or separate window.

<https://videohive.net/item/find-and-replace-script/10855153>

Expression Toolbox

A centralized location for you to store all your expressions. When you apply an expression from the toolbox, it will rename and/or create any necessary controllers so the expression works properly.

<https://aescrpts.com/expression-toolbox/>

Animation Composer

Animation Composer contains motion presets, precomps, and sounds that you can use in your projects. Everything is super easy to use and to adjust.

<https://misterhorse.tv/animation-composer>

Ease and Wizz

Ease and Wizz is a set of expressions for After Effects that give you more ways to interpolate between values. The obvious use is in motion, but they can be used on animated properties of any kind. They're applied with an After Effects-ish palette that can be docked, so it's very easy to use.

<https://aescrpts.com/ease-and-wizz/>

BOUNCr Script

This dockable script allows you to apply an Overshoot or Bounce Back to selected properties. It also creates a Controller Effect to the selected layers that allows you to quickly adjust the parameters.

<https://ukramedia.com/product/bouncr/>

Under the Influence Script

This script allows you to set the *influence* parameter for *keyframe velocity* without opening the dialog box. You can even apply the values to multiple properties at once.

<https://motionarray.com/tutorials/after-effects-tutorials/how-to/how-to-easily-create-better-animations-in-adobe-after-effects>

EZ Inertia

An easier way to add expressions for realistic bounce.

<https://motionarray.com/tutorials/after-effects-tutorials/how-to/how-to-create-a-realistic-bouncing-effect-in-after-effects>

About the author

Jarle Leirpoll is a filmmaker, motion graphics artist, and trainer based in Norway, and author of [The Cool Stuff in Premiere Pro](#). He is a Premiere Pro Master Trainer and runs [PremierePro.net](#), where he shares free Premiere Pro templates, tutorials, presets, and projects.

Jarle worked at the Norwegian Broadcasting Corporation (NRK) for 14 years before starting his own company in 1996, making corporate movies and documentaries. He's also writing books and articles and does a lot of Premiere Pro and After Effects training all over the world. He loves to make custom MOGRTs for companies, organizations, and individuals.

Jarle has trained people at top broadcasters and media production companies Disney, BBC, NRK, DR, Monster, Norwegian TV2, Swedish TV4, Valve Media Oy, Warner Bros., Discovery, and many, many others. He has even trained Adobe's own employees in Premiere Pro! He is also an official Test Pilot for Premiere Pro, created “handheld camera” effects for the Deadpool movie, and was the tech editor for the 2015 and 2018 versions of [Premiere Pro CC Classroom in a book](#).

Jarle presents at events like Editors Retreat, Adobe Video World, Post Production World, and Adobe MAX and at internal events for broadcasters and production companies.

If you google him, you'll find out about his past. Jarle used to be a magician, was president of the Norwegian Magic Circle for two periods, and has performed twice at the Magic Castle in Hollywood.

Blog: <https://PremierePro.net>

Vimeo: <https://vimeo.com/leirpoll/>

Jarle Leirpoll
(photo by Polina
Sudokova)



Thanks

I want to thank Dennis Radeke, Michelle Gallina Gannon, and Theresa Rostek from Adobe for believing in my idea of writing a book on making MOGRTs. I also want to thank Dacia Saenz, Thomas Spota and Eric Philpott (also from Adobe) for all the help and support—and I also want to thank R. Neil Haugen for proofreading this thing in its early stages, and copy editor Nancy Watson for the final touches.

And this book couldn't have been written if Dan Ebberts wasn't so generous with his advice on expressions all over the Internet.

Want to sell your MOGRTs?



Adobe Stock

Become a contributor to Adobe Stock.

<https://adobe.ly/contributemogrts>

<https://spark.adobe.com/page/6sORsTLKTQVzA/>



Adobe
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com

Adobe, the Adobe logo, Adobe Premiere, After Effects, Creative Cloud, and Adobe Fonts, are either registered trademarks or trademarks of Adobe in the United States and/or other countries. All other trademarks are the property of their respective owners.

© 2018 Adobe. All rights reserved.