

Prysm 2020.4

We are proud to announce the release of Prysm 2020.4 - a great milestone for the product and our team!

New Architecture

When we started developing Prysm in 2014, it was early in the development of Adobe Animate (which was called Adobe Edge at that time). Prysm grew organically with the development of Animate, which meant that we had to add and change stuff all the time, to accommodate Animate platform improvements and changed and improved APIs. At a certain point we reached the stage where we were spending more time keeping up with the changes in Animate and keeping Prysm stable, and less time adding the features from our roadmap.

The user interface of Animate also changed as Adobe improved on the product. As Prysm user interface was built to fit older versions of Animate, it became visibly different and started to look dated, increasing time that designers familiar with newer Animate user interface had to invest to learn it. Our team couldn't take advantage of many modern user experience paradigms, because Prysm was based on several years older Animate plugin framework.

There was one more problem - the developments of the underlying Gameface platform that powers Prysm. Gameface SDK had also evolved a lot and there were many new features that Prysm didn't use and take advantage of.

Improvements in those 3 areas were the driving forces for us to start ground up reengineering and a new architecture for Prysm. There was no easy way to change the architecture of the product incrementally, so we took the opportunity to design and improve a lot behind the scenes in the next-generation of Prysm. We set the following high-level goals:

- Solve issues and implement what was impossible before – 9-slice, gradients, masks, etc.
- Reduce the designer learning curve by unifying the user interface style.
- Do not break existing projects. When exporting existing documents make sure they are visually the same as exported with older versions.
- Improve developer quality of life by providing improved file and content naming, stable identifiers, and improved separation of structure, functionality and style.
- Improved workflow when multiple team members are working with multiple Animate documents. Improve shared folders functionality and allow more granular export (e.g. widgets).
- Improve feature turnaround time for the team and allow faster roadmap implementation.
- Take advantage of new features and platform enhancements available in Gameface.
- Improve experience and compatibility of upgrading existing Scaleform content to Prysm.
- Improved mapping of Animate concepts to HTML and allowing creating more responsive UIs.

Our improvements were split in 2 phases: one - frontend – mainly visual style of the user interface but including migrating to a modern frontend framework (React), and reviewing and improving integration

using newer versions of Adobe Custom Platform Support Development Kit. Second phase was the re-written backend.

With our 2020.3.0 release we have released the new frontend for Prysm. It is much faster, and includes a new visual style for the plugin, based on Adobe Spectrum (<https://spectrum.adobe.com/>) design system.

Release 2020.4

With Prysm release 2020.4 we are also enabling the new backend implementation. As this is a major change we recommend trying the new functionality by making a pilot upgrade and re-exporting everything to test it out before deploying it to the entire team. We will maintain 2020.3 with fixes and patches until the end of 2020 to ensure that teams have enough time to migrate their content to the new output.

We have carefully compared the output of many scenes and documents to make sure that the generated output is correct and created content with previous versions of Prysm will continue to work correctly. Developers should not expect visual changes when upgrading and exporting Animate documents with Prysm 2020.4.

Under the hood we have made many improvements to the structure and the content of the exported assets. If those assets are used with external content, or referenced by external styles or JavaScript, some changes need to use the new identifiers and filenames as listed in “Notable differences” section below.

New features

- Feature REM units are now supported. The "Units" tab and the "Default CSS Length Units" section are extended with a rem option. A "rem base" input in the "Default CSS Length Units" defines the base rem pixel value.
- Feature 9-sliced movie clips are now supported.

Notable differences and breaking changes

1. Output structure and content

Note: we plan to improve the final naming scheme and provide ability to be customized.

- a. Object position and transformations are mapped and exported as styles differently

Prysm 2020.3	Prysm 2020.4
Position of the element is extracted from the Adobe Animate transformation matrix and exported as top and left CSS properties. Resulting transformation matrix is decomposed to rotation and scale or skew and scale. Because this decomposition is not always defined, sometimes transformation matrices can't be decomposed and an error occurs.	Position of the element is extracted from the Adobe Animate transformation matrix and exported as top and left CSS properties, according to the transform-origin (registration point in Adobe Animate). Resulting transformation matrix is directly exported as CSS transformation matrix. Note: the X and Y values of a symbol or element in Adobe Animate do not match the exported values in HTML/CSS.

- b. SVG files are generated in subdirectory named “*svg*” instead of “*img*”
- c. The naming of exported scenes has been changed

Prysm 2020.3	Prysm 2020.4
For each scene in the Animate Document, a file <name of the scene>.html is generated, containing the exported HTML.	For each scene in Animate Document, a file named <name of Animate document>_<name of the scene>.html is generated, containing the exported HTML for the scene.

- d. The naming, structure, and contents of Cascading Style Sheet (CSS) files for each exported scene has changed

Prysm 2020.3	Prysm 2020.4
For each scene in the Animate Document, a single CSS file <name of the scene>.css is generated.	Multiple files are generated: <ul style="list-style-type: none"> ● coherent_prysm_css_clear.css – containing the Prysm HTML baseline styles for all scenes ● <name of Animate Document>_<name of the scene>.css_styles.css – containing the layout and styles for the scene ● If a scene contains animations, <name of Animate Document>_<name of the scene>.css_keyframe_declarations.css – containing the keyframes for the animations ● If a scene contains text objects, <name of Animate Document>_<name of the scene>.css_font_faces.css with font face declarations

- e. More stable object identifiers are introduced, we hope to improve experience with source control and referencing objects by external scripts

Prysm 2020.3	Prysm 2020.4
Each generated element (tag) in the HTML file is styled through multiple css classes named: layoutStyle<random number> , visualStyle<random number> , positionStyle<random number> , animationStyle<random number> , imageStyle<random number> .	Each generated element (tag) in the HTML file is styled through a single css class named prysm_<stable identifier> . The stable identifier is based on the internal order of the object in the Animate document object model.

- f. The naming scheme of generated HTML/CSS animation keyframes has been changed

Prysm 2020.3	Prysm 2020.4
The generated CSS keyframes styles for animations are named animationFrames<random number> .	The generated CSS keyframes styles are split into groups according to their purpose. There are: prysm_<element id>_layoutAnimation , prysm_<element id>_lifetimeAnimation ,

	<i>prysm_<element id>_blendModeAnimation, prysm_<element id>_colorEffectAnimation, prysm_<element id>_movieClipVisibilityAnimation, prysm_<element id>_depthAnimation.</i>
--	--

- g. Additionally, generated CSS keyframes contain a single data point for the start of a tween frame. This leads to a smoother animation. The end frame of a tween still generates two data points for the completeness of the animation and for correct animation control through JavaScript.
- h. Frame labels JavaScript file name has changed. Also frame labels are exported on multiple lines to improve readability. The scene labels are now under the key “scene_labels” so that no clashes with symbol names can happen. In previous versions the key was “global_animations”.

Prysm 2020.3	Prysm 2020.4
The frame labels JS file is named <i>CLAnimations_<name of the scene>_<name of Animate Document>.js</i>	The frame labels JavaScript file is named <i>CLAnimations_<name of Animate Document>_<name of the scene>.js</i> to conform to the rest of the naming convention.

- i. The naming of generated SVG files has been changed

Prysm 2020.3	Prysm 2020.4
For each scene in the Animate document, exported SVG files are named <i><name of the scene>_<random number>.svg.</i>	The generated SVG files are named <i><name of the Animate document>_<stable identifier>.svg.</i> The stable identifier is based on the internal order of the first shape in the SVG file.

- j. In older versions of Prysm symbol shapes, when exported as SVG, have the entire geometry exported as a single “<path>” element. Starting with Prysm 2020.4, all shapes are exported as a “<g>” tag that contain “<path>” tags for the filled regions, and separate “<g>” tags that contain <path> tags for the stroke groups of the shape.
- k. Scaled instances of shapes are exported in Prysm 2020.4 in one SVG file with multiple groups, instead of creating separate SVG files for each instance.
- l. Animate Dynamic Texts are exported as “<p>” HTML elements, instead of “<div>” elements.
- m. The width and height of a movie clip is exported by the AABB of its content in all of its frames. The AABB is correctly calculated irrespective of movie clip transformation and doesn’t grow (e.g. when rotating).
- n. Choosing “position static” and “position relative” in Prysm properties will not export the top and left CSS values of an element to allow relative positioning by external styles.
- o. Text margins (left and right) are included in the AABB of the element.
- p. The coordinate system of each symbol (movie clip) is transformed so all child symbols and elements are at positive coordinates, by calculating the top left bounds of the AABB and using it as the origin of the coordinate system. This is done so that attached events (e.g. onmouseover) and overflows work as expected.
- q. VH and VW units are calculated based on the size of the stage.

- r. Percent units are calculated relative to the size of the parent symbol. Percent units for elements on the stage are calculated relative to the size of the stage. The size of the stage is added to the body tag as pixels.
- s. Older versions of Prysm tried to match rectangular and oval shapes and export them as <div> HTML elements, styled appropriately. There were multiple conditions for this to happen, based on transformation, radiuses, and border styles of the shapes. If conditions were not met, the shapes were exported as images. Rounded borders and bevel borders are properly supported for rectangles.

The usability and predictability of this feature made it inconvenient for developers. Starting with Prysm 2020.4, all geometric shapes are exported as SVG files. Prysm follows Animate rules for grouping and combines multiple shapes in single SVG (shapes in the same frame span with consecutive depths are combined into a single SVG file; shapes with consecutive depth from frame spans from consecutive layers are combined into a single SVG file). In case of animations and tweens, if shapes do not share 100% frame spans on the timeline, they will not be combined, but exported as separate SVG files.

- t. Symbol events for all scenes are exported in a single JavaScript file per Animate document, instead of separate files for each scene. When there are no symbol events no file is generated.
 - u. Experimental widgets publishing is supported, allowing simplified widget or component workflow, without using Animate’s design-time library sharing. It supports scenarios when developers only want to export a specific symbol(s) from the Animate document as independent HTML fragments, that can be easily included in other HTML documents.
 - 1. A new “Publish library item as component” checkbox has been added. Please note that although the setting is available for symbol instances, it is global per library item. All instances of this library item on the scene are affected.
 - 2. Animate scene is exported as usual but empty “div” elements are created in the place of the symbols marked as “publish as component”.
 - 3. For each library item published as a component, a subfolder named as the symbol is created and all a fragment HTML (not containing HTML head and body) is generated. The item is exported as if it is part of the original document, but all identifiers are prepended with the symbol name to avoid naming conflicts.
 - 4. Widgets (components) are exported as self-contained entities, all images, fonts are always copied (overriding “use shared folders” setting).
 - 5. All generated symbol assets are in subfolders relative to the base widget (component) fragment.
2. Bug fixes and feature support that change the generated output
- a. Borders around texts (Animate “Show Border”) are supported in Prysm 2020.4.
 - b. Multiline and Multiline no-wrap text behaviors work properly, previously there was a visual difference.
 - c. Exporting text elements with motion tween is now supported.

- d. Margins on text boxes are now exported as margins correctly, instead of just transforming the position of the textbox.
 - e. Transforming a text element with a transform origin outside the text box now works properly, previously the text was not positioned correctly.
 - f. Layer color effects are now supported.
 - g. Layer blend modes are now supported.
3. Several features that were not properly working in Prysm 2020.3 and earlier have been disabled. We are re-writing part of these with correct and full support in the next versions and they will be re-enabled.
- a. Adobe Animate provides functionality to turn off features which are not supported by a specific document type. Prysm documents were disabling incompatible features, but the API we used was unreliable and not working correctly at all times. With Prysm 2020.4 all of the unsupported and disableable Adobe Animate features are correctly disabled for Prysm documents.
 - b. In case some Animate features still create unsupported content, it is ignored by the publisher and logged to the log file (e.g. vertical align cannot be disabled for text elements, but it is ignored).
 - c. Mask layers
 - d. Gradient fills and strokes
 - e. Filters
 - f. Bitmap fills and strokes
 - g. Support for MacOS

There are many more additional bug fixes and improvements which don't affect generated output. Those are listed in the product changelog.

What is next

Next on our roadmap are re-enabling features – mask layers, gradient fills, filters, support for MacOS, enhancing 9-slice, and improvements to font exporting and enhancing widget workflows and improving design time sharing, runtime sharing, and Prysm shared library workflows. We are also in the process of completely re-working our documentation.