

## Designing a VuMark in Adobe Illustrator

You can use Adobe Illustrator to design your own VuMark template and export the resulting SVG file, which will be used to generate VuMark instances in the Target Manager. This article will show you how to set up Illustrator to use the VuMark Designer scripts and it will walk you through the steps needed to define your VuMark's Code Elements and their corresponding layers.

Be sure to have first read the VuMark Design Guide to understand the role and significance of a VuMark's design principles. See: [VuMark Design Guide](#)

There are three custom scripts for use with Adobe Illustrator that will help you create, verify and export your VuMark design.

### • Setup script

This script creates a new Illustrator file for your VuMark, which we refer to as the VuMark template. It automatically creates the layers that you will place various parts of your VuMark design into.

### • Verify script

This script helps you test your VuMark design by providing a list of design requirements, Pass/Fail results for each requirement and suggestions for resolving any issues that may have been found. You will typically run this script multiple times during the design process as you make changes to correct any failed conditions.

### • Export script

Once your design has been run through the Verify script and it is determined to be ready for export, this script will export your VuMark template as a Scalable Vector Graphics (SVG) file. You will upload this SVG file into the Vuforia Engine Developer Portal Target Manager to create the dataset that will be used within your iOS, Android, Unity, or Windows 10 application.

To help you with an effortless VuMark creation process, we have made a tutorial that covers the design process step-by-step and the design principles in detail.



## Installing and using the Illustrator Scripts ♂

[Download the VuMark Illustrator scripts from the Vuforia Engine Developer Portal](#) and copy it to the folder corresponding to your OS:

OS	Folder
OSX	/Applications/Adobe Illustrator CC folder/Presets/en_US/Scripts
Windows	C:\Program Files\Adobe\Adobe Illustrator CC\Presets\en_US\Scripts

**NOTE:** If you have Adobe Illustrator open, you must quit the application and re-open it to use the newly installed scripts.

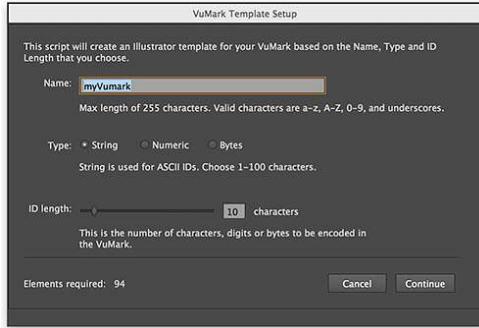
The scripts require that the fonts *Myriad-Pro regular*, *semibold*, and *bold* are installed. Install them by opening one of the Vuforia VuMark Illustrator sample files (*Chateau.ai* or *MortonTuxedos.ai*) that were downloaded together with the scripts. Or visit [Adobe Fonts](#) and activate the font through your Adobe account.

You can now run the scripts from the Illustrator menu: **File > Scripts > VuMark**.

## Setup a VuMark File in Adobe Illustrator ♂

### Using the Setup script

When you run the *Setup* script, a dialog will appear that asks you to enter in a *Name*, *Type* and *ID Length* for your VuMark. The *Type* and *ID Length* fields are used to choose what type of data that you want to encode in your VuMark, and this determines how many **Elements** that your VuMark template must have. Given these inputs, the script will configure your template accordingly.

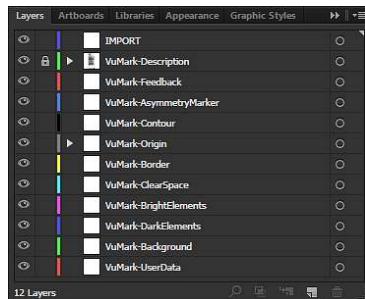


VuMark Template Setup

Type	ID Length
<b>String</b>	Use this if an ID will consist of printable ASCII characters.
<b>Numeric</b>	Use this if an ID will only use numeric digits 0 - 9.
<b>Byte</b>	Use this for data in a byte format.

### Template Layers

After running the **Setup** script, you will have a new Illustrator file containing the layers that are designated to hold the various parts of a VuMark, as well as several additional layers that are also used in the design process. During the design process you may change the order of the layers, but make sure other layers do not hide the **Border** and **Clear Space** graphics. Also ensure **Bright** and **Dark Elements** layers are always visible in your design and not covered by other graphics.



The VuMark native public sample shows an example of a detection highlight graphic displayed over the VuMark. The blue line is a path we added to the *VuMark-UserData* layer. By using this method, you can build a UI that is agnostic to the actual shape of the VuMark.

<b>IMPORT</b>	It is not required that you use this layer, but we have found it helpful to have a workspace layer where you can paste your VuMark design. From this layer, you can begin separating the various parts of your VuMark design by moving them to the corresponding layers.
<b>VuMark-Description</b>	INFORMATIONAL, LOCKED DO NOT ATTEMPT TO MODIFY THIS LAYER. This layer contains the details of your VuMark such as the <i>Name</i> , <i>Type</i> , <i>Length</i> and number of required <b>Elements</b> . The Exporter script uses the information contained in this layer and that is why it is locked by default to prevent accidental deletion.
<b>VuMark-Feedback</b>	This layer is empty at first, but after you run the Verify script, it will be populated with feedback presented on top of your VuMark design to help you verify that it meets the design requirements.
<b>VuMark-AsymmetryMarker</b>	CAN BE EMPTY When creating a rotational symmetric <b>Contour</b> , this layer can be used to add <b>Asymmetry Markers</b> that prevent from accidental wrong ID readout. See section <a href="#">Advanced Topics</a> for more details.
<b>VuMark-Contour</b>	This layer shall only contain one closed path with 4-20 straight edges. Once you have placed your <b>Border</b> and <b>Clear Space</b> parts into their corresponding layers, you must use the <b>Pen</b> tool in Illustrator to draw a closed path along the invisible line that is formed where these two parts touches. This path must be placed in the <i>VuMark-Contour</i> layer.
<b>VuMark-Origin</b>	This layer contains a circle shaped object that represents a point that will be considered the origin on your VuMark design. Augmentations that will appear registered to the VuMark relative to this point. Move them in your design to match required origin. The circle can be scaled.
<b>VuMark-Border</b>	This layer should only contain the object representing your VuMark's <b>Border</b> .
<b>VuMark-ClearSpace</b>	This layer should only contain the object representing your VuMark's <b>Clear Space</b> .
<b>VuMark-BrightElements</b>	This layer will contain the <b>Elements</b> in their <i>Bright</i> state. The number of objects placed in this layer must match the number of required <b>Elements</b> exactly. Make sure <b>Elements</b> are actually brighter than the <i>Dark</i> state.

<b>VuMark-DarkElements</b>	This layer will contain the <b>Elements</b> in their <i>Dark</i> state. The number of objects placed in this layer must match the number of required <b>Elements</b> exactly. Make sure <b>Elements</b> are actually darker than <i>Bright</i> state.
<b>VuMark-Background</b>	CAN BE EMPTY Contains background design. Use rich background if you intend to track VuMark in Vuforia Engine.
<b>VuMark-UserData</b>	CAN BE EMPTY This layer can contain graphical elements for various purposes, e.g. UI overlays, printing marks, metadata, etc. The graphic placed in this layer will be exported as part of the VuMark template SVG file and provided with <code>vuMarkuserdata()</code> ( <code>vuMarkObserverGetUserData()</code> for native) method inside the SDK when the VuMark is loaded.

## Using the Verify script

After you have assembled the parts of your VuMark into their respective layers, you are ready to test your VuMark design.

The Verify script will first check that your VuMark conforms to the design requirements and then provides feedback in the form of a **Pass**, **Fail**, or **Verify** status message in the artboard adjacent to your main artboard in Illustrator.

After running the *Setup* script, you will have two artboards: your main design artboard, and a *Feedback* artboard adjacent to it.

The *Feedback* artboard provides a summary of the VuMark properties you have chosen, as well as a checklist of what will be verified by the script.

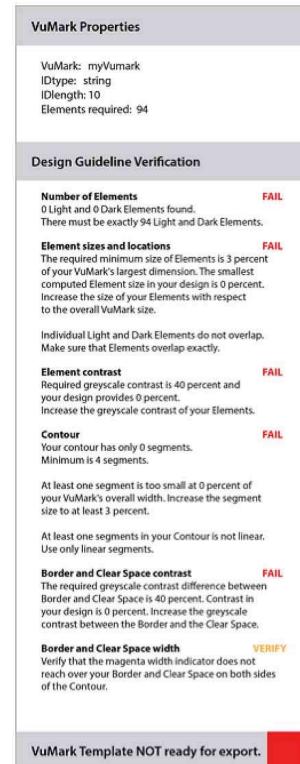
Before you run the *Verify* script, the *Design Guideline Verification* section of the *Feedback* artboard will show a **FAIL** status for each item. It is useful to refer back to these status messages since they provide a reminder of the key design requirements your VuMark must adhere to.

Once you run the *Verify* script, the *Feedback* artboard will update to display the latest verification results.

### The script verifies

- If the number of **Elements** is correct.
  - **Elements'** sizes and locations.
  - **Elements'** contrast.
- **Contour** segments as defined by
  - **Border** and **Clear Space's** contrast.
  - **Border** and **Clear Space's** width.  
(visual verification by user is required).

## The Feedback Artboard in Illustrator



## Filling out the Layers

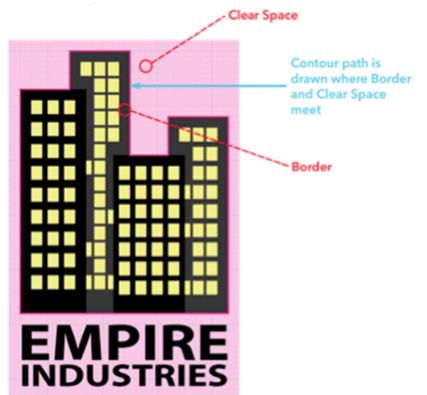
A common design workflow starts a conceptual phase where you experiment with various VuMark design concepts. Once you have chosen a design to move forward with, you can copy and paste it into the *IMPORT* layer of your newly created template. We have found it useful to place the design concept in the *IMPORT* layer and then start moving each part of the VuMark design into their corresponding layers in the Illustrator template.

### Drawing the Contour

As described earlier, the **Contour** is normally not explicitly defined in your VuMark's design up to this point! It is instead generated by the **Border** and **Clean Space** composition.

Place the **Border** and **Clean Space** parts into their corresponding layers. You must use the *Pen Tool* in Illustrator to draw a path along the length of the invisible line that is formed where these two parts touches. Make sure that this path is placed in the *VuMark-Contour* layer.

**TIP:** Trace the **Contour** directly in the *VuMark-Contour* layer using Illustrator's *Pen Tool*, by manually tracing the underlying contrast border between the **Border** and the **Clean Space** anchor-to-anchor.



#### NOTE

- The **Contour** path will not be visible in real instances of your VuMark.
- The **Clear Space** in the above example has been filled in pink to help show its boundaries; it is normally white in such designs.

It is helpful to give the path a 1-2-point stroke with a highly visible highlight color like magenta or cyan so that you can easily see where the **Contour** is drawn.

#### Adding Code / Elements

The empty space between the **Background / Design Area** and the **Clear Space** is designated to place the **Elements**. The **Elements** must be added in two states, *Dark* and *Bright Elements*. There is a layer for each of the two states. The number of **Elements** need to be the same value as the *ID Length* from the *Setup* script. We recommend drawing up the **Element** shape and arrangement in an empty Illustrator page, and thereafter copy them into the VuMark template Illustrator file as *Dark* and *Bright Elements*.

- The contrast between the *Dark* and *Bright Elements* must have a high contrast. As an example, a contrast between two gray colors should have a "K" value difference of at least 30-40%. The "K" value can be found in the *Color* panel when the color space is set to CMYK.

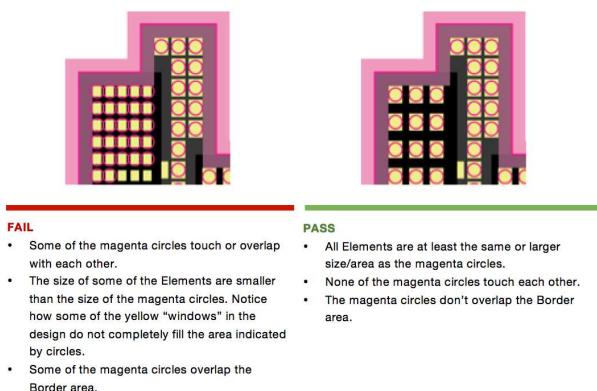
**TIP:** Simple shapes for **Elements** might prove easier to work with when it comes to structuring them within the vacant space.



#### Using the Feedback Layer Information

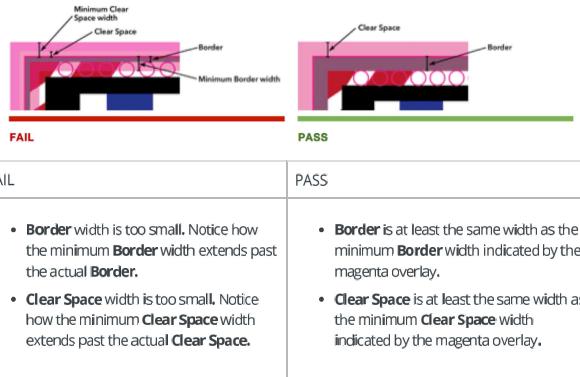
##### Element sizes

The *Verify* script provides a **PASS/FAIL** result for **Element** sizes. It draws magenta circles in the VuMark *FeedbackLayer* on top of every **Element** it identifies. Inspect the **Elements** in your design to make sure that they occupy or fill at least the same size and area as illustrated with the circles. This visualization also helps to identify which **Element** causes the script to fail.



#### Border & Clear Space width

The *Verify* script does not automatically provide a **PASS/FAIL** result for the **Border** and **Clear Space** widths but the *Feedback* layer will display overlays that indicate the required **Border** and **Clear Space** sizes on each side of the **Contour**. In the examples below you can see how a magenta fill is overlaid on each side of the **Contour** to indicate the minimum width required for the **Border** and **Clear Space**.



Both the **Border** and **Clear Space** must be filled paths or compound paths.

**TIP:** If you have designed your **Border** or **Clear Space** with a path that has a simple stroke width, you can use Illustrator's **Object->Expand** or **Object->Expand Appearance** to convert it to a compound path. Any **Appearance Attributes** that are assigned to your shapes, paths, or objects in your VuMark must be removed prior to exporting as an SVG.

- You can use the **Appearance Panel**'s menu option - *Reduce to Basic Appearance* - to remove the **Appearance Attributes**.

The **VuMark-Border** and **VuMark-ClearSpace** layers should only contain a single path or compound path each. The **VuMark-Contour** should contain only a single path that has exactly as many anchor-points as there are corners in your **Contour**. Make sure there is no overlapping anchors in your path. Use the **Document Info -> Object** panel to list the anchor points while the **Contour** path is selected.

## Using the Export Script

### Export Script

When your VuMark design receives a **PASS** status on each of the items in the *Design Guidelines Verification* section and the message "VuMark Template ready for export" appears on the **Feedback** artboard, you may proceed to run the *Export* script.

The *Export* script will run the verification again, create a new file in Illustrator, and move contents from your VuMark template file gradually over. It will automatically trim your design artboard so that it is exactly the size of your VuMark design. Thereafter, the script will prompt you to save your VuMark template as an **SVG** file.

### Troubleshooting

Occasionally, you may find that the artboard trimming does not work correctly, and your artboard still appears larger than your VuMark design. Typically, the reason this happens is that there are one or more objects on your artboard that is registered as being a part of your VuMark design. To fix this, you should inspect each of your layers looking for any objects that are not part of your intended design and remove them. Sometimes there are invisible elements in **Compound Paths** or **Sub-Layers** preventing correct clipping. After cleanup, you can run the *Export* script again and check whether the artboard is trimmed correctly.

## Target Generation in the Vuforia Engine Developer Portal

In the Vuforia Engine Developer Portal create a VuMark database by using the saved SVG file.

Upload the VuMark SVG template to create a VuMark instances of the uploaded template.

For a complete guide to using the Vuforia Target Manager and uploading VuMark Templates and create VuMark instances, please see [How to Work with VuMark Databases](#).

## Advanced Topics

### Rotational Symmetry Asymmetry Markers

We recommend creating designs that are not rotational symmetric as illustrated below:



Figure 8 – Comparison of symmetric and non-symmetric shapes.

### Rotational symmetry examples

When **contours** are rotational symmetric, there is a minimal, theoretical chance that a **Code** contained in a certain specific instance may not be read out correctly. If you want to assure that this never happens you will need to add **Asymmetry Markers** to your design.

**Asymmetry Markers** are little circles manually added to the **VuMark-AsymmetryMarkers** layer, named in order a0000, a0001, a0002,. Their purpose is to mark up one arbitrary, rotational symmetric location in your design with respect to



Figure 10. Order 2 asymmetry marked up in ACME design

the **Contour** rotation, that fulfills following criteria (see Figure 10 and Figure 11 for examples):

- The number of **Asymmetry Markers** must be identical to the symmetry order of your **Contour** edges (see Figure 8 for examples).
- When rotated around the symmetry center, **individual Asymmetry Markers** need to fall onto each-other. Basically, they need to be on rotational symmetric locations (as shown in Figure 9).
- One, and only one out of the multiple **Asymmetry Markers** must have in its center a different color than the other markers. The different fill locks the orientation to the specific **Asymmetry Marker**. The contrast between the one and the other markers should be similar to what is used for *Dark/Bright* elements.



Figure 9. Order 5 Asymmetry Marker Placement Example

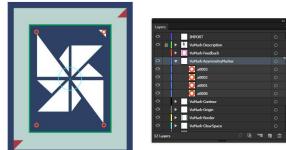


Figure 11. Order 4 asymmetry marked up in windmill design

## Tips & Troubleshooting ☰

### Troubleshooting VuMark creation/detection fails

Due to the variety of ways how a VuMark design can be created there may be still cases when verification and export of the VuMark succeeds, but the VuMark generation upon upload to the Developer Portal fails, or even that a downloaded VuMark dataset is not capable of detecting instances (e.g. the 0 instance displayed in the VuMark Target detail panel).

If such a situation occurs, try to apply one of the recommended fixes from the below listed possible explanations.

- **Overlapping points in contour path causing more points than actual corners in contour.** When the **Contour** is not created carefully using the *Pen Tool* as recommended, more points end up in contour path than your visual design actually has. Please verify using **Document Info -> Objects** that you have as many points in the closed path as your design actually has when counting corners of the **Contour** visually.
- Check the **maximum angle limit** between contour segments is at maximum 150 °.
- **Shape transformation in Contour/Border/Clear Space elements.** *Move/rotation/scale* of an object results in an update to the anchor points of a path or compound path. Shapes, however, can store above changes as either an update to their width/height and position coordinates OR in a separate transformation. This can cause a failed VuMark SVG Template file. Use the **Object -> Shape -> Expand Shape** command to flatten them as paths.
- **Do not use appearance attributes for Elements/Contour/ClearSpace/Border!** When listed objects have *Appearance Attributes*, their export to SVG is not always properly rendered and verification may provide incorrect results. You can easily identify the issue, as *Appearance Attributes* are displayed with a filled dot in *Layers* panel:



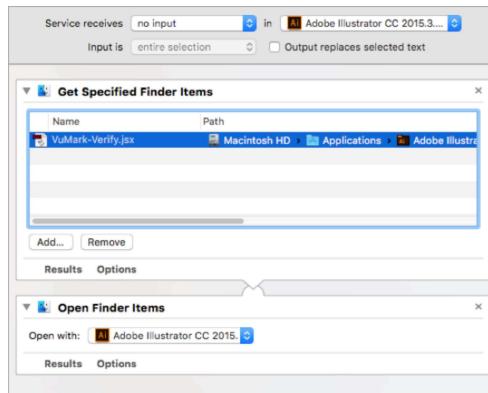
Appearance attribute is used in VuMark-Border layer

To fix this issue select the object and use **Object -> Expand Appearance**.

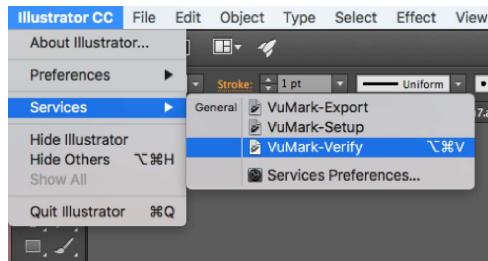
## Calling Scripts with Keyboard-Shortcuts on MacOS ☰

To save time during iterative checks using the VuMark Verify script, you might want to setup a keyboard shortcut to run the script. Shortcuts are OS platform-specific and techniques will vary. As a convenience, instructions for one way to do so on the Mac version of Illustrator are provided below:

1. **Launch** Mac Automator.
2. Select **Service** and click the **Choose** button to accept
3. **Set** the following values: *Service receives no input in Adobe Illustrator*
4. **Add a Get Specified Finder Items** and click **Add** to select *Vumark-Verify.jsx* script located in the *Presets/en\_US/Scripts/* folder of the Illustrator application.
5. **Add an Open Finder Items** and set it to the Illustrator application
6. **Save** your new workflow service with name such as *VuMark-Verify*



7. Open **Keyboard** in **Settings** and select **Shortcuts** tab
8. With **Services** selected, scroll till you find VuMark-Verify and then click **Add Shortcut** and choose an available key combination.
9. **Restart** Illustrator and check **Services** for your script with assigned hotkey.



## Learn More

---

[VuMark Design Guide](#)

[How to Work with VuMark Databases](#)

[Working with VuMarks in Unity](#)

[Can this page be better?](#)

[Share your feedback via our issue tracker](#)

## Vuforia Engine

[Home](#)  
[Support Center](#)  
[Downloads](#)  
[Library](#)

## PTC, Inc.

[Products](#)  
[Latest News](#)  
[Careers](#)  
[Contact Us](#)

## Policies

[Developer Agreement](#)  
[Terms of Use](#)  
[Privacy Policy](#)  
[PTC Trust Center](#)