Release Notes for Mari 3.0v4

Release Date

19 July 2016

NOTE: The Mari documentation available in PDF format will soon be deprecated in favor of the help in HTML format. This documentation is shipped with the product and can accessed without Internet connection.

System Requirements

NOTE: Mari increases its level of performance with newer, more advanced hardware configurations. However, Mari is tested and capable of operating on many older, prior-generation systems. For this reason we are listing below-minimum requirements, which are recommended, and on which tests have been performed. Your particular needs may vary from that of other users.

Officially Supported Operating Systems

- Mac OS X 10.9.5 (Mavericks) or higher
- Windows 7 64-bit or higher
- Linux 64-bit operating system (CentOS/RHEL 5.4)

Minimum Hardware Requirements

- Quad-core processor
- 10+GB disk space available for caching and temporary files
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers

- 1GB of graphics memory
- OpenGL 3.2* or higher

*Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

Recommended System Requirements

- 2.5+Ghz Quad-core processor
- 250+GB disk space available for caching and temporary files. SSD is preferable.
- 16GB RAM with additional virtual memory*
- Display with 1920 x 1080 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers
- 2+GB of graphics memory
- OpenGL 4.4 or higher support

*The use of virtual memory improves stability and helps prevent data loss on large projects.

'Recommended' does not guarantee that it meets your particular needs.

Tested Workstation Hardware

The configurations listed below are those that The Foundry have tested with Mari 3.0v3. Due to the constantly changing nature and wide variety of computer hardware available in the market, The Foundry is unable to officially certify hardware. The list below can be used as a recommendation and does not guarantee that it meets your particular needs.

Please download and install the latest graphics driver from the NVIDIA or AMD websites, and ensure that you are using 8.982.1 drivers or higher for AMD cards.

If you encounter any issues, please contact support@thefoundry.co.uk.

Tested Workstation Hardware	
Apple	• MacPro4,1 • MacPro5,1 • iMac13,2 • MacBookPro10,1
HP	• HP Z600 • HP Z620 • HP Z800

Tested GPU Hardware	
AMD Prosumer Graphics Cards	• AMD Radeon HD - D500 • AMD Radeon R7 - 260x
AMD Enterprise Graphics Cards	• AMD FirePro W7000
NVIDIA Prosumer Graphics Cards	• NVIDIA Geforce GTX 680 • NVIDIA Geforce GTX Titan • NVIDIA Geforce GT 650M
NVIDIA Enterprise Graphics Cards	• NVIDIA Quadro 4000 • NVIDIA Quadro K5000 • NVIDIA Quadro M6000

New Features

There are no new features for this release.

Feature Enhancements

- BUG ID 208461 - The GL version has been updated to 4.3 (hardware dependent).
- BUG ID 218253 - A **Virtual Texture Depth** preference has been added to control the number of 2D texture layers used in the virtual texture Mari uses for display. Increasing this may resolve issues with flickering textures. This value is dependent on the graphics card capabilities. This preference is disabled for Mac OS X platforms.

Bug Fixes

- BUG ID 127948 - Smart selection didn't work correctly in Edge Angles and Orientation modes.
- BUG ID 173066 - Copying and pasting layers across objects rendered the wrong default colors as well as shallow copies of the layers.
- BUG ID 176239 - The layer hash method did not take into consideration any non-layer nodes or the Node Graph.
- BUG ID 196364 - The **Objects** palette sort option didn't work.

  It now sorts the root object names alphabetically, leaving the children objects names in the same order they were imported in Mari.
- BUG ID 201092 - Layers lost their cache when upgrading archives across versions.
- BUG ID 201143 - Mari was significantly slower when using OCIO roles rather than using the corresponding colorspaces.

- BUG ID 201151 - Mari wasn't clamping the results to the zero to one range from the colorspace conversion on the images used in the **Paint through** tool. This meant the texture image in the paint buffer didn't match the paint that was being sent through it.
- BUG ID 201428 - Disabling a mask caused a rendering artifact and detached the channel from the current Viewer.
- BUG ID 207548 - Mari introduced very noticeable stepping artifacts when painting byte (8-bit) textures.
- BUG ID 207662 - Adding multiple output nodes in a Group node did not show an error dialog and stop the action.
- BUG ID 208567 - Zoom to selected faces (**A** key by default) didn't work.
- BUG ID 215278 - Copying Paint nodes across objects resulted in empty Paint nodes.
- BUG ID 216423 - The Third Party Library Versions appendix did not include the GCC, OpenSubDiv, and Ptex libraries.
- BUG ID 216761 - Mari 3.0 was significantly slower than earlier versions, such as 2.6. This was particularly the case when working with large and complicated projects.
- BUG ID 217000 - Reverting snapshots caused Mari to crash.
- BUG ID 218222 - Mac OS X only: Mari sometimes failed to export channels with flattened layers
- BUG ID 219347 - The **Layout.mws** could become corrupted, causing a crash on startup when migrating from Mari 2.6.
- BUG ID 219482 - Mac OS X only: Very noticeable lighting artifacts appeared in the canvas render when using environment lights with an AMD GPU.

Known Issues and Workarounds

Mari Tools

- BUG ID 13640 - The **Blur** tool can be slow to use on the initial stroke.

  Wait for Mari to process the blur before applying a second stroke.
- BUG ID 13394 - Using the **Select Items** tool with the **Facing** set to **Front** to select and hide a portion of faces causes some of the faces within the selection to remain visible when zoomed in.

  To catch all selected faces, either:
  - select **Facing > Through** instead of **Front**, or
  - zoom in closer to the object.
- BUG ID 118852 - The **Erase** mode in the **Slerp** tool does not work.

Shaders

- BUG ID 34729 - Mari displays a rendering error on the canvas when it is unable to create a shader. More information has been included to help you determine the cause of the error. Some solutions might be to hide groups and layers, or to cache parts of your layer stack until a shader can be created.

- BUG ID 34679 - On extremely large projects, issues can arise with shader limits, and reaching the maximum allowed texture slots available. To avoid reaching these shader limits on large projects, try the following workarounds:
  - hide groups and layers, or
  - cache groups and layers.

Layers

- BUG ID 34690 - Flattening or caching layers or channels on complex projects may cause Windows to reset the graphics driver due to the long processing time. To work around this issue, you can try to flatten or cache fewer layers at a time, or reduce the value of the **Max Render Size For Baking** setting. This setting can be found under **Preferences** > **GPU** > **Baking and Projection**.

  Reducing this size breaks the flattening or caching operation up into smaller pieces, which individually take less time to calculate, and thereby avoids a Windows graphics driver reset.
- BUG ID 26460 - Painting a mask in a Mask Layer Group sometimes results in unexpected paint results. To prevent this from happening, either:
  - Use a white “color” layer at the bottom of your mask stack. Any layer used over this initial “color” layer should then be fine, or
  - If you want to create a mask in a Mask Layer Group, simply add another layer on your Mask Layer Group instead, and paint white into it to create a mask.

Importing and Exporting

- BUG ID 51655 - Mari becomes unresponsive when attempting to export Ptex textures.
- BUG ID 50886 - Session Scripts: Imported shaders don't have channels assigned.
- BUG ID 49634 - Session Scripts: Not all features new to Mari 3.0 are supported by session scripts.
- BUG ID 49131 - High polygon **.obj** files, exported using the OBJ Exporter plug-in, cannot be read back in to Mari.
- BUG ID 29386 - When using the **Export for Maya** script, Maya’s viewport may incorrectly show some patches as transparent. This can be resolved by selecting **High Quality Rendering** or **Viewport 2.0** from the Renderer menu within Maya.
- BUG ID 16324 - Windows only: you cannot currently import an image into a channel using a relative file path. To work around this, use an absolute path when importing images.
- BUG ID 14985 - There may be a slight pause after importing textures when creating new projects, while Mari saves the project.

Nuke<>Mari Bridge

- BUG ID 23010 - Nuke<>Mari Bridge: If Mari crashes when receiving incoming components from Nuke when the **Virtual Texture Type** is set to **Float**, lower the **Virtual Texture Size** to a value below 8192x8192.
- BUG ID 19780 - Nuke<>Mari Bridge: A projector created in **Ortho** view in Mari does not re-project correctly in Nuke.

Ptex

- BUG ID 17626 - It can take a long time to import very large or very high polygon count Ptex models.

  The workaround is to assign a small uniform face size (1x1 or 2x2) on import, and then increase the resolution of the relevant bits of the model as necessary after loading.

- BUG ID 17618 - Ptex does not bake properly if the resolution of the face is too small.

  The workaround is to increase the resolution of the selected faces you are having problems with.

Graphics Cards

- BUG ID 217864 - Mac OS X only: AMD D500 and D700 graphics cards, found on modern Mac Pro's, are physically limited to 16 Texture Mapping Units, which can block operations such as multiple procedural layers.
- BUG ID 207913 - Mari may crash, or operate with lower than expected performance, when using the latest graphics drivers with some AMD FirePro cards.

  There have been reports that Plays.tv and Raptr, which come bundled with the AMD FirePro drivers, can cause instability in Mari. If you are experiencing instability, please try uninstalling these applications.

- BUG ID 18457 - Using NVIDIA graphics cards from the **Fermi** series with drivers older than version 270 results in various rendering issues when the **Virtual Texture Type** is set to **Half** or **Float**.

  To resolve this, please download and install the latest graphics driver for your card from the NVIDIA website.

- BUG ID 12567 - Enabling **Sync to VBlank** in NVIDIA settings can drastically reduce Mari’s performance. If you experience very slow interaction, even with low-polygon models, on one of the Tested Workstation Hardware, navigate to:
  - Linux: **NVIDIA X Server Settings > X Screen 0 > OpenGL Settings** and turn off **Sync to VBlank**.
  - Windows: **NVIDIA Control Panel > 3D Settings > Manage 3DSettings > Vertical Sync > Force off**

  Then, restart Mari.

Node Graph

- BUG ID 168753 - AIStandard nodes created in Mari3.0v1 are not compatible with Mari 3.0v2, or later.

  As a workaround, remove the old AIStandard nodes and recreate them in Mari 3.0v2, or later.

- BUG ID 51462 - Creating a shader and attempting to view it in the **Node Graph** palette gives the impression that the DiffuseColor input edge is missing from the shader node. The input is present, but is incorrectly hidden. This is related to bug 51263 below.
- BUG ID 51452 - When importing a gizmo, the nodes in the gizmo do not retain their organization if they were created using the item name in the menu.
- BUG ID 51263 - It is not currently possible to attach the Viewer node to standard Mari shaders, because they are hidden.

  As a workaround, you can click the **View the current channel** button above the Node Graph.

- BUG ID 51247 - Channel transfer doesn't transfer Graph Layers as expected.

- BUG ID 51082 - The Ambient Occlusion node's properties don't include a **generate AO** option.
- BUG ID 48790 - **Autoplace** does not respect Backdrop nodes.

Miscellaneous

- BUG ID 221920 - When copying channels between objects, a warning message can pop up stating the requested channel cannot be created. However, clicking **OK** to this message usually results in the channel copying successfully, regardless.
- BUG ID 208632 - Mac OS X only: Mari now allocates an additional Texture Mapping Unit to improve performance.
- BUG ID 194832 - Bake times were significantly increased when baking into channels containing non-linear data.
- BUG ID 177430 - Mari occasionally crashes on startup during the initialization of Python.
- BUG ID 167883/51934 - When a project that contains a Tiled procedural is upgraded from 2.6 to 3.0, the frame rate drops drastically.
- BUG ID 129292/51771 - Removing, changing, or hiding subdivided objects takes a long time.

  To avoid this, replace the image in the Tiled procedural once the project has been upgraded.
- BUG ID 99115/46223 - The **Sponge Desaturate** mode does not work through the full dynamic range as it uses HSL for desaturation. HSL cannot be used with HDR because HSL works well only in LDR values.
- BUG ID 51370 - Heavier projects are initially slower to render when colorspace is enabled.
- BUG ID 51322 - Modo Render: The preview occasionally fails to update fully.
- BUG ID 51273 - Mac OS X 10.10 and above: Rendering is occasionally incorrect when using multiple lights.
- BUG ID 51199 - The AiStandard, RedshiftArchitectural, and VRayMtl shaders are not connected to the Current Channel automatically.
- BUG ID 51185 - PythonAPI: Mari's Paint node does not appear in **typeList()**.

  To add a Paint node Pythonically, call:

  ```
  ng = mari.geo.current().nodeGraph()
  ng.createPaintNode(width, height, bitDepth)
  ```
- BUG ID 51084 - Animated objects can take a long time to subdivide.
- BUG ID 50830 - The **Objects** palette lock doesn't prevent objects from being transformed.
- BUG ID 50548 - Modo Render: Only camera moves are respected by live update.
- BUG ID 50520 - Although faces with degenerate UVs can be loaded into Mari, they can cause issues in some cases. They do not occupy any space in UV, so it's impossible to properly paint on such faces. There is also the risk that some shaders may show undesirable lighting effects on faces with degenerate UVs.
- BUG ID 50149 - Texture transfer does not take object transformation into account.
- BUG ID 50898 - Existing subdivision calculations are lost when recalculating, even if recalculation fails.
- BUG ID 49557 - Shadows and colors can appear incorrect after subdividing geometry.
- BUG ID 47180 - Mac OS X 10.10 and above: Resizing palettes is restricted to a certain width or height per drag action.
- BUG ID 46600 - Ambient Occlusion must be updated after any OpenSubdiv calculation.

- BUG ID 43020 - Mac OS X with retina screens only: When a hidden menu is unfurled over the canvas, the canvas zooms in dramatically.
- BUG ID 41573 - Windows 8 only: The Windows key (Meta key) does not disengage when used in conjunction with a Wacom pen.

  To successfully disengage, you need to press the Windows key again over the desktop, which rectifies the problem.
- BUG ID 37140 - Mac OS X only: By default, when you first install Mac OS 10.8 or higher, the security preferences are set so that any applications not downloaded from the Apple App Store can't be installed.
  To ensure that Mari installs correctly, navigate to **System Preferences** > **Security & Privacy** on your Mac and select **Anywhere** for the **Allow applications download from** field.
- BUG ID 33293 - Linux: Launching Mari with the language set to one without certain character symbols resulted in Mari failing with an error that the specified transform could not be loaded.

  To work around this, set the locale (language) to English.
- BUG ID 31946 - Sometimes paint is not baked because of memory management issues on the graphics card. This can be due to issues such as a high resolution paint buffer, a high bit-depth paint buffer, large virtual texture size, or even a large scale value on the paint buffer transform. These issues can usually be identified by **glError: 0x505 out of memory** messages in the log.

  Try reducing any or all of these values to prevent it happening. Graphics drivers are continually improving, so it's also worth checking whether upgrading your drivers resolves the problem.
- BUG ID 20510 - If you find that the startup time for Mari is longer than usual, please check that the LIC files in your RLM licensing data folder do not refer to obsolete server ports. If they do, place them in another directory and restart Mari.
- BUG ID 20021 - Textures in the canvas intermittently switch between lower and higher resolutions.

  This issue is more likely to occur if your virtual texture resolution is low, and you're working on a complex model with displacement. Possible workarounds include increasing your virtual texture size, reducing the number of channels Mari has to access at once (for example, by reducing the number of channels required for the current shader), to reduce the patch resolution of patches in the channels used in the shader, or to use a smaller canvas window or monitor.
- BUG ID 14201 - Linux only: Mari becomes unresponsive after the system is woken from sleep.
- BUG ID 13700 - Adjusting the **Camera > Perspective** settings for a **Projector** is not reflected on the canvas until the **Projector** is made Current.
- BUG ID 13571 - Launching a new version of Mari for the first time, when a config file exists from a previous version, sometimes results in an object not appearing in the **Ortho** view.

  To solve this, close Mari, delete the following config file and relaunch Mari:

  - Linux: **~/.config/TheFoundry/Mari.conf**
  - Windows: **C:/Users/<login>/.mari/TheFoundry/Mari.ini**
- BUG ID 13294 - Windows: Mari sometimes crashes when trying to load data on large projects due to the program exhausting all window manager objects.

  To reconfigure the user object limit:

  - Open regedit and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\Windows**, and

- Edit **USERProcessHandleQuota** to a larger number.

If this number gets too large, you may also have to modify **GDIProcessHandleQuota.**

- BUG ID 12102 - Current brush settings do not get saved as part of the project. Instead, Mari reverts to the default settings when you close and relaunch it.
- BUG ID 11874 - Mari doesn't recognize 3-digit padded **.obj** sequences as animation.

Developer Notes

These are the changes relevant to developers.

New Features

There are no new features for this release.

Feature Enhancements

There are no feature enhancements for this release.

Bug Fixes

There are no bug fixes for this release.

Release Notes for Mari 3.0v3

Release Date

13 April 2016

System Requirements

NOTE: Mari increases its level of performance with newer, more advanced hardware configurations. However, Mari is tested and capable of operating on many older, prior-generation systems. For this reason we are listing below-minimum requirements, which are recommended, and on which tests have been performed. Your particular needs may vary from that of other users.

Officially Supported Operating Systems

- Mac OS X 10.9.5 (Mavericks) or higher
- Windows 7 64-bit or higher
- Linux 64-bit operating system (CentOS/RHEL 5.4)

Minimum Hardware Requirements

- Quad-core processor
- 10+GB disk space available for caching and temporary files
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers
- 1GB of graphics memory
- OpenGL 3.2* or higher

*Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

Recommended System Requirements

- 2.5+Ghz Quad-core processor
- 250+GB disk space available for caching and temporary files. SSD is preferable.
- 16GB RAM with additional virtual memory*
- Display with 1920 x 1080 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers
- 2+GB of graphics memory
- OpenGL 4.4 or higher support

*The use of virtual memory improves stability and helps prevent data loss on large projects.

'Recommended' does not guarantee that it meets your particular needs.

Tested Workstation Hardware

The configurations listed below are those that The Foundry have tested with Mari 3.0v3. Due to the constantly changing nature and wide variety of computer hardware available in the market, The Foundry is unable to officially certify hardware. The list below can be used as a recommendation and does not guarantee that it meets your particular needs.

Please download and install the latest graphics driver from the NVIDIA or AMD websites, and ensure that you are using 8.982.1 drivers or higher for AMD cards.

If you encounter any issues, please contact support@thefoundry.co.uk.

Tested Workstation Hardware	
Apple	• MacPro4,1 • MacPro5,1 • iMac13,2 • MacBookPro10,1
HP	• HP Z600 • HP Z620 • HP Z800

Tested GPU Hardware	
AMD Prosumer Graphics Cards	• AMD Radeon HD - D500 • AMD Radeon R7 - 260x
AMD Enterprise Graphics Cards	• AMD FirePro W7000
NVIDIA Prosumer Graphics Cards	• NVIDIA Geforce GTX 680 • NVIDIA Geforce GTX Titan • NVIDIA Geforce GT 650M
NVIDIA Enterprise Graphics Cards	• NVIDIA Quadro 4000 • NVIDIA Quadro K5000 • NVIDIA Quadro M6000

New Features

There are no new features for this release.

Feature Enhancements

- BUG ID 121132 - A new Python function named **runGLSLCode** has been added. It runs GLSL code and writes the result into the specified ImageSet with optional blending.
- BUG ID 123179 - It is now possible to automatically merge selection groups with the same name imported into Mari .
- BUG ID 174741 - In the **Channel Transfer** dialog, a new option to set the texture resolution of the destination channel has been added.
- BUG ID 175230 - Mari now detects whether to subdivide each Alembic geometry piece, based on subdivision flags set in the file.
- BUG ID 194832 - The accuracy of mip-map generation has been improved, at the cost of performance, by linearizing data before downsampling.

  To disable this, and restore legacy functionality and performance, a new option has been added to the **Preferences** called **Mip-Map Generation**.

Bug Fixes

- BUG ID 97445 - When using shared groups, Mari did not make it clear, on selection of a group, which other groups were shared, and therefore also selected. Additionally, shared layers and paint targets were also not clearly indicated.

- BUG ID 101109 - Applying a **Gaussian blur** filter caused Mari to crash.
- BUG ID 109277 - Node Graph: Clicking on any part of the pipe disconnected a node.
- BUG ID 109721 - Node Graph: The default input and output nodes of Graph Layers overlapped each other.
- BUG ID 110001 - Node Graph: Undoing an auto-insertion of a node required three undo actions.
- BUG ID 110465 - Transparent channels were exported when flattened.
- BUG ID 111329 - Node Graph: Auto-placing nodes within a backdrop disrupted the node positions.
- BUG ID 112666 - Node Graph: Paint nodes were created at the position of the cursor, wherever it was placed in the Mari interface, instead of in the visible part of the Node Graph.
- BUG ID 116323 - Node Graph: Layer icons in the **Layers** palette occasionally changed from paintable to procedural when a Dot node was added to the Node Graph.
- BUG ID 119781 - The current color profile set on the **Color Manager** wasn't applied to the screenshots.
- BUG ID 121798 - Changes made to the colorspace of an image in the **Image Manager** were not respected by the **Paint Through** tool.
- BUG ID 122619 - Removing channels from a project stopped the **Color Picker** working correctly.
- BUG ID 123448 - The **Channel Transfer** dialog showed objects that had been deleted in the available **Object** dropdown.
- BUG ID 126386 - Node Graph: The node connectors did not go back to its default position when disconnected.
- BUG ID 126970 - **Paint Buffer Symmetry** did not select the correct source point when painting using the **Clone Stamp** tool.
- BUG ID 128337 - In the **Shaders** palette, the **Display Features** sub-section showed incorrect titles with extra apostrophes for the new shaders **AiStandard**, **RedShiftArchitectural**, and **VrayMtl**.
- BUG ID 128730 - Color swatches didn't match the picked color when the **Color Swatches and Pickers** preference was set to **Color Manager**.
- BUG ID 129902 - Mari rendered turntables incorrectly when the **4096x2160** preset was used.
- BUG ID 129913 - Mari output rendered turntables in the incorrect colorspace for the selected image format.
- BUG ID 165965 - The subdivision error log was repeated for each level chosen and for merged geometries.
- BUG ID 167072 - In the OCIO **Colorspace** toolbar, clicking **Clear current LUT** disabled the **Display** filter of the corresponding color profile in the **Color Manager** palette.
- BUG ID 167239 - Resizing faces of a model whose UV data was missing caused Mari to crash.
- BUG ID 168832 - The **File Space** parameter, when adding a new channel through the Node Graph, was always set to the default value **Normal**, even when another **File Space** was specified through the **Add Channel** dialog.
- BUG ID 168923 - Exporting and importing Alembic geometry through session scripts did not take into account merging of geometry.
- BUG ID 169544 - Tiled procedurals were not correctly saved and referenced, causing Mari to lose them when the associated image was deleted from the **Image Manager**.
- BUG ID 170071 - The Abs node had a redundant "text" input.
- BUG ID 170079 - The Shuffle node was oval in shape rather than rectangular.

- BUG ID 170414 - Running the safe mode command `safe=7` caused Mari to crash.
- BUG ID 170682 - Environment lights didn't support the new OCIO colorspace system, making it impossible to use a colorspace specified in the project.
- BUG ID 170940 - The **Open Archive** dialog showed the wrong title.
- BUG ID 171562 - The **Clone Stamp** action in **Paint Through** mode did not work.
- BUG ID 171743 - The **Group Properties** palette's tree was expanded by default.
- BUG ID 171746 - Under the **Group Node Knobs** dialog, the **Group Properties** panel has been renamed to **Pick Knobs to Add**.
- BUG ID 171830 - Rendering turntables omitted some tiles and the final result displayed random missing sections in random frames.
- BUG ID 171831 - Cutting and pasting merged layers in the same channel resulted in a loss of data.
- BUG ID 172660 - Deleting the default input node from a Graph layer resulted in breaking the layer.
- BUG ID 172693 - When using the **Paint Through** and **Clone Stamp** tools on 8-bit image data, the paint displayed in the paint buffer appeared extremely pixelated compared to the expected resolution of the paint buffer.
- BUG ID 173066 - Copying and pasting layers across objects resulted in a loss of the paint information of the layers.
- BUG ID 173110 - The **scalar** option was missing from the colorspace settings on images, meaning that when painting masks or heights, unless you remembered to switch off the OCIO toolbar, colors came out too dark.
- BUG ID 173378 - When choosing a custom OCIO config file, the OCIO Python module resulted in an error.
- BUG ID 174570 - The **Repeat** mode of the **Paint Through** tool was not working correctly for the 8-bit paint buffer.
- BUG ID 174753 - When painting 8-bit image data, Mari introduced quantization errors due to the lack of precision. Baking, converting procedurals to paintable, mip-map generation, and so on introduced severe banding.
- BUG ID 174809 - The colorspace selected through the **Open an Image** dialog was ignored, and the image reverted and used the automatic colorspace instead.
- BUG ID 175718 - Mari introduced severe quantization errors when baking into, converting procedurals into paintable, and generating mip-maps on byte (8-bit) image data.
- BUG ID 176057 - Changing **File Space** in the Paint node properties caused Mari to crash.
- BUG ID 176388 - The roughness of the **AiStandard** shader was incorrectly calculated.
- BUG ID 177653 - Creating a Paint node using the color picker in the **Select Color** dialog caused Mari to become unresponsive.
- BUG ID 178398 - Mari's environment light lost the reference to its corresponding image and displayed nothing when upgrading between different versions releases of Mari 3.0.
- BUG ID 184235 - When stamping the gradient down, Mari filled the areas outside the gradient with black when they should have been unaffected.
- BUG ID 194958 - The camera movement around subdivided models didn't perform correctly in earlier versions of Mari 3.0.

Developer Notes

These are the changes relevant to developers.

New Features

PythonQt to PySide Change

In 3.0v1, Mari migrated from PythonQt to PySide for Python bindings. Although the functionalities remain the same, there are some differences such as function signatures, which may affect existing scripts. This section lists the changes to Mari's Python engine, and some notes regarding new features in Mari 3.0v1 as well as the changes resulting from the migration from PythonQt to PySide.

Major Changes

- Updated the version of Python shipped with Mari from 2.6.5 to 2.7.3.
- Changed the Python API bindings generation from PythonQt to PySide.
- Added Python API bindings for the new Node Graph feature.
- Added Python API bindings for the new OpenSubdiv feature.
- Added Python API bindings for the new Colorspace support.
- Added Python API bindings for the new Locator feature.
- Added Python API bindings to interact with new Image and Geometry C API plug-ins.
- Added many functions to OpenColorIO support.

Implicit Python API Changes Resulting from the Change to PySide Binding Generation

You should not use temporary objects when connecting to signals as they make PySide crash due to reference counting. Always store off the temporary object so a reference is held to it and it doesn't get garbage collected. The following example crashes:

```
def createComboBox():
  return PySide.QtGui.QComboBox()
def doSomething():
  print 'Hello World!'
createComboBox().currentIndexChanged.connect(doSomething)
```

As a workaround, do the following:

```
combo_box = createComboBox()
```

```
combo_box.currentIndexChanged.connect(doSomething)
```

NOTE: This is stated in the PySide documentation, see: https://wiki.qt.io/PySide_Shiboken_Object_Ownership#Common_pitfalls.

PySide is strict on types, so some functions do not accept the equivalent integer value to an enum. For example, the following no longer works:

```
channel = mari.current.channel()
channel.setDepth(16, 1)
```

Do the following instead:

```
channel.setDepth(mari.Image.DEPTH_HALF, mari.Channel.CONVERT_ALL)
```

List of Minor API Changes Since 2.6v5

- Exposed **MkdirCommand** to the Python API.
- Added **MriRegExpChannelNameExtractor** and Python API.
- Added extra signals to **ImageManager** for Python API.
- Added **transform()** method to Image Python API.
- Added project signal callbacks to Python API.
- Added Python bindings for selection group. This implied creating five more API objects, **SelectionGroupManager**, **SelectionGroup**, **ObjectSelectionGroup**, **FaceSelectionGroup** and **PatchSelectionGroup**. Also, added an example script in the Python examples folder that apppears in Mari Python examples menu.
- Texture Transfer - Added Python API for displacement, height, and Gaussian blur operations.
- Texture Transfer - Added supporting GUI for new operations chains through Python.
- Added select selection groups through Python.
- Node Graph API.
- Added Python API to support C API Image plug-in.
- Added code to pass extra parameters from Python to C API texture and geometry.
- OpenColorIO - Added **OpenColorIO** to Python API.
- OpenColorIO - Extended **ImageManager** in the Python API so the colorspace properties can be specified when opening an image.
- OpenColorIO - Extended Image in Python API so the colorspace properties can be accessed and modified.
- OpenColorIO - Extended Python API to include methods checking whether a colorspace or role is present in a configuration file.
- OpenColorIO - Made Python API throw an exception when a colorspace stage argument is out of range.
- OpenColorIO - Registered **ColorspaceType** enum with Python API.

- Custom Display Pythom API - Added Python function to create a PySide widget that hosts the custom widget for display driver API.
- Fixed Python API channel info decorator so it correctly sets the specified filespace.
- The old **COLOR** and **SCALAR** system in the Python API was deprecated.
- Added colorspace defaults changed signal to Python API.
- Added colorspace config changed signal to Python API.
- OpenColorIO - Added method to resolve automatic file names to the config class within the Python API.
- Added Python API for subdivision.
- Added method to Python API to determine whether a config file name is a standard one shipped with Mari.
- Added ability to specify the name of the colorspace config changed signal in the Python API.
- Added a Python API signal for changes made to a channel's scalar colorspace config.
- Renamed Python API class **EventWrapper** to **WrappedEvent** so that it would build and work with PySide.
- Switched Python API bindings over from PythonQt to PySide.
- Removed redundant PythonQt ways of signal connection in **mari.utils.connect()**.
- Added Python **LocatorEntity** class.
- Extended **GeoEntity** to get and set transforms.
- Extended **GeoManager** to list locators.
- Added pause shader compilation on the canvas in Python API.
- Added support for passing compression option from Python API to OpenEXR file writer.
- Re-enabled the **DIFFERENCE** blend mode for layers in the Python API.
- Added **hasAdjustmentStack**, **isAdjustmentStackEnabled**, **setAdjustmentStackEnabled** to Python API.
- Python API - Separated **ImageSet.importImages()** and **ImageSet.importImagesFromDict()** to support importing images from a dict of Image objects and file paths.

Feature Enhancements

- BUG ID 125486 - Mari now includes Python API documentation for Node Graph API.

Bug Fixes

- BUG ID 89624 - Python API : The **GeoEntityVersion.nameChanged** signal was not emitting correctly.
- BUG ID 123934 - Python API: The **Channel.exportImagesFlattened()** function did not respect the **BACKGROUND_EXPORT** flag.

Release Notes for Mari 3.0v2

Release Date

15 December 2015

WARNING: The information regarding the migration from PythonQt to PySide was unintentionally omitted from the first set of Release Notes for Mari 3.0v2, see PythonQt to PySide Change for more information.

System Requirements

NOTE: Mari increases its level of performance with newer, more advanced hardware configurations. However, Mari is tested and capable of operating on many older, prior-generation systems. For this reason we are listing below-minimum requirements, which are recommended, and on which tests have been performed. Your particular needs may vary from that of other users.

Officially Supported Operating Systems

- Mac OS X 10.9.5 (Mavericks) or higher
- Windows 7 64-bit or higher
- Linux 64-bit operating system (CentOS/RHEL 5.4)

Minimum Hardware Requirements

- Quad-core processor
- 10+GB disk space available for caching and temporary files
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers

- 1GB of graphics memory
- OpenGL 3.2* or higher

*Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

Recommended System Requirements

- 2.5+Ghz Quad-core processor
- 250+GB disk space available for caching and temporary files. SSD is preferable.
- 16GB RAM with additional virtual memory*
- Display with 1920 x 1080 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers
- 2+GB of graphics memory
- OpenGL 4.4 or higher support

*The use of virtual memory improves stability and helps prevent data loss on large projects.

'Recommended' does not guarantee that it meets your particular needs.

Tested Workstation Hardware

The configurations listed below are those that The Foundry have tested with Mari 3.0v3. Due to the constantly changing nature and wide variety of computer hardware available in the market, The Foundry is unable to officially certify hardware. The list below can be used as a recommendation and does not guarantee that it meets your particular needs.

Please download and install the latest graphics driver for the NVIDIA or AMD websites, and ensure that you are using 8.982.1 drivers or higher for AMD cards.

If you encounter any issues, please contact support@thefoundry.co.uk.

Tested Workstation Hardware	
Apple	• MacPro4,1 • MacPro5,1 • iMac13,2 • MacBookPro10,1
HP	• HP Z600 • HP Z620 • HP Z800

Tested GPU Hardware	
AMD Prosumer Graphics Cards	• AMD Radeon HD - D500 • AMD Radeon R7 - 260x
AMD Enterprise Graphics Cards	• AMD FirePro W7000
NVIDIA Prosumer Graphics Cards	• NVIDIA Geforce GTX 680 • NVIDIA Geforce GTX Titan • NVIDIA Geforce GT 650M
NVIDIA Enterprise Graphics Cards	• NVIDIA Quadro 4000 • NVIDIA Quadro K5000 • NVIDIA Quadro M6000

New Features

There are no new features for this release.

Feature Enhancements

- BUG ID 107889/48031 - A button has been added in the **Node Properties** palette to delete all the node property panels.
- BUG ID 126666/51423 - It is now possible to view detailed information about a node by selecting it and either pressing **I**, or right-clicking and selecting **Edit** > **Info Viewer**.
- BUG ID 127421/51517 - There is now an **R** (reset) button in the Mari Preferences, in the **Node Graph** tab for the **Advanced View** option.

Bug Fixes

- BUG ID 95164/44548 - EXR images were exported through the Nuke<>Mari bridge with overly high color values.
- BUG ID 99336/46312 - The **Hard Mix** blend mode didn't work correctly with **.hdr** images.
- BUG ID 99433/46361 - The **Vivid Light** blend mode didn't work correctly with **.hdr** images.
- BUG ID 112496/48031 - Caching and baking a node with multiple outputs didn't work correctly.
- BUG ID 118765/50117 - When added to Mari as merged geometry, **.fbx** and **.abc** geometry didn't preserve their selection sets.
- BUG ID 120314/50389 - The Modo **Bake** and **Render** functions were not enabled when Mari was launched using safe mode.

- BUG ID 123505/50878 - When converting a channel from 8-bit to 16-bit, the automatic colorspace remained in **sRGB** rather than **linear**.
- BUG ID 124200/51044 - The names of a merge node's inputs didn't reflect the ones in the **Node Properties**, although they meant the same thing.
- BUG ID 124993/51131 - When taking a snaphot, Mari didn't respect the colorspace of the file being written to.
- BUG ID 125514/51208 - Mari 3.0 was significantly slower at importing image data than Mari 2.6 and earlier.
- BUG ID 125799/51243 - The **Add Channel** dialog didn't appear when creating a Channel node.
- BUG ID 126101/51299 - Nodes sometimes jumped away from the position of the mouse, when it was moved after having panned in the Node Graph using the middle mouse button.
- BUG ID 126455/51383 - The Paint Buffer size was smaller than the canvas size upon first launch of Mari.
- BUG ID 126975/51476 - The canvas was slow to appear when changing tabs.
- BUG ID 127248/51510 - A Channel node created automatically at the start of a new project was not aligned properly with its first layer in the Node Graph.
- BUG ID 128148/51596 - The parameters **StartFrame** and **EndFrame** were not evaluated properly when trying to import a camera from an Alembic file, through Python.
- BUG ID 128202/51606 - Mari incorrectly brought up **Duplicate File Names** dialogs for PTEX export with $GEO keywords.
- BUG ID 128495/51655 - Trying to export Ptex textures for a scene consisting of lots of objects, caused Mari to become unresponsive.
- BUG ID 128695/51681 - Paint nodes, which had been exported and imported back in a gizmo file, were not deep-copied and thus shared paint with other paint nodes.
- BUG ID 128807/51702 - Mac OS X only: Subdividing certain Ptex objects caused Mari to crash.
- BUG ID 128843/51711 - When upgrading a project from an earlier version, Mari asked you whether you wanted the colorspace changes to be applied to each channel, even though the **All** button was pressed on the first channel converted.
- BUG ID 128987/51725 - Mari failed to reload the OCIO **Colorspace** toolbar when a Mari Python module was imported through an external script.
- BUG ID 166197/51874 - Creating a merged Ptex project from geometry that had multiple objects and some of which had normals missing caused Mari to crash.
- BUG ID 166255/51876 - The **Help** > **SDK** > **C API** > **Open Examples Folder** link opened the incorrect example folder.
- BUG ID 166260/51878 - Multiple Group nodes were created on top of each other if the first Group node was within the boundaries of a backdrop.
- BUG ID 166269/51880 - The **Help** > **SDK** > **Python** > **Open Examples Folder** link was incorrect and didn't open anything.
- BUG ID 166606/51891 - Using the color picker found in the **Select Color** dialog resulted in a color being in the wrong colorspace when the **Pick raw pixels** mode was used.
- BUG ID 166607/51892 - When picking colors through the **Select Color** dialog, the resulting color was in the incorrect colorspace when the **Pick screen pixels** mode was enabled.

- BUG ID 166975/51900 - Subdividing a model, the UV of which after subdivision required extra patches to be generated, caused Mari to crash. Mari generated redundant extra UV patches (UDIMs) for geometries where the UV just touched integer boundaries.
- BUG ID 167108/51912 - The link to Python in the documentation was outdated and pointed to an old version of Python.
- BUG ID 167228/51920 - Ptex face resizing didn't work for triangulated objects.
- BUG ID 167337/51920 - Loading some OBJ sequences with Ptex option enabled caused Mari to crash.
- BUG ID 167607/51924 - Sending byte channels to Nuke through the Nuke<>Mari bridge resulted in Mari becoming unresponsive.
- BUG ID 167672/51926 - Mac OS X only: Changing the resolution of a Paint node, in some cases, caused Mari to crash.
- BUG ID 167716/51927 - The tooltip to the "**p**" button in the Group node properties for promoting parameters was missing.
- BUG ID 167883/167883 - When upgrading a project that contained a **Tiled** procedural from 2.6 to 3.0, the frame rate dropped dramatically due to Mari continually loading the tile procedural's corresponding image every frame.

Developer Notes

These are the changes relevant to developers.

New Features

PythonQt to PySide Change

In 3.0v1, Mari migrated from PythonQt to PySide for Python bindings. Although the functionalities remain the same, there are some differences such as function signatures, which may affect existing scripts. This section lists the changes to Mari's Python engine, and some notes regarding new features in Mari 3.0v1 as well as the changes resulting from the migration from PythonQt to PySide.

Major Changes

- Updated the version of Python shipped with Mari from 2.6.5 to 2.7.3.
- Changed the Python API bindings generation from PythonQt to PySide.
- Added Python API bindings for the new Node Graph feature.
- Added Python API bindings for the new OpenSubdiv feature.
- Added Python API bindings for the new Colorspace support.
- Added Python API bindings for the new Locator feature.
- Added Python API bindings to interact with new Image and Geometry CAPI plug-ins.
- Added many functions to OpenColorIO support.

Implicit Python API Changes Resulting from the Change to PySide Binding Generation

You should not use temporary objects when connecting to signals as they make PySide crash due to reference counting. Always store off the temporary object so a reference is held to it and it doesn't get garbage collected. The following example crashes:

```
def createComboBox():
  return PySide.QtGui.QComboBox()
def doSomething():
  print 'Hello World!'
createComboBox().currentIndexChanged.connect(doSomething)
```

As a workaround, do the following:

```
combo_box = createComboBox()
combo_box.currentIndexChanged.connect(doSomething)
```

NOTE: This is stated in the PySide documentation, see: https://wiki.qt.io/PySide_Shiboken_Object_Ownership#Common_pitfalls.

PySide is strict on types, so some functions do not accept the equivalent integer value to an enum. For example, the following no longer works:

```
channel = mari.current.channel()
channel.setDepth(16, 1)
```

Do the following instead:

`channel.setDepth(mari.Image.DEPTH_HALF, mari.Channel.CONVERT_ALL)`

List of Minor API Changes Since 2.6v5

- Exposed **MkdirCommand** to the Python API.
- Added **MriRegExpChannelNameExtractor** and Python API.
- Added extra signals to **ImageManager** for Python API.
- Added **transform()** method to Image Python API.
- Added project signal callbacks to Python API.
- Added Python bindings for selection group. This implied creating five more API objects, **SelectionGroupManager**, **SelectionGroup**, **ObjectSelectionGroup**, **FaceSelectionGroup** and **PatchSelectionGroup**. Also, added an example script in the Python examples folder that apppears in Mari Python examples menu.
- Texture Transfer - Added Python API for displacement, height, and Gaussian blur operations.
- Texture Transfer - Added supporting GUI for new operations chains through Python.

- Added select selection groups through Python.
- Node Graph API.
- Added PythonAPI to support CAPI Image plug-in.
- Added code to pass extra parameters from Python to CAPI texture and geometry.
- OpenColorIO - Added **OpenColorIO** to Python API.
- OpenColorIO - Extended **ImageManager** in the Python API so the colorspace properties can be specified when opening an image.
- OpenColorIO - Extended Image in Python API so the colorspace properties can be accessed and modified.
- OpenColorIO - Extended Python API to include methods checking whether a colorspace or role is present in a configuration file.
- OpenColorIO - Made Python API throw an exception when a colorspace stage argument is out of range.
- OpenColorIO - Registered **ColorspaceType** enum with Python API.
- Custom Display Pythom API - Added Python function to create a PySide widget that hosts the custom widget for display driver API.
- Fixed Python API channel info decorator so it correctly sets the specified filespace.
- Deprecated old **COLOR** and **SCALAR** system in Python API.
- Added colorspace defaults changed signal to Python API.
- Added colorspace config changed signal to Python API.
- OpenColorIO - Added method to resolve automatic file names to the config class within the Python API.
- Added Python API for subdivision.
- Added method to Python API to determine whether a config file name is a standard one shipped with Mari.
- Added ability to specify the name of the colorspace config changed signal in the Python API.
- Added a Python API signal for changes made to a channel's scalar colorspace config.
- Renamed Python API class **EventWrapper** to **WrappedEvent** so that it would build and work with PySide.
- Switched Python API bindings over from PythonQt to PySide.
- Removed redundant PythonQt ways of signal connection in **mari.utils.connect()**.
- Added Python **LocatorEntity** class.
- Extended **GeoEntity** to get and set transforms.
- Extended **GeoManager** to list locators.
- Added pause shader compilation on the canvas in Python API.
- Added support for passing compression option from Python API to OpenEXR file writer.
- Re-enabled the **DIFFERENCE** blend mode for layers in the Python API.
- Added **hasAdjustmentStack**, **isAdjustmentStackEnabled**, **setAdjustmentStackEnabled** to Python API.
- Python API - Separated **ImageSet.importImages()** and **ImageSet.importImagesFromDict()** to support importing images from a dict of Image objects and file paths.

Feature Enhancements

There are no feature enhancements in this release.

Bug Fixes

- BUG ID 91297/43466 - Printing to the Python buffer became increasingly slower as more messages were printed. The performance of Python script console output has been improved and a new Python API function has been added, **mari.app.clearScriptOutput()**, to clear the Python script output.
- BUG ID 118423/50057 - Python for the Texture Access API didn't work.
- BUG ID 120342/50391 - The shaderAdded signal was emitted before the shader name was set. **GeoEntity.shaderAdded** now emits after the name is set. The **GeoEntity.shaderAdded** function has been deprecated, which still works. The **GeoEntity.shaderCreated** signal has been introduced for consistency, and fixed **GeoEntity.shaderRemoved**, which was not emitted correctly.
- BUG ID 128312/51631 - Python API: **PaintBuffer.depthChanged** was not emitted.
- BUG ID 129295/51774 - Mari exported an image in the wrong colorspace when done through the Python API by setting each image's colorspace in an image set.
- BUG ID 129160/51748 - Python documentation didn't list the correct function argument names.
- BUG ID 166679/51895 - Loading C API animation failed.
  The restriction on animation geometry loading has been eased so that some missing data for some animation frames is supplemented by another frame.

Release Notes for Mari 3.0v1

Release Date

02 December 2015

WARNING: The information regarding the migration from PythonQt to PySide was unintentionally omitted from the first set of Release Notes for Mari 3.0v1, see PythonQt to PySide Change for more information.

System Requirements

NOTE: Mari increases its level of performance with newer, more advanced hardware configurations. However, Mari is tested and capable of operating on many older, prior-generation systems. For this reason we are listing below-minimum requirements, recommended, and on which tests have been performed. Your particular needs may vary from that of other users.

Officially Supported Operating Systems

- Mac OS X 10.9.5 (Mavericks) or higher
- Windows 7 64-bit or higher
- Linux 64-bit operating system (CentOS/RHEL 5.4)

Minimum Hardware Requirements

- Quad-core processor
- 10+GB disk space available for caching and temporary files
- At least 4GB RAM
- Display with 1680 x 1050 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers

- 1GB of graphics memory
- OpenGL 3.2* or higher

*Displacement preview is currently only available on the cards and drivers that support OpenGL 4.0 or newer.

Recommended System Requirements

- 2.5+Ghz Quad-core processor
- 250+GB disk space available for caching and temporary files. SSD is preferable.
- 16GB RAM with additional virtual memory*
- Display with 1920 x 1080 pixel resolution
- An NVIDIA or AMD* graphics card with the latest drivers
- 2+GB of graphics memory
- OpenGL 4.4 or higher support

*The use of virtual memory improves stability and helps prevent data loss on large projects.

'Recommended' does not guarantee that it meets your particular needs.

Tested Workstation Hardware

The configurations listed below are those that The Foundry have tested with Mari 3.0v3. Due to the constantly changing nature and wide variety of computer hardware available in the market, The Foundry is unable to officially certify hardware. The below can be used as a recommendation and does not guarantee that it meets your particular needs.

Please download and install the latest graphics driver for the NVIDIA or AMD websites, and ensure that you are using 8.982.1 drivers or higher for AMD cards.

If you encounter any issues, please contact support@thefoundry.co.uk.

Tested Workstation Hardware	
Apple	• MacPro4,1 • MacPro5,1 • iMac13,2 • MacBookPro10,1
HP	• HP Z600 • HP Z620 • HP Z800

Tested GPU Hardware	
AMD Prosumer Graphics Cards	• AMD Radeon HD - D500 • AMD Radeon R7 - 260x
AMD Enterprise Graphics Cards	• AMD FirePro W7000
NVIDIA Prosumer Graphics Cards	• NVIDIA Geforce GTX 680 • NVIDIA Geforce GTX Titan • NVIDIA Geforce GT 650M
NVIDIA Enterprise Graphics Cards	• NVIDIA Quadro 4000 • NVIDIA Quadro K5000 • NVIDIA Quadro M6000

New Features

Node Graph

- Mari's under-the-hood Node Graph is accessible through the **Node Graph** palette, allowing you to control the layering system more accurately. Nodes are created and manipulated automatically as you interact with the layer system, but you can manipulate individual nodes in the **Node Properties** palette to achieve the required result.
- Mari's Node Graph has two modes, **Basic** and **Advanced**, exposing different levels of detail within the palette. Basic mode, the default mode, only allows you to work with Graph layers from the **Layers** palette. **Advanced** mode exposes the entire project in the **Node Graph** palette, including all layers, channels, and shaders.

NOTE: You can enable the **Advanced** Node Graph in the **Preferences** dialog under **Node Graph** > **General** > **Advanced View**.

Gizmos and Publishing

- Group nodes can be saved as **.gizmo** files, allowing you to save layers from the Node Graph and share them in a collaborative workflow. The property view of a **.gizmo** allows you to select which parameters of the member nodes to expose in the exported gizmo.

  Gizmos can be loaded back into Mari by:

  - registering the **.gizmo** in the Python API,
  - registering the **.gizmo** in a node catalog file, or
  - copying the **.gizmo** into the **~/Mari/Gizmos** directory, which is searched automatically by Mari at startup.

Baking Using Modo

- Mari can now round-trip texture bakes to Modo, allowing you to modify parameters and to see the preview update. You can add new bake recipes easily by creating **.lxo** files.

Modo Render Integration

- Mari now includes a **Modo Render** script that walks Mari's Node Graph and sends the shading information to Modo. Modo then renders the scene using its offline renderer.

  The **Modo Render** script also contains examples of how to integrate Mari and an offline renderer.
- Mari uses the following environment variables for loading Bake and Render recipes: **MARI_MODO_BAKE_PRESETS** and **MARI_MODO_RENDER_PRESETS**.

Per Channel Color Management

- Colorspace properties are now available on a per-image and per-channel basis through the third-party OpenColorIO library. You can also specify a project default, similar to Nuke's **OpenColorIO config** setting.

Open Subdivision Support

- Mari now supports geometry subdivision, providing the following benefits:
  - Observable smooth surfaces similar to the final offline renderer's subdivision,
  - UV layouts can be modified by subdivision, allowing you to view more accurate UV layouts, and
  - No need to pre-subdivide geometry.
- The **Subdivision** dialog includes the option to calculate subdivision in the foreground, by clicking **OK**, or background by clicking **In Background**.
  - **OK** - a progress bar displays as Mari calculates the subdivision. Once complete, the level of subdivision specified is applied to the object immediately. An error dialog displays if subdivision fails when run in the foreground.
  - **In Background** - the subdivision calculation is submitted to a background process and no progress bar is displayed. The subdivision **Level** must be applied manually in the **Objects** palette when the "running man" icon disappears from the Status Bar.

Arnold, V-Ray, Unreal, and Redshift Shaders

- Mari now includes Arnold, V-Ray, Unreal, and Redshift shaders, allowing you to paint textures underneath more realistic industry-standard shaders, improving the pre-visualization of their textures.

Session Scripts

- Session Scripts allow you to create a light weight archive of Mari projects to merge the data into other projects. Session Scripts can be used to see assets in context with each other or to share data, such as images, channels, and shaders. The data can also be used as a template system for setting up projects with default channels, shaders, and so on.

Texture Transfer Through PartIO

- Mari now allows you to generate height and normal maps between high- and low-polygon models, including seamless blurring across patch boundaries and overlapping UVs by texture transfer.

Entity Locators

- Mari's **Move** tool now displays axes and the direction of movement, and allows you to rotate and scale geometry.

FBX Geometry Support

- Mari now supports importing geometry from **.fbx** files, including the selection of specific child geometry from the file using the tree widget.

Selection Groups

- You can now manipulate selection groups using the Python API, allowing you to automate selection group-related functionality in your pipeline.
- BUG ID 42287 - A new smart selection type, called **Selection Group**, has been added to allow you to cycle through the various selection groups to which a selected face belongs.

Python Upgrade

- Mari now ships with Python 2.7.3, in line with other leading industry products.

Safe Mode

- Mari now includes **--safe** and **--safer** startup modes, similar to Nuke's, with different logging levels, which can help diagnose problems when running Mari.

Miscellaneous

- BUG ID 47144 - The **Selection Mask** layer/node type has been added to the **Procedural** > **Geometry** layer menu. This layer/node outputs the current selection as 1.

Feature Enhancements

- You can now fill patches with **Face** selection enabled, as you would with objects or patches. Simply select the faces you want to fill, and select **Patches** > **Fill** from either the menubar or right-click menu.
- You can now create a Pass Through blend mode for **Group** layers that work similarly to the Pass Through blend mode in Photoshop. In the **Layers** palette, with the **Group** layer selected, click on the Pass Through icon.
- BUG ID 43764 - Object metadata is now written into new versions of loaded **.abc** or **.fbx** geometry, so that the metadata can be read later on and the scene recreated using Session Scripts.

- BUG ID 46073 - Support has been added for passing compression options for the Python API to the OpenEXR file writer.
- BUG ID 50082 - The **Data** > **Cache** category has been renamed to **Data** > **Project** in the Mari Preferences.

Bug Fixes

- BUG ID 32251 - Pressing the **PgUp** and **PgDown** keys moved between the previous channel and next channel, rather than moving to the top and bottom of the visible area of the channel list.
- BUG ID 34502 - It was not possible to launch Mari with user-customizations disabled.

  Two modes have now been added: **--safe** and **--safer**. This disables user-customizations, among other things.
- BUG ID 34542 - The shortcut to toggle the **Tool Properties** palette (**F7**) was unassigned, and it no longer appeared in the **Manage Keyboard Shortcuts** dialog.
- BUG ID 37717 - In some cases, Mari crashed when trying to save.
- BUG ID 37927 - It was not possible to rename objects from the **name** field in the **Objects** palette.
- BUG ID 40963 - Mari crashed when loading projectors using the GUI.
- BUG ID 41487 - Mari did not recognize OpenColorIO 'looks' contained in an OCIO color configuration file, such as config.ocio.
- BUG ID 42426 - There was no native support for the V-Ray shader.
- BUG ID 42861 - There was a typo in the shader source code files that meant "Bitangent" appeared as "biTangent".
- BUG ID 43732 - Attempting to undo caching did not always undo the action.
- BUG ID 44964 - Copying a group layer or layer mask did not deep-copy the image sets of the member layers.
- BUG ID 47732 - The Nuke<>Mari Bridge mapping scheme was serializing incorrectly, causing the bridge script to fail.
- BUG ID 47749 - The default layout of the layer stack meant that the procedural properties section was mostly obscured from view.
- BUG ID 48268 - Mari read the **MARI_PLUGINS_PATH** environment variable in the opposite order to the expected UNIX order, giving precedence to those entries nearest the end of the string.
- BUG ID 48861 - The colorspace toolbar locked up and did not respond to the toggle on/off management ⬡ button.
- BUG ID 49335 - Converting multiple procedural layers to be paintable sometimes failed or caused Mari to crash, as the order in which they were processed could cause them to conflict.
- BUG ID 49600 - Mari sometimes offset each patch in a project by -1.
- BUG ID 49671 - The Nuke<>Mari Bridge was failing to send components under certain conditions.
- BUG ID 49690 - Repeatedly copying a channel when an object's UV map encroached into negative space, caused Mari to crash.
- BUG ID 51062 - Mac OS X only: Menu items were grayed out and unavailable if that particular Mari session was the first to create the **Mari.ini** file.

Developer Notes

These are the changes relevant to developers.

New Features

PythonQt to PySide Change

In 3.0v1, Mari migrated from PythonQt to PySide for Python bindings. Although the functionalities remain the same, there are some differences such as function signatures, which may affect existing scripts. This section lists the changes to Mari's Python engine, and some notes regarding new features in Mari 3.0v1 as well as the changes resulting from the migration from PythonQt to PySide.

Major Changes

- Updated the version of Python shipped with Mari from 2.6.5 to 2.7.3.
- Changed the Python API bindings generation from PythonQt to PySide.
- Added Python API bindings for the new Node Graph feature.
- Added Python API bindings for the new OpenSubdiv feature.
- Added Python API bindings for the new Colorspace support.
- Added Python API bindings for the new Locator feature.
- Added Python API bindings to interact with new Image and Geometry CAPI plug-ins.
- Added many functions to OpenColorIO support.

Implicit Python API Changes Resulting from the Change to PySide Binding Generation

You should not use temporary objects when connecting to signals as they make PySide crash due to reference counting. Always store off the temporary object so a reference is held to it and it doesn't get garbage collected. The following example crashes:

```
def createComboBox():
  return PySide.QtGui.QComboBox()
def doSomething():
  print 'Hello World!'
createComboBox().currentIndexChanged.connect(doSomething)
```

As a workaround, do the following:

```
combo_box = createComboBox()
combo_box.currentIndexChanged.connect(doSomething)
```

NOTE: This is stated in the PySide documentation, see: https://wiki.qt.io/PySide_Shiboken_Object_Ownership#Common_pitfalls.

PySide is strict on types, so some functions do not accept the equivalent integer value to an enum. For example, the following no longer works:

```
channel = mari.current.channel()
channel.setDepth(16, 1)
```

Do the following instead:

```
channel.setDepth(mari.Image.DEPTH_HALF, mari.Channel.CONVERT_ALL)
```

List of Minor API Changes Since 2.6v5

- Exposed **MkdirCommand** to the Python API.
- Added **MriRegExpChannelNameExtractor** and Python API.
- Added extra signals to **ImageManager** for Python API.
- Added **transform()** method to Image Python API.
- Added project signal callbacks to Python API.
- Added Python bindings for selection group. This implied creating five more API objects, **SelectionGroupManager**, **SelectionGroup**, **ObjectSelectionGroup**, **FaceSelectionGroup** and **PatchSelectionGroup**. Also, added an example script in the Python examples folder that apppears in Mari Python examples menu.
- Texture Transfer - Added Python API for displacement, height, and Gaussian blur operations.
- Texture Transfer - Added supporting GUI for new operations chains through Python.
- Added select selection groups through Python.
- Node Graph API.
- Added PythonAPI to support CAPI Image plug-in.
- Added code to pass extra parameters from Python to CAPI texture and geometry.
- OpenColorIO - Added **OpenColorIO** to Python API.
- OpenColorIO - Extended **ImageManager** in the Python API so the colorspace properties can be specified when opening an image.
- OpenColorIO - Extended Image in Python API so the colorspace properties can be accessed and modified.
- OpenColorIO - Extended Python API to include methods checking whether a colorspace or role is present in a configuration file.
- OpenColorIO - Made Python API throw an exception when a colorspace stage argument is out of range.
- OpenColorIO - Registered **ColorspaceType** enum with Python API.

- Custom Display Pythom API - Added Python function to create a PySide widget that hosts the custom widget for display driver API.
- Fixed Python API channel info decorator so it correctly sets the specified filespace.
- Deprecated old **COLOR** and **SCALAR** system in Python API.
- Added colorspace defaults changed signal to Python API.
- Added colorspace config changed signal to Python API.
- OpenColorIO - Added method to resolve automatic file names to the config class within the Python API.
- Added Python API for subdivision.
- Added method to Python API to determine whether a config file name is a standard one shipped with Mari.
- Added ability to specify the name of the colorspace config changed signal in the Python API.
- Added a Python API signal for changes made to a channel's scalar colorspace config.
- Renamed Python API class **EventWrapper** to **WrappedEvent** so that it would build and work with PySide.
- Switched Python API bindings over from PythonQt to PySide.
- Removed redundant PythonQt ways of signal connection in **mari.utils.connect()**.
- Added Python **LocatorEntity** class.
- Extended **GeoEntity** to get and set transforms.
- Extended **GeoManager** to list locators.
- Added pause shader compilation on the canvas in Python API.
- Added support for passing compression option from Python API to OpenEXR file writer.
- Re-enabled the **DIFFERENCE** blend mode for layers in the Python API.
- Added **hasAdjustmentStack**, **isAdjustmentStackEnabled**, **setAdjustmentStackEnabled** to Python API.
- Python API - Separated **ImageSet.importImages()** and **ImageSet.importImagesFromDict()** to support importing images from a dict of Image objects and file paths.

Texture Access API

- A new C API has been added, allowing you to manipulate texture data live in Mari to improve the inter-operability of Mari with external renderers. The C API allows external renders to request specific texture data and apply it to geometry when rendering the scene from Mari.
- Accessing texture data using the C API provides the external application with real-time uncompressed raw data, managed by Mari's DataManager, allowing for efficient memory paging.

Display Driver API

- A new C API has been added, allowing you to write your own Mari plug-ins to receive and display external, streamed image data such as external renderer results.

  The Display Driver API enables Mari to be integrated into LookDev pipelines more closely because it forms a part of LookDev integration in which Mari sends texture and shading information to external rerenderers and Mari receives the rendered result.

Custom Graph Context API

- A new C API has been added, allowing you to add contextual information to a node in Mari's Node Graph.

  The Custom Graph Context API enables Mari to be integrated into LookDev pipelines more closely.

  The example usages of Custom Graph Context API includes:

  - an external real-time viewport,
  - custom exporting and importing,
  - adding custom attributes,
  - external and network rendering or baking, and
  - porting shading information to other software.

Geometry Access API

- The Mari C API has been extended to include two more geometry access APIs, **MriGeometryAccessPlugin** and **MriUserGeometryOperationPlugin**. These allow you to create plug-ins to access the geometries inside Mari projects and to perform operations on those geometries.

  The **Example OBJ Exporter**, a built-in example geometry access plug-in, is also available. It demonstrates the ability to access the geometry and perform operations on it, and exports the geometry in the **.obj** format.

Miscellaneous

- BUG ID 51590 - You can clear all node properties from the **Node Properties** palette by entering the following command in the Python Console:

  ```
  b = mari.actions.get("/Mari/NodeGraph/Delete All Properties") b.trigger()
  ```

Feature Enhancements

- BUG ID 44024 - A Python API method, **removeKeyFrame**, has been added to remove key frames from an imageset.
- BUG ID 46034 - Example code has been added to the *Python Developer's Guide* to create a projector from an Alembic file.
- BUG ID 46073 - Support has been added for passing compression options for the Python API to the OpenEXR file writer.
- BUG ID 46539 - Notes have been added to **GeoEntity.patchList()** and **GeoEntityVersion.patchList()** to indicate that GeoEntity returns the combined list of patches from all versions while GeoEntityVersion returns the list of patches only from the version.
- BUG ID 49554 - Python API: Mari's **AdjustableLayer** class now includes **isAdjustmentStackEnabled()** and **setAdjustmentStackEnabled()** methods.
- BUG ID 50171 - Python API: The **ImageSet.importImages()** and **ImageSet.importImagesFromDict()** functions have been separated in order to support importing images from a dict of Image objects and file paths.

Bug Fixes

- BUG ID 47138 - Linux only: The Mari bundle did not contain the **libQtDesigner.so** file, which meant that QtUiTools was unavailable.
- BUG ID 48239 - The QtUiTools module was missing from Mari's compilation of PySide.