FTV469105

Arnold + USD = Love

Frederic Servant
Autodesk

Sebastien Blaineau-Ortega
Autodesk

Pal Mezei
Autodesk

Learning Objectives

- Learn how to import and export USD from any of the applications supported by Arnold.
- Learn how to use Arnold and USD to build studio pipelines.
- Learn how to use Arnold and USD in production today.
- Learn about the Arnold commitment to open source for USD.

Description

Maximize the power of Pixar's USD with Arnold software in production scenarios. We'll show examples of exporting and importing USD assets—including full shading, material assignments, and render setup—from any of the applications that Arnold supports, such as Maya software, 3ds Max software, Houdini, or Cinema 4D. We'll also demonstrate how you can take advantage of the power of USD's composition and variants to address common scenarios in modern studio pipelines. All of Arnold software's effort for USD is open source. You'll learn what's out there and how you can contribute.

Speakers

Frederic is the Arnold Software Development Manager and is based in London, where he looks over the Arnold core and plugins teams scattered around the globe. He was previously the main developer of the Arnold integration in Houdini. Prior to working on Arnold, Frederic worked as an R&D Engineer at The Mill in London and La Maison in Paris.

Sebastien has been working on the Arnold-to-Maya plugin (MtoA) for several years, and on the USD support in Arnold. He developed the Arnold RenderView, and works occasionally on Arnold core. He works remotely from a small village near Marseille, France. Prior to joining the Arnold team, he worked at “The Bakery” on a lighting tool called Relight, and in French VFX studios (BUF and Def2Shoot).

Pal is a software engineer with more than 12 years of professional experience, in animation, visual effects and software development. He started his career working on commercials and trailers for games like Mass Effect and Assassin’s Creed. He first joined the Arnold team in 2012 for 3 years before heading to the land of roos to work on Marvel movies like Antman, Avengers or Doctor Strange. In his free time, he likes playing games, writing silly stuff, climbing the ivory tower and dreaming about the future where programming is memory safe, high performance and functional.

What is Arnold?

If you're not familiar with **Arnold**, here's a few key points:

- Arnold was one of the first production path tracers, something that was recognized by a Scitech award. Arnold is used in hundreds of studios worldwide.
- Arnold scales well for large production scenes and tackles the complex rendering challenges such as skin, hair, volumes needed in features films.
- Arnold is customizable through a powerful API in C++ and Python.
- Arnold support both CPU and GPU rendering, and you switch interactively from one device to the other.
- Arnold now ships with Maya and Max, and it's the default renderer in those applications.
- Arnold is intergrated in Houdini, Cinema4D, Katana, and also in unsupported applications such as Gaffer or Blender.

See https://www.arnoldrenderer.com/

What is USD?

Universal Scene Description (**USD**) is a framework for interchange of 3D computer graphics data, created by Pixar. The framework focuses on collaboration, non-destructive editing, and enabling multiple views and opinions about graphics data.

See https://graphics.pixar.com/usd/docs/Introduction-to-USD.html

Arnold USD is open source

At SIGGRAPH last year we had announced that the Arnold components for USD were being open sourced on Github under a very permissive Apache 2.0 license.

While you can compile these components yourself, we ship them in the Arnold core SDK and our various application plugins, so you can use them out of the box.

The development process itself is done openly. Here you see an ongoing sprint and you can follow the development work being updated live. The current priority of features and bugs is also available and up for discussion.

We encourage you to submit feature requests and report bugs on Github. Of course, we also welcome your contributions and pull requests.

See https://github.com/Autodesk/arnold-usd

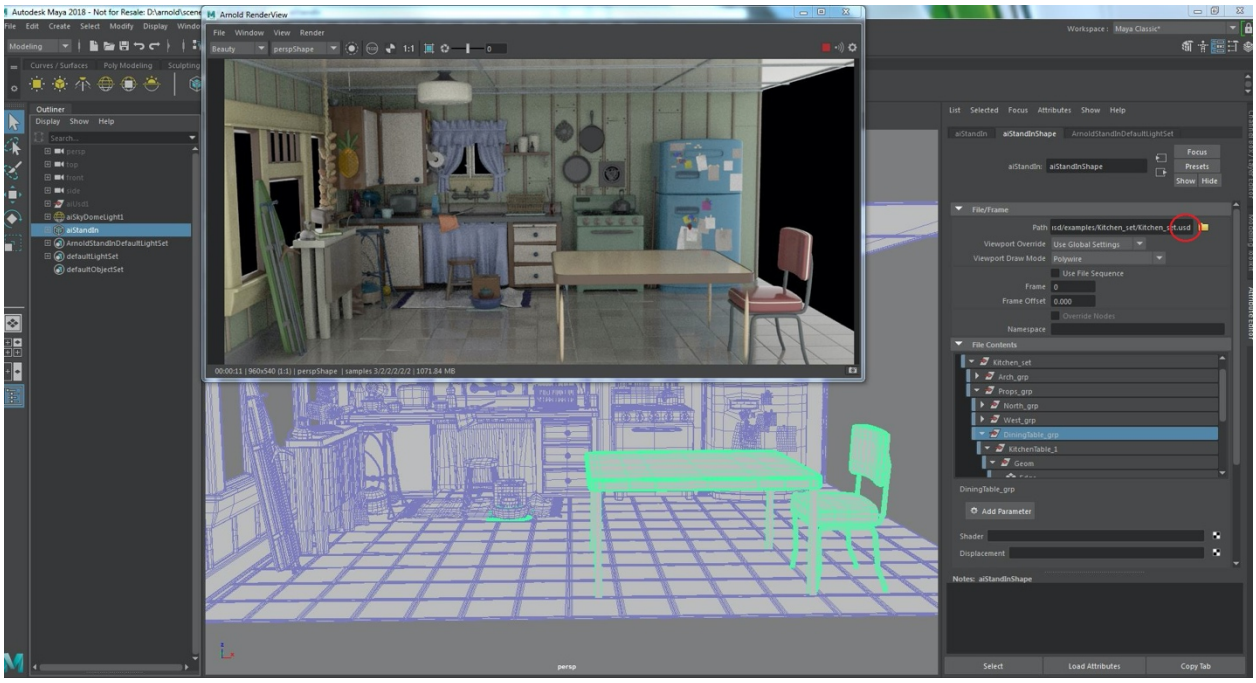
Using USD data with Arnold

There are many ways to ingest USD data with Arnold.

USD Procedural

The simplest way to consume USD data in Arnold is to use the **USD Procedural**. It's just another procedural, similar to the Alembic or the .ass file procedurals. What this means is that you can use the workflows that you know and love in any of the applications we support, but with USD data.

For example, in Maya, you can introspect the USD content, select items in the viewport or in the renderview, and use operators for parameter overrides and assigning shaders.



Sebastien demonstrates the procedural in the video presentation, showcasing these workflows in depth.

Shapes

The procedural supports the vast majority of USD shapes. It's important to note that **primvars are translated as user data** and thus available in Arnold to drive the shading for example.

Shaders

While we support the few native USD shaders, this is not enough for production rendering. That's why **we also support all the Arnold shaders as UsdShade nodes**, where the **info:id** attribute contains the Arnold shader type.

Arnold Extensions

We also provide **USD extensions to support all the Arnold nodes as USD nodes**. USD does not have a concept of operators or a definition for implicit surfaces for example.

You can also set Arnold-specific attributes on native USD shapes, like subdivision iterations on a mesh. They need to be prefixed with the **arnold:** namespace.

USD Scene

In Arnold 6.0.2, we introduced a **new scene file format API**. This API allows you to create plugins to define how you load and save an Arnold scene in a custom file format. You've guessed it, we have such a plugin for USD. Note that .ass, Alembic, and other file formats, are also implemented as such.

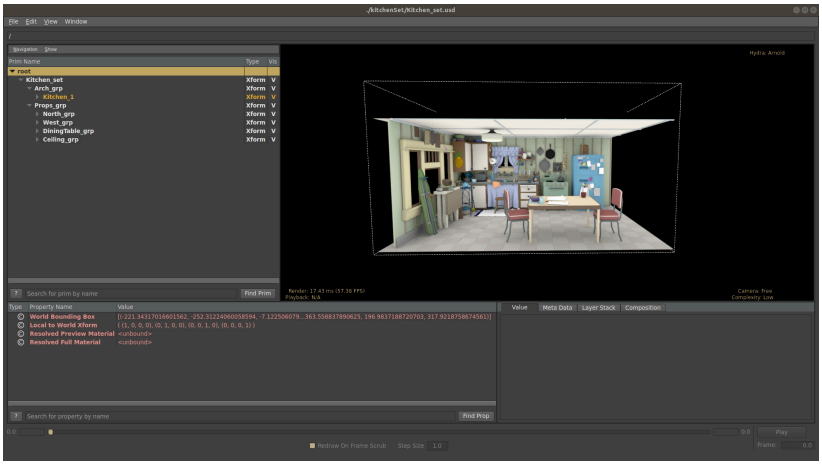
This means that we can now **import and export Arnold scenes transparently in USD from any application** we support. Moreover, the USD file will contain all the information needed to render with Arnold, including shaders, operators, render options, etc. like an .ass file.

You can also import and export parts of the scene such as shading networks in USD.

Because we have a 1:1 mapping with .ass files, **we can render USD scenes directly** with the kick command line tool. Contrarily to .ass files, because USD can contain an animation, you need to specify which frame you'd like to render.

Hydra Render Delegate

Another important USD component is the Arnold Hydra render delegate. The render delegate is very important for **interactive rendering** and **easy integration in the applications that support Hydra**, such as usdview, Houdini Solaris, Katana, NVIDIA Omniverse or Maya USD.



Stay tuned for more details about Hydra and see the Arnold render delegate in action, demonstrated by Pal in the video presentation.

Conclusion and next steps

This concludes our tour of USD in Arnold and I'd like to talk a little bit about what's next.

If you browse the tickets on Github, you'll see that there still a lot to do, and this is where we need your help with development priorities. If you are an Arnold customer and are interested in deploying Arnold and USD in production, **we'd like to hear from you**. Tell us what is blocking, what is not working, what you require for your USD pipeline.

This is exactly what we have been doing with a select number of studios and it's been incredibly helpful. Big thanks to:

- Luma Pictures
- RodeoFX
- Superprod
- Qvisten Animation
- Megalis
- Skydance

We'd also like to thank the following software vendors. We've managed to collaborate in good spirits and our common customers and the entire industry are benefiting from it, which is the point of USD.

- Pixar
- SideFX
- OTOY
- The Foundry