

## 4.1.0.71

05 Aug 2020

MAXtoA 4.1.0 is a feature release introducing the Arnold RenderView, initial support for tyFlow, enhanced baking to texture.

This version uses the **Arnold 6.0.4.0** core, see the [Release Notes](#).

### Installation

1. Download [Arnold for 3ds Max](#).
2. Follow the [installation instructions](#).

 **Autodesk Network Licensing** in Arnold 6 requires new license files with an updated **2020 version**. Please follow the instructions on [this page](#) to generate your license file. More info about Arnold 6 licensing can be found [here](#).

### Compatible 3ds Max Versions

- 3ds Max 2019
- 3ds Max 2020
- 3ds Max 2021

### System Requirements

- Windows 7 or later, with the Visual Studio 2015 redistributable
- CPUs need to support the SSE4.1 instruction set
- Optix™ denoiser requires an NVidia GPU with [CUDA™ Compute Capability 5.0](#) and above
- Arnold GPU requires an NVIDIA GPU of the Turing, Volta, Pascal, or Maxwell architecture. We recommend using the [451.77](#) or higher drivers on Windows. See [Getting Started with Arnold GPU](#) for more information.

### Enhancements

- **Arnold RenderView**: an interactive rendering (IPR) utility designed to give real-time feedback on any changes made to the scene while solving the limitations of the legacy render view.
- **tyFlow instancer**: support for both the 'Shape/Mesh' operators and tyFlow's new 'Instance Node' operator (WIP). Tested internally with tyFlow v0.16089.
- **Baking using GPU** is now matching CPU, except for some very specific elements such as Lighting (3ds Max 2021 only).

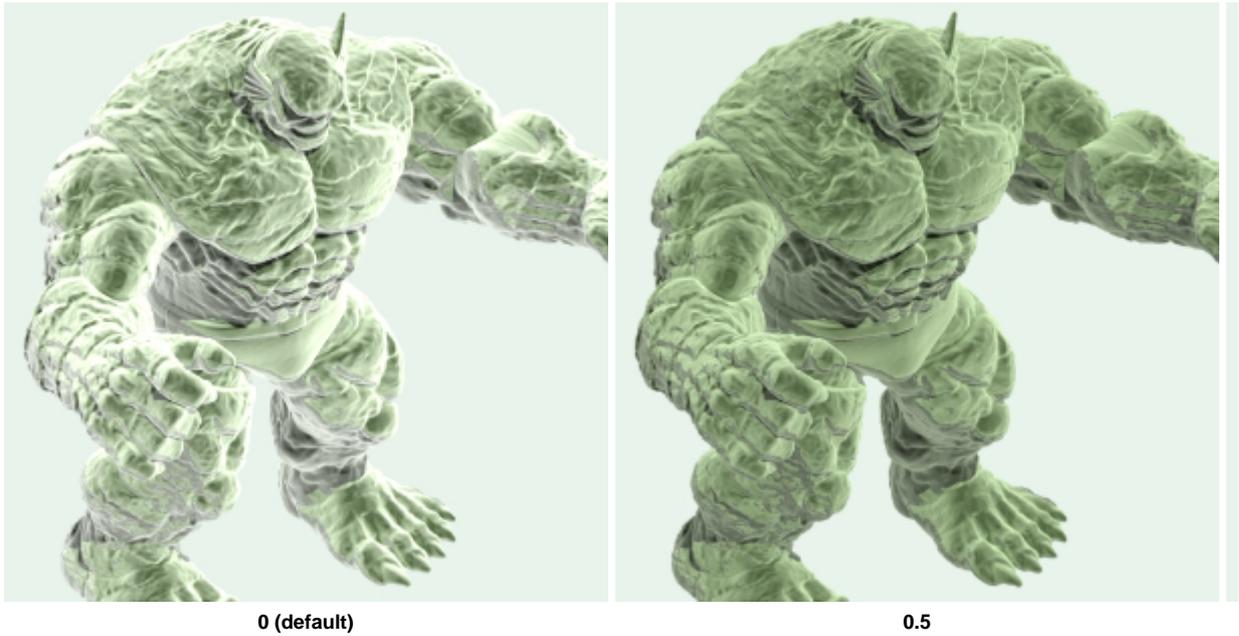
- **Baking:** several projected elements can now be baked in a single render pass, instead of one pass per each element (3ds Max 2021.2 only).
- **Baking:** roughness and metalness can now be baked also when the parameters are unconnected (3ds Max 2021 only).
- **Logging to file:** leveraging the new messaging API is fixing the problem of void or incomplete rendering log files.
- **Mikk Tangents** and binormals can now be exported as vector user data, through the Arnold Properties modifier (3ds Max 2021 only).
- Enhanced tooltips for the Arnold lights and properties modifier.

## Arnold Enhancements

- **Automatic file path remapping:** String parameter values can now be automatically modified according to the OS Arnold is running on. For example, this is useful when the artist workstations are running Windows but the renderfarm nodes are running Linux. Configuration is done with a JSON file that contains the strings to search and replace for each OS. This configuration file can be selected through the `ARNOLD_PATHMAP` environment variable. (#9311) For example:

```
{
  "windows":
  {
    "this_path/": "some/other_path/windows/"
  },
  "mac":
  {
    "this_path/": "some/other_path/mac/"
  },
  "linux":
  {
    "this_path/": "some/other_path/linux/"
  }
}
```

- **Rim light control for the Toon shader:** A new `rim_light_tint` float parameter was added to the `toon` shader to tint the color of the rim light using the base color of the surface. See the [full documentation](#). (#9003)



- Performance improvements on Windows with many cores and texture-heavy scenes:** Several performance improvements have been made, particularly for those running Windows, using many cores, tiff based textures, and/or reading many textures/sec. Expected speedups can be anywhere from a few percent to orders of magnitude faster on many-core Windows machines. (#9515, #9685, #9686, #9695, #9709, #9761) For a medium complexity scene that uses exr based .tx textures we get the following speedups on Windows :

CPU	Arnold 6.0.3.0	Arnold 6.0.4.0	Speedup
AMD 3970X (64 threads)	439s	397s	1.11x
AMD 3990X (128 threads)	379s	267s	1.42x
AMD 3995WX (128 threads)	351s	276s	1.27x
Intel Xeon 8280 x 2 (112 threads)	356s	337s	1.06x
Intel i9-9980XE (36 threads)	748s	720s	1.04x



For a texture heavy stress scene (8GB of tiff based .tx texture tiles end up being read in) we get the following speedups on Windows:

CPU	Arnold 6.0.3.0	Arnold 6.0.4.0	Speedup
AMD 3970X (64 threads)	2002s	9.9s	202x
AMD 3990X (128 threads)	2103s	10.1s	208x
AMD 3995WX (128 threads)	2130s	9.0s	237x
Intel Xeon 8280 x 2 (112 threads)	1585s	11.2s	142x
Intel i9-9980XE (36 threads)	1145s	16.8s	68x

- **String parameter path metadata:** A new `path` metadata can be assigned to string parameters to give specific hints about how they will be handled. Currently available values are: `file`, `folder` and `folderlist`. Arnold will expand environment variables, apply the remapping, and handle backslashes on string parameters with the metadata. (#4969)
- **AA\_adaptive\_threshold in log:** The `options.AA_adaptive_threshold` used is now output in the log file. (#9841)
- **Upgraded OSL:** Arnold now uses OSL 1.11.6 bringing multiple optimizations and bug fixes, see the [full list](#) of changes. (#8917)

## GPU Enhancements

- **Global AOV shaders:** The `aov_shader` option is now supported on GPU. This option defines a list of shaders that the renderer will evaluate after the regular surface shader. With this, it's possible to add shaders to set specific AOVs without modifying the original shader tree. See the [option documentation](#). (#9394)
- **Initial support for attributes in OSL:** You can now retrieve user data bound to shapes via the `getAttribute` function or via a `lockgeom` parameter attribute. This initial implementation still has some limitations when compared to the CPU implementation. Queryable attributes are currently restricted to those bound to the current shaded shape and attributes of the active render camera. Querying node parameters is also not yet supported. (#9666)
- **Initial support for tracing in OSL:** You can now use the `trace` function to trace rays in OSL shaders along with the `getMessage` function to retrieve information regarding any intersected object. This initial implementation has the restriction that you cannot yet query parameters on an intersected object. (#9797)

## Bug fixes

- `maxtoa#667`: skip exporting void uvw channels (was firing annoying warnings about invalid user data)
- `maxtoa#750`: use Existing `.tx` doesn't apply to OSL maps
- `maxtoa#767`: The legacy map warning must also warn about GPU
- `maxtoa#769`: legacy maps can't access the face number id
- `#9828` `AiMetaDataGet/Set` functions crash with a NULL node entry

- #9746 [Alembic] sampled faceset data causes crash
- #9816 [Alembic] Velocities data on curves not applied correctly
- #9515 Allow IPR to properly scale past 64 cores on Windows
- #9809 Allow maketx to use either single dash or double dashes for arguments
- #9684 Arnold ignores non-linear output color spaces on images wider than 8 bit
- #9020 Back slashes in strings get changed to forward slashes when queried or written to .ass
- #9672 Better duplicate output detection
- #9749 BVH with 3+ motion keys is missing AiProfileBlock
- #9575 Crash when calling AiMsg while the Arnold session is ending
- #9725 [GPU] Arnold prematurely runs out of memory with on demand textures
- #9748 [GPU] Crash in GPU prepare when failing to compile programs
- #9794 [GPU] fov not set when using custom perspective camera
- #9852 [GPU] GPU crash with multiple shaders connected to the UV camera post\_bake
- #9580 [GPU] Incorrect "Insufficient device memory" error when using NVLink
- #9692 [GPU] Incorrect result when calling user data shaders from uv\_camera.post\_bake
- #9771 [GPU] OptiX denoiser incorrectly accumulates feature buffers in progressive mode with multiple filters
- #9774 [GPU] OSL: Don't throw exceptions when getattribute is called
- #9636 [GPU] OSL texture crash when toggling devices
- #9202 [GPU] Precision issue in bump mapping
- #9532 [GPU] Random crash during the GPU prepare stage
- #9801 [GPU] "Unable to load Optix library" error on Tesla and TCC GPUs on Windows
- #8956 kick logs show superfluous plugin loading messages before official log start
- #9758 Make smart opaque robust to numerical precision error
- #7934 mesh\_light with degenerate triangles has poor performance
- #9522 [OSL] Arnold can fail to find stdosl.h when the path to Arnold contains specific characters on Windows
- #9840 [OSL] Crash when SIP is enabled on MacOS 10.14 when using OSL
- #9607 Parsing errors in ass file parameters should abort scene loading
- #9659 Potential memory leak when using \_triggers\_reinitialize metadata
- #9753 Prevent texture cache from being smaller than 2GB
- #9863 Print license error details at warning level
- #9845 Python binding for AiParamGetType returns c\_int instead of c\_uint8
- #9762 Slow build with many ginstances of many-prim objects
- #9595 Slow node update of VDB volumes on Windows and Linux even when volume did not change
- #9610 Spurious render message when using -default\_nodes in kick
- #9644 Using AOV write shaders increases texture usage
- #9638 Wrong exterior illumination with portal lights in interior\_exterior mode
- [usd#463](#) Texture coordinates of texcoord2f type are not read correctly