Table of Contents

1 Introduction

Air Stream is a Maya plug-in for rendering with SiTex Graphics' Air rendering software. In addition to this plug-in, you must also have a registered or free version of Air installed.

Once Air Stream and Air are installed, you can render with Air just like any other Maya renderer. See the Rendering section for specific instructions on basic rendering. For non-trivial scenes, the Optimizing Scene Export section describes how to optimize export and rendering performance using archives.

This manual describes the Air Stream interface and should provide enough information to start rendering with Air. For more information about specific Air features, please see the Air User Manual.

We welcome feedback from anyone using Air Stream. Please send bug reports, support requests, and feature ideas to:

SiTex Graphics Support support@sitexgraphics.com

Quick Links:

2 What's New

(For a list of earlier changes, see the History section.)

Release 3:

- Outlines:
  - New section in the Outlines tab for the new Stroke shaders in Air 13.18.
  - New outline option to set a max edge length for geometry-based edges.

- Air Materials
  - New menu item to load an Air material file created with the new standalone Air Material Editor (Windows only)

- Textures:
  - New option to automatically convert images to Air texture maps for more efficient rendering. The control can be found in the Textures section of the Shading tab in the render globals, and it is enabled by default. Note that using automatic texture conversion and the option to convert from sRGB color space requires Air 13.02 or later to render correctly.
  - New option in the Textures section of the Shading tab to enable automatic mipmapping for images not converted to Air textures (requires Air 13.02 or later)
  - New option in the Texture section of the Shading tab to control how textures are resized during conversion to Air texture maps.
  - Increased default texture cache size to 32mb (from 20mb). In Air 13.01 and later, there is a single cache shared by all threads (vs a cache per thread in earlier Air releases).

- Added builds for Maya 2015 for Linux and Windows. Dropped builds for Maya 2010.

- Shading:
  - Added File and FileOffset parameters to the FileTexture shader
  - MirrorU and MirrorV parameters of the Place2dTexture node are now supported, and automatic texture conversion chooses an appropriate corresponding wrap mode.
  - Place3dTexture InheritTransform attribute is now supported.
  - In Perspective mode a Projection node with linked camera now respects the camera aperture and fit/fill settings.
  - In nodes with Color Balance attributes, alpha is no longer clamped to the unit interval after the application of alpha gain and offset

- Baking:
  - Valid ranges for the following bake parameters have been increased: bake map size, bake shading samples, and bake pixel samples.
  - A new baked map file type of "Compressed TIFF" saves maps as TIFF files compressed with LZW.
  - New option to specify the command to use for baking maps in the Render Control section of the Render tab.
  - New shaders for AddDoubleLinear and MultDoubleLinear nodes

- Miscellaneous:
  - New option to choose how surface tolerance is measured (in pixels or object space) in the Geometry section of the Render tab.

- Fixes:
  - Fix for volume rendering to work with autosize
  - Fix for translation of connected color parameters of Air light shaders
  - Fixed export bug that prevented bake quality settings from being applied to the baked maps

3 Installation

Linux Installation

- Unzip the contents of the Air Stream archive from the parent directory where you would like Air Stream to be installed. E.g.,

  ```
  cd /usr/local
  unzip  airstream_xxx.zip
  ```

  would place the Air Stream files in

  ```
  /usr/local/airstream
  ```

  which will be referred to as the installation directory.

- Set the environment variable AIRSTREAM_HOME to the installation directory:

  ```
  export AIRSTREAM_HOME=/usr/local/airstream
  ```

  Append this line to to your `$HOME/.bashrc` file if needed.

- Include the `scripts` subdirectory of the Air Stream installation directory in the MAYA_SCRIPT_PATH environment variable:

  ```
  export MAYA_SCRIPT_PATH=/usr/local/airstream/scripts
  ```

- Include the `plug-ins` subdirectory of the Air Stream directory in the MAYA_PLUG_IN_PATH environment variable:

  ```
  export MAYA_PLUG_IN_PATH=/usr/local/airstream/plug-ins
  ```

- Restart Maya.

Windows Installation

- Unzip the contents of the Air Stream archive to the parent directory where you would like to install Air Stream. E.g., if you unzip the archive to

C:\

the Air Stream files will be found in

C:\airstream

which will be referred to as the installation directory.

- Create an environment variable named AIRSTREAM_HOME with the full path to the installation directory:

```
AIRSTREAM_HOME=C:\airstream
```

- Append the installation directory to the MAYA_MODULE_PATH environment variable. If the environment variable is not currently set, set it to the installation directory:

```
MAYA_MODULE_PATH=C:\airstream
```

- Add the `scripts` sub-directory of the Air Stream installation directory to the MAYA_SCRIPT_PATH environment variable:

```
MAYA_SCRIPT_PATH=C:\airstream\scripts
```

- Add the `plug-ins` sub-directory of the Air Stream installation directory to the MAYA_PLUG_IN_PATH environment variable:

```
MAYA_PLUG_IN_PATH=C:\airstream\plug-ins
```

- Restart Maya.

Network Installation

Machines on a network do not in general need to have access to the Air Stream plug-in to render scenes created by Air Stream. However, to render scenes with Maya shaders, network machines will need to be able to find the shaders distributed in the shaders directory of the Air Stream installation. The RIB files produced by Air Stream include

```
$AIRSTREAM_HOME/shaders
```

in the search path for shaders. You can provide access to those shaders by installing Air Stream in a shared location and creating an AIRSTREAM_HOME environment variable with the Air Stream installation location for each remote rendering client.

Python Path

Some Air Stream modules are written in Python. On Windows, be sure the PYTHONPATH environment variable points to the correct Python installation for the version of Maya you are running.

Maya Environment Variable File

If you use the Maya environment file (Maya.env) to set the Maya environment variables, the MAYA_MODULE_PATH variable should appear before MAYA_SCRIPT_PATH or the plug-in may fail to load.

Air Stream Shelf

Air Stream can create a Maya shelf with buttons for common shaders and commands. Just selecte

Create Shelf from the Air Stream menu.

4 Rendering

Loading the Air Stream plug-in

Note: If you have not yet installed the Air Stream files, complete the Installation instructions first.

- Start Maya.
- Open the Plug-in Manager (in the Window menu, go to Settings/Preferences and choose Plug-in Manager)
- Click the Browse button. In the file dialog, navigate to the plug-ins directory of your Air Stream installation and choose the appropriate plug-in for your version of Maya.
- The Air Stream plug-in should load, adding a new Air Stream menu to the Maya menu bar.

Rendering with Air as the Maya Renderer

Once loaded, the Air Stream plug-in can be used to render your scene just like any other rendering plug-in:

- In the Maya Render Globals dialog, choose Air Stream as the renderer.
- Click on Maya's Render button.

When a rendering is started with the Maya Render button, Air Stream exports the scene data and starts Air as an external process. When the rendering is complete, the rendered image is loaded and displayed in the Maya render view.

Update: under Windows rendering to the Maya preview window now sends data directly to Maya over a socket connection on port 51155. Make sure any security software that is installed allows communication between Air and Maya over that port.

Rendering from the Air Stream Menu

The Air Stream menu offers several ways to render with Air, whether or not Air is not selected as the current renderer in Maya:

Render - Preview

The Render Preview option exports the scene data and starts an external process that renders a preview image to the stand-alone Air Show framebuffer. Unlike the Maya render, this preview method allows you to continue working in Maya while the preview rendering runs as a separate process.

If additional render passes have been created (on the Images tab of the Render Globals) and the option to include extra passes in a preview render is selected, the additional passes will appear as extra layers in the preview image.

Render - Current Frame

The Render Current Frame option exports the current frame and starts an external process that renders the specified output images to the standalone Air Show framebuffer. You may continue to work in Maya while the frame is rendered as a separate process.

The primary image and secondary render passes are all rendered according to their parameters, except that output is redirected to Air Show instead of the specified display driver.

Render - Sequence

The Render Sequence option first exports scene data for every frame in the current animation range, usually one rib file for each frame. An external process is then started to render all frames. On Windows the external process runs at low priority.

The primary image and secondary render passes are all rendered according to their parameters and saved using the specified display drivers.

Distributed Rendering over Multiple Machines (Windows only)

The Vortex tool included with Air allows simple distributed rendering of single frames and sequences over a network of machines. To use Vortex for distributed rendering, set the render command to `vortex` on the Render Tab of the Air Stream globals dialog.

Submitting Jobs to the Air Control Render Queue (Windows only)

The Windows distribution of Air includes a simple render queue manager named Air Control. To submit jobs to the Air Control render queue, set the render command to `airq` on the Render Tab of the Air Stream globals dialog.

See Also:

Optimizing Scene Export
Air User Manual: Tools -> Air Control
Air User Manual: Tools -> Vortex

5 Image Quality

High quality images require the elimination of aliasing - artifacts in an image due to undersampling some aspect of the scene. Air provides extensive support for fine-tuning the sampling in a scene.

Quality Overview

The Render tab in the Render Settings dialog sets the default values for the major quality control parameters:

Shading Samples

This value determines how often an object's shading is evaluated per pixel. E.g., a value of 4 would result in 4 shading samples per pixel.

Pixel Samples

This setting determines how often the scene's geometry is sampled, defined as an N x N grid of

samples.

Filter

The filter is applied to the sub-pixel samples to produce the final output values.

Surface Tolerance *(under Geometry)*

Air automatically tessellates NURBs, subdivision surfaces, and curves at render time based on the tolerance parameter, which gives the maximum allowed distance in pixels between the tessellated mesh and the true surface or curve. Higher values produce a coarser tessellation, which will use less memory and render more quickly.

Air provides independent control over the sampling of geometry and shading. This is an important capability: for many scenes rendering time is dominated by shading calculations. With Air you can increase the sampling of a scene's geometry to smooth rough edges without increasing the number of shading calculations that are performed.

The following sections describe how to eliminate common sources of aliasing in rendered images.

Jagged Edges

Increasing the number of pixel samples will help to smooth jagged edges.

Air also has a special sub-pixel edge mask option that can help capture the details of small or thin objects such as tiny particles or fine hair. Check **Smooth edges** to enable the sub-pixel edge mask.

Shading Aliasing

Shading can introduce aliasing from a pattern, reflections, specular highlights, small bumps, shadow edges, and other sources. The simplest way to improve shading quality is to increase the number of shading samples.

Increasing the number of shading samples globally can significantly slow down rendering: rendering time is often directly proportional to the number of shading samples, so doubling the number of shading samples can also double rendering time. Instead, it is more efficient to increase the number of shading samples only for those objects that require more detail.

For Maya shaders the number of shading samples can be increased using the **Shading Samples** override control in the Quality section of an assigned Air Stream node.

For an Air surface shader, the number of shading samples can be adjusted in the Advanced Attributes section of the Air Stream Surface Shader.

Shadows

Ray traced shadows can exhibit shading artifacts in the form of jagged edges or, for blurry shadows, excessive noise. In both cases increasing the number of shadow rays on the will improve shadow quality.

For Maya lights the number of shadow rays is taken from the **Shadow Rays** attribute in the Raytrace Shadow Attributes section of the light's shape node.

Most custom Air light shaders have a **Shadow Samples** parameter that controls the number of

shadow rays.

Reflections

Ray-traced reflections may exhibit artifacts in the form of jagged edges or excessive noise. As with shadows, increasing the number of rays will improve image quality.

For Maya materials the default number of reflection and refraction rays is set in the Ray Tracing section of the Shading tab. You can also use the Mental Ray reflection and refraction blur controls to tune the number of rays on a per-material basis.

Most Air surface shaders that offer reflections have a **Reflection Samples** parameter that specifies the number of reflection rays to trace.

Motion Blur

The smoothness of motion blur is determined by the number of pixel samples. For heavily blurred scenes you may need to increase the number of pixel samples to 8 x 8 or higher.

Objects that are heavily motion-blurred can often be rendered with a reduced surface tolerance. In the Motion Blur section of the Render tab, enable **Reduce quality for blurred objects** to have Air automatically increase the tolerance based on the distance an object is blurred. The **Quality Reduction Factor** can be used to control how aggressively this optimization is applied.

Summary

- A reasonable goal is to be able to render a final quality image with 1-4 shading samples per pixel.
- For a soft filter, try a gaussian filter with a width of 2. For a sharper filter, use a mitchell filter with a width of 4.
- To smooth noisy shadows for a light, increase the number of shadow rays cast by the light.
- To smooth reflections, increase the number of reflection rays for the reflective material.
- For an individual material that exhibits artifacts, increase the number of shading samples only for that material.

See Also:

Air User Manual: Image Quality

6 Optimizing Scene Export

To render a scene with Air, Air Stream first exports all the relevant scene data to one or more files in the RIB format.

By default all scene data is written to a single file, and a separate file is created for every frame. For non-trivial scenes this simple export method can take a significant amount of time and produce a lot of redundant data.

Optimizing Scene Export with Archives

Air Stream provides an option to optimize the export of scene data by storing the geometry for each object in a separate archive file and referencing those archive files in the main RIB file. As long as the object geometry data is not changed, the same archive files can be used for subsequent renderings.

The default archive mode for all objects is set with the **Archive Mode** control on the Export tab of the Air Stream render globals. To enable archiving for a new scene, set the **Archive Mode** to `Write`. The next time the scene is rendered or exported, Air Stream will create a separate archive file for each object. The archive name is based on the DAG path for the object in Maya. The archives will be created in the archive directory specified on the Export page in the render globals.

Once the archives have been created, set the **Archive Mode** to `Read` to re-use the archives on subsequent renders. For a new scene with an empty archive directory, you can simply set the **Archive Mode** to `Read` initially (instead of `Write`), and Air Stream will automatically create the missing archives.

Loading Archives On-Demand

To optimize memory and render time for large scenes, Air allows archives to be loaded as they are needed during a rendering. To enable this delayed reading of archives, set the **Archive Read Method** on the Export tab to `'Read when object bound is hit'`. In delayed read mode, each archive is referenced using a bounding box and an associated file name. If and when the renderer encounters the bounding box, the associated archive file is read into memory. Objects that are invisible - either off-screen or obscured by other geometry - will not be loaded at all.

This method allows Air to render scenes with high geometric complexity in a relatively small memory footprint as long as objects do not need to be retained for ray-tracing. Air will discard each object as soon as it has been rasterized, freeing memory for later use as new objects are loaded on-demand. By using maps or point clouds for effects such as shadows, reflections, and global illumination, scenes with huge geometric data sets can be rendered with modest memory consumption.

Note that by default all objects are retained for possible intersection with shadow rays and reflection rays. You can change this global default behavior in the Ray Trace section of the Shading tab by unchecking **Enable shadows** and **Enable reflections**.

Per-Frame Archives

For a static object in an animated sequence, only one copy of the geometry data needs to be exported, and the same archive can be referenced for every frame. For an object whose vertex positions change over time, a separate archive must be exported for each frame. By default Air Stream assumes all objects except particles are static. To tell Air Stream to create an archive per-frame:

- Select the object. From the Air Stream menu, choose Air Stream Node -> Create New Node.
- Select the Air Stream node and expand the Export section.
- Set the **Archive Frequency** control to `Per Frame`.

Updating Archives

The archive file for an object contains the following information:

- Vertex positions, normals, and texture coordinates

- Motion blur data
- Additional vertex data for particles
- User-assigned primitive variable data

An archive needs to be re-written whenever any of the above information has been modified. You can force re-writing of all archive files in a scene by setting the **Archive Mode** on the Export tab to `Write`.

If only a few objects have been modified, it can be faster to use an Air Stream node to enable archive updating for only the modified objects:

- Select the modified objects and assign an Air Stream node (if one has not already been assigned).
- In the RIB Export section, set the **Archive Mode** to `Update archive`.
- Export or render to update the archives. For a static object, only the current frame needs to be exported or rendered. For an object with per-frame archives, the whole sequence must be exported or rendered.
- Set the **Archive Mode** back to its previous setting (normally `Inherit`).

Custom File Names for Archives

By default Air Stream automatically generates the file name used to write and to read an archive. You can instead explicitly specify the name for an archive, allowing you to "bake" geometry for re-use in another scene or application.

To export geometry to a specific file:

- Select the geometry and assign an Air Stream node.
- In the RIB Export section of the Air Stream node, set the **Custom Archive Name** field to the desired file name. To generate a separate file for each frame, include the frame number in the file name using $F*n*, where *n* is the number of digits for the frame number.
- Set the **Archive Mode** to `Update archive`. To save a separate archive per frame, set **Archive Frequency** to Per frame.
- Export or render the scene to generate the archive file(s).

To use a pre-written archive file or sequence:

- Select the geometry and assign an Air Stream node.
- In the RIB Export section of the Air Stream node, set the **Custom Archive Name** field to the file name. To reference a separate file for each frame, include the frame number in the file name using $Fn, where n is the number of digits in the frame number.
- Set the **Archive Mode** to `Read only`. To read a separate archive for each frame, set **Archive Frequency** to `Per frame`.
- Air Stream will substitute the archive or archive sequence for the object geometry when exporting the scene.

Creating Asset Archives with Multiple Shaded Objects

Air Stream's automatic archiving capability writes only the geometric shape data for each object. For more complex archives, the Air Stream export command provides an option to write all currently selected objects to a single archive, including the shaders and attributes for each object. This option is also available from the Air Stream export menu.

The exported asset archive includes all the shading and geometry information required to render the included objects. Such an archive can be included in another scene using a standin object with an

Air Stream node assigned. Simply set the **Archive mode** to `Read only` and provide the asset archive name as the **Custom Archive Name** (in the Export section of the Air Stream node).

See Also:

Export Tab
Air Stream Node

7 Output Images and Passes

An Air rendering process produces one or more output images. Use the Images tab of the Render Globals dialog to select the output values to save for the current rendering task.

Passes

Most Air shaders and translated Maya shaders provide a standard set of output values.

Pass	Description
__ambient	surface illuminated by ambient light only
__constant	basic unlit surface color
__diffuse	surface illuminated by diffuse light only
__diffuse_unshadowed	direct diffuse result without shadows
__diffuse_shadow	difference between direct diffuse lighting without and with shadows
__shadow	shadow fraction computed from __diffuse and __diffuse_unshadowed
__indirect	indirect diffuse light from the global illumination light source
__reflect	reflection component of the shading result
__refract	refraction component of the shading result
__environment	reflected environment contribution
__background	imager shader contribution
__occlusion	ambient occlusion
__light[*N*]	per-light output channel

The following global values can also be saved:

z	camera depth value
N	shading normal vector
P	shading position in camera space

Each output can be saved to a separate file or included as a layer in a multilayer image (for image formats that support multiple layers).

Groups

By defaut all visible objects in the scene will appear in each output image. Air's grouping mechanism can be used to restrict the objects that appear in an image to the members of a set of groups, specified in the Groups parameter of the render pass window. Objects that are not in the list of groups for the pass will be rendered as matte objects.

Global group membership can be set on the global Render tab, in the *Groups* section. Group membership for individual objects can be customized by assigning an Air Stream Node and using the controls in its *Groups and Sets* section.

Light Channels

Air allows the contributions of individual lights to be recorded separately. Light channels can be manipulated in a composite to effectively relight a scene in comp without having to re-render. To specify the output channel for a light:

- Select the light.
- From the Air Stream menu, choose Light Node -> Create New Light Node.
- Select the new Air Stream light node in the Attribute Editor.
- Set the Light Channel attribute to the desired channel number.

Multiple lights may be saved in a single channel, in which case the light contributions are simply summed.

Note that the contribution of the global illumination light source (if present) will be saved in channel 0.

The Light Channels section of the Images tab has two controls that affect the behavior of light channels:

Reflect light channels

Air surface shaders will optionally include the effects of reflection and refraction when the **Reflected light channels** option is enabled. When this option is enabled, the Reflect output pass should not be used. You may wish to save the Environment pass which holds the contribution of the reflected environment to the scene.

Bounce light channels

Air will optionally include the effects of indirect light bounces in per-light channels if the **Bounce light channels** option is checked. When this option is enabled, the indirect environment contribution will be stored in light channel 0.

When both of the above options are enabled, the light channel will include the total contribution of the light to the output image.

Occlusion

When an occlusion pass is requested, Air Stream automatically adds an additional generic shader to the surface shader stack for each model. The additional shader provides a __occlusion output variable with the current occlusion value. The occlusion quality settings are taken from the Global Illumination render tab.

See Also:

Images Tab
Air Stream Light Node

Air User Manual: Output

8 Shaders

Air Stream allows you to use Maya shaders and Air shaders and even combine Maya and Air shaders in a single shading network.

Air Stream uses Air's network shader capability to provide flexible and efficient support for rendering with Maya and Air shaders:

- A Maya shading network assigned as a surface or displacement shader will automatically be exported as a network of Air shaders.
- Air custom surface or displacement shaders may be assigned as the surface or displacement shader for a material.
- Air shader parameters can be connected to Maya shading nodes.
- Air generic shaders can be used in a Maya shading network.
- Air environment shaders and imager shaders can be driven by a Maya shading network
- Air Stream's shader library can easily be extended to support custom Maya shading nodes.

Maya shader conversion relies on the new shader network capability available in Air 11.

Shader Types

Air Stream supports the following Maya and Air shader types:

Surface
Displacement
Light
Imager
Environment
Instancer
Generic
Volume
Stroke

See the documentation page for each shader type for more information on using Air shaders in your Maya scene.

See Also:

Air User Manual: Shader Guide

8.1 Maya Shading Networks

Supported Maya Nodes

The following Maya shading nodes are currently supported:

Surface Materials

Anisotropic, Blinn, Lambert, Layered Shader, Phong, Phong E, Ramp Shader, Surface Shader, Use Background

Ray-traced reflections and refractions are supported, and they are enabled by default. Blurry reflections and refractions can be rendered by using the Mental Ray reflection and refraction blur controls. Air will use the specified number of rays for blurry reflections and refractions. For crisp reflection and refraction, the default number of reflection and refraction rays can be set in the Ray Tracing section of the Shading tab in the Render Settings window.

Translucence is currently implemented as a simple back-facing diffuse illumination component.

2D Textures

Bulge, Checker, Cloth, File Texture, Fractal, Grid, Mountain, Noise, Ramp

3D Textures

Brownian, Cloud, Crater, Granite, Leather, Marble, Rock, Snow, SolidFractal, VolumeNoise, Wood

Displacement Materials

Displacement

The method used to simulate displacement can be specified globally in the Displacement section of the Shading tab in the Render Globals. By default displacement is applied only as a bump effect. If you wish to render with true displacement, you can enable that globally on the Shading tab. To enable true displacement for only the current object, apply an Air Stream node to the object and change the displacement method in the Displacement section of the Air Stream node.

Environment

envChrome, envCube, envSky, envSphere

Utilities

AddDoubleLinear, BlendColors, Bump2d, Bump3d, ClampColor, Condition, Contrast, GammaCorrect, HsvToRgb, Luminance, MultDoubleLinear, MultiplyDivide, ParticleSamplerInfo, Place2dTexture, Place3dTexture, Projection, Reverse, RgbToHsv, SamplerInfo, Stencil

Other Textures

LayeredTexture

Mental Ray

mia_material, misss_fast_shader*, mi_car_paint*, mib_illum_phong, mib_illum_lambert, mib_illum_cooktorr, mib_illum_ward_deriv, dgs_material, dielectric_material

The misss* shaders use Air's built-in subsurface scattering capability, which requires that an object belong to a subsurface scattering group. To do this, assign an Air Stream node to the object, and assign a point set name in the Groups and Sets section.

Adding New Maya Nodes

The Air Stream translator for Maya shading networks will attempt to translate any node for which there is a corresponding Air shader in the `$AIRSTREAM_HOME/shaders` directory. You can add support for additional shading nodes by writing an Air generic shader that implements the behavior of the corresponding Maya node.

Here's how the translator works: for each Maya node in a shading network, Air Stream forms a shader name by prepending the string ASM_ to the node type name. Air Stream then looks for a corresponding Air shader in the shaders directory of the Air Stream installation. E.g., for a Phong node, Air Stream attempts to load:

```
$AIRSTREAM_HOME/shaders/ASM_Phong.slb
```

If the target Air shader loads successfully, Air Stream then goes through the list of shader parameters looking for equivalent Maya attributes. If a match is found, Air Stream takes the shader parameter value from the corresponding Maya attribute, or forms a connection to another node if the attribute is connected.

Each Air shader parameter should take its name from the corresponding Maya attribute - with the first letter capitalized.

Example: Here's the source code for Maya's Reverse node:

```
generic ASM_Reverse(
  color Input = 0;
  output color Output = 0;
)
{
  Output = 1 - Input;
}
```

8.2 Surface Shaders

Air Stream supports Maya surface shaders and Air surface shaders. Maya surface shader nodes are automatically translated for rendering with Air when the scene is exported.

Assigning an Air Surface Shader

You can apply an Air surface shader to an object using an Air Stream surface shader node, which can be assigned to an object just like any other Maya surface shader:

- Select the desired object.
- From the Lighting/Shading menu, select Assign New Material, then pick the Air Stream Surface Shader.

In the Attribute Editor, click the button next to the **Shader** text box to choose an Air surface shader. The shader selection dialog will list all Air surface shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

The top section of the surface shader node displays a small preview image of the surface shader. When the **Preview** box is checked, the preview image will be automatically updated whenever a shader parameter value is changed. During IPR rendering you may wish to disable shader preview to accelerate redrawing of the IPR view.

Surface Shaders and Shadows

To speed the computation of ray-traced shadows, by default Air does not evaluate the surface shader assigned to an object when it is intersected by a shadow ray. This works well as long as the object's transparency is based solely on the assigned opacity attribute (which Air Stream derives from the transparency value in Maya). For surface shaders or shader networks that compute a complex transparency value, a special attribute must be enabled to tell Air to evaluate the surface

shader network for shadow rays.

For Maya surface shaders, Air Stream will automatically force shader evaluation for shadow rays if the Maya transparency attribute is connected to another node. For Air Surface shaders, check **Surface modifies transparency** under Basic Attributes to enable shader evaluation for shadow rays.

Tips:

- The Air Stream shelf has pre-configured buttons for assigning common Air surface shaders to the currently selected objects.

See Also:

Air User Manual: Shader Guide -> Surfaces
Air Stream Surface Shader

8.3 Displacement Shaders

Air Stream supports Maya displacement shader networks and Air displacement shaders. Maya displacement shader networks are automatically translated for rendering with Air when the scene is exported.

Assigning an Air Displacement Shader

Use an Air Stream Displacement Shader node to assign an Air displacement shader:

- Select the surface shader associated with the target object.
- Click on the companion shading group tab.
- Click on the Displacement navigation button.
- In the Materials tab, choose Air Stream Displacement from the Displacement Materials list.

Click the button next to the **Shader** text box to display a file browser for choosing an Air displacement shader. The shader selection dialog lists all Air displacement shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

The top section of the displacement shader node displays a small preview image of a sphere with the displacement shader. When the **Preview** box is checked, the preview image will be automatically updated whenever a shader parameter values is changed. During IPR rendering you may wish to disable shader preview to accelerate redrawing of the IPR view.

Displacement Modes

Air Stream allows you to choose from three methods of evaluating a displacement shader:

Bump only

Displacement is simulated by altering the shading normal, but the surface itself is not moved. This is the fastest and most memory efficient method.

True displacement

The displacement shader moves the surface. Each surface is broken into individual polygons, and

each polygon is displaced separately. This method is usually faster than the mesh displacement mode described below, but it can use more memory and exhibit cracks in some cases.

Mesh displacement

The displacement shader moves the surface. The entire surface is displaced at once to prevent cracks and optimize the storage for the final displaced mesh.

The default displacement mode is set in the Displacement section of the Shading tab. You can override the displacement mode on a per-object basis using an Air Stream node.

Displacement Bounds

In order to correctly render displaced surfaces, Air needs to know the maximum distance that a displacement shader might move the surface. This displacement bound is used to expand the bounding box for the object to make sure that all parts of the displaced surface will be visible in the final rendering.

Air Stream will compute a default displacement bound for Maya and Air displacement shaders as follows: For a Maya Displacement Shader node, the Scale parameter is used for the displacement bound. For an Air displacement shader, the displacement bound is taken from the Bump Max parameter value. For other cases, the shader will inherit the default displacement bound set in the Displacement section of the Shading Tab.

The computed displacement bound can be overridden by assigning an Air Stream node to an object and using the controls in the Displacement section of the Air Stream node.

Displacement Quality

For true displacement or mesh displacement, Air tessellates the surface to form a fine mesh of polygons to represent the detail of the displacement pattern. The quality of the mesh tessellation is based on the surface tolerance setting, which gives a view-dependent distance in pixels. The default value of 0.5 produces polygons that are about 1 pixel in size. A larger tolerance produces a coarser mesh with larger polygons that will render faster and use less memory.

The default tolerance value is set in the Image Quality section of the Render tab. Tolerance can be set on a per-object basis using the controls in the Quality section of an Air Stream node.

See Also:

Air Stream Displacement Shader
Air User Manual: Shader Guides -> Displacement

8.4 Imager Shaders

Imager shaders can be used to modify the final rendered image prior to output.

Adding an Air Imager Shader

To add an imager shader to a scene:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Imager Shaders section.

- Click Create Imager to create a new imager shader node. A shader selection dialog will apear. The shader selection dialog lists all Air imager shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Air Stream allows multiple imager shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

Using Maya Nodes with an Imager Shader

Imager shader parameters can be driven by a Maya shading network. Texture coordinates are defined for an imager shader such that the unit square maps to the output image, allowing standard 2D texture nodes to be used.

For example, here's how a ramp texture could be used to color the image background:

- Add a new imager shader as described above, choosing the `background` shader from the list of Air imager shaders.
- Click the connection button next to the **background** parameter, and select a ramp texture.
- Render to see the ramp texture used as a backdrop for the rendered image.

See Also:

Air Stream Imager Shader
Air User Manual: Shader Guide -> Imagers

8.5 Environment Shaders

An Air environment shader is used to shade rays that miss all objects in a scene.

Adding an Environment Shader

To create a new environment shader:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Environment Shaders section.
- Click Create Environment to create a new environment shader node.
- The shader selection dialog appears with a list of all Air environment shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. Choose an Air environment shader from the list or pick `[Select File]` to load a compiled shader file from a different location.

Air Stream allows multiple environment shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

Environment Shader Visibility

Environment shaders may be evaulated by any of the following ray types:

- When a reflection or refraction ray exits the scene.
- When an indirect ray cast to compute indirect global illumination exits the scene.
- When a pixel in the final rendered image is not fully opaque (i.e., the "camera" ray exits the scene)

You can choose which ray types a given environment shader affects by setting the corresponding shader visibility parameter:

- To make an environment shader invisible to reflection/refraction rays, set **Reflection Visibility** to 0.
- To make an environment shader invisible to indirect rays (and thereby not contribute to global illumination), set **Indirect Visibility** to 0.
- To exclude a shader from the final image background, set **Camera Visibility** to 0.

By setting the visibility parameters appropriately a different shader could be used for each ray type.

Using Maya Nodes with an Environment Shader

Maya shading nodes can be used to drive an Air environment shader. The simple envColor shader can be used as a base for building a Maya shading network. Here's how to set up a Maya network to use an environment map for the scene environment:

- Create a new environment shader as described above, selecting the envColor environment from the list of shaders.
- Click the connection icon next to the **Color** parameter and choose envSphere from the list of Maya nodes.
- Connect the **Image** property of the envSphere node to a file texture node.
- Choose a lat-long environment map for the file texture node.
- Render to see the final image.

You can use the normal positioning controls for the envSphere and file texture nodes to position the environment map. In IPR mode the IPR view will automatically update as you make changes.

Tips

- In IPR mode, the environment contribution to global illumination will interactively update to reflect changes to environment shader parameters if the global illumination mode is occlusion and the occlusion environment sampling method is approximate.

See Also:

Air Stream Environment Shader
Air User Manual: Shader Guide -> Environments

8.6 Instancer Shaders

Air's instancer shaders provide a powerful mechanism for adding new objects to a scene based on an existing object. An instancer shader can query almost any aspect of the base object, and use that information to generate new geometry in any manner desired.

Sample uses for instancer shaders include:

- Air Stream uses instancer shaders to implement the mult-streak and mult-point particle rendering modes.
- The `MassiveAgents` instancer shader can be used to add Massive crowds to a scene.
- The `VInstanceArchive` instancer can be used to instance geometry using a particle system, with per-particle attributes driving unique shading for each instance.

- The `instMultipleArchives` instancer can be used to instance multiple archives with a particle system.

Assigning an Instancer Shader

An Air Stream instancer shader can be assigned to an object using the Air Stream menu:

- Select the desired object.
- From the Air Stream menu, choose Instancer Shader -> Create New Instancer Shader.

Click the button next to the **Shader** text box to choose an Air instancer shader. The shader selection dialog lists all Air instancer shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Instancer Bounds

To correctly process an instancer shader Air needs to know the region in space that will be occupied by the new objects. The bound is given as a distance used to expand the bounding box for the base object to generate a bounding box for the instancer. Use the **Region bound** attribute of the Air Stream Instancer node to define an appropriate distance.

See Also:

Air Stream Instancer Shader
Air User Manual: Shader Guide -> Instancers

8.7 Generic Shaders

Air's generic shader is a general shader type for components that will be used in shading networks. Generic shaders are compatible with all other shader types.

Adding a Generic Shader

To add an Air generic shader to a network:

- Open the Hypershade window in Maya.
- Drag the shading network you'd like to work on to the Work Area.
- Under the Create tab, find the General Utilities section and click on the Airstream Generic Shader entry to add that node.
- In the attribute editor for the Airstream Generic Shader node, click on the file browser button next to the **Shader** name to choose an Air generic shader. The shader selection dialog lists all Air generic shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.
- Once the Air generic shader is loaded, you can modify its input parameter, and connect its inputs or outputs to other nodes in the shading network.

See Also:

Air Stream Generic Shader
Air User Manual: Shader Guide -> Generics

8.8 Volume Shaders

Volume shaders can be used to simulate atmospheric effects such as smoke and fog.

Adding an Air Volume Shader as an Atmosphere

To add a volume shader as the atmosphere shader for all objects in the scene:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Atmosphere Shaders section.
- Click Create Atmosphere to create a new volume shader node.
- In the Air Stream Atmosphere Shader window, click on the browse button next to the **Shader** text box to choose an Air volume shader. The shader selection dialog lists all Air volume shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Air Stream allows multiple volume shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

Note that Atmosphere shaders are evaluated when an object is shaded (after the surface shader is run). Atmosphere shaders thus only affect the appearance of objects in the scene. Uncovered pixels in the output image will not exhibit atmospheric effects.

See Also:

Air User Manual: Shader Guide -> Volumes

8.9 Stroke Shaders

Stroke shaders can be used to customize line drawing for Air's outline ability. Note that a stroke shader only applies to outlines generated with one of the object-based edge detection methods supported by Air.

Adding an Air Stroke Shader

To add a stroke shader:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Outlines tab and expand the Stroke Shaders section.
- Click Create Stroke to create a new stroke shader node.
- Select a stroke shader from the displayed list, or click on Custom to select a shader file.

Air Stream allows multiple stroke shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

See Also:

Air User Manual: Shader Guide -> Strokes

8.10 Primitive Variables

Air allows arbitrary user-defined data to be attached to geometric primitives as primitive variables or "prim vars". When a primitive variable name matches the name of a shader input parameter, the prim

var value at the current shading location is automatically passed to the shader. This behavior allows shader inputs to be controlled or overridden on a per-object basis.

Primitive variables can be attached at different levels of granularity, determined by the variable's class declaration:

Class	Frequency
constant	1 per primitive
uniform	1 per face
varying or vertex	1 per vertex
facevarying	1 per vertex for each face

Maya Attributes as Prim Vars

Maya attributes attached to a shape node can be exported as prim vars for rendering with Air using an Air Stream Prim Var Node:

- Select the shape.
- From the Air Stream menu, choose Primitive Variable -> Create New Prim Var Node.
- Select the Prim Var node in the attribute editor.
- Set **Maya Attribute Name** to the name of the Maya attribute to export.
- Set **RIB Variable Name** to the name of the variable as it will appear in the rib file. If no name is provided, the Maya attribute name will be used.
- Select the variable class, type, and array size.

Color Sets as Prim Vars

Color sets attached to a polygon mesh can be exported as prim vars by attaching an Air Stream node and enabling **Export color sets** in the RIB Export section. Each color set will be exported as a separate variable of type color and class varying.

9 Lighting

Maya Lights

Air Stream translates standard Maya spotlights, point lights, directional lights, area lights, and volume lights for rendering with Air.

Ray-traced shadows, in-memory shadow maps, and disk-based shadow maps are all supported.

Air Light Shaders

Air custom light shaders may be used by attaching an Air Stream Light Node to a Maya light. Air ships with a number of custom light shaders, including:

`texturedarealight`

Use the texturedarealight shader to apply a texture map to an area light.

`portallight`

Assign this shader to an area light fitted to a window or door to add illumination from the global environment to an enclosed area.

`photometric_pointlight`

This shader allows illumination to be defined using physical units or an IES light profile.

`sunlight`

The sunlight shader computes the color and direction for a sun-like light source based on observer position, date & time, and atmospheric conditions.

`uberlight`

Assign the uberlight to a spotlight or distant light source to customize the illumination cast by the light using distance shaping, cross-section shaping, noise, and texture mapping.

See Also:

Shadows
Global Illumination
Air Stream Light Node
Air User Manual: Lighting
Air User Manual: Shader Guide -> Lights

9.1 Shadows

Shadows may be generated using ray-tracing or depth maps.

Ray-Traced Shadows

Air Stream takes ray-traced shadow settings from the properties in the Raytrace Shadow Attributes section of the light shape dialog.

To render shadows using ray tracing, check **Use Ray Trace Shadows**. The **Light Angle** value is converted to a shadow blur amount for Air. The **Shadow Rays** attribute determines the number of rays cast for each shadow query. More rays produce smoother results at the expense of longer rendering times.

For area lights, the shadow rays count is treated as the number of area light samples.

Depth Map Shadows

Air Stream uses the properties in the Depth Map Shadow Attributes section of the light shape dialog to control depth-mapped shadows.

Check **Use Depth Map Shadows** to enabled rendering with map-based shadows.

The map width and height are taken from the **Resolution** setting. Air Stream automatically upscales the resolution to a power of 2. E.g., a resolution setting of 491 would produce a depth map of 512 x 512.

For directional lights, the **Width Focus** value determines the map coverage. Note that **Width Focus** is always used even if Use Auto Focus is checked. The light position of directional lights is also always used, even if **Use Light Position** is unchecked.

If **Disk Based Dmaps** is set to Off, mapped shadows are generated using Air's automap capability, which generates map tiles internally on-demand using ray tracing. Objects must be visible to shadow rays for automap shadows to work properly. For scenes with large automap shadow maps or many such maps, increasing the cache size for automaps in the Texture Memory section of the Shading Tab may improve performance.

To use disk-based maps, set the Disk Based Dmap mode to `Overwrite`. If a separate shadow map is required for each frame, check **Add Frame Ext**; otherwise, a single shadow map will be used for all frames.

Shadow maps must first be generated as a separate rendering task before they can be used. From the Air Stream menu, choose **Render Shadow Maps** to render shadow maps for every light with a disk-based depth map mode of `Overwrite`.

Deep Shadow Maps

Deep shadow maps store additional depth values as well as coverage and transparency information which help produce more accurate shadows. To produce and use deep shadow maps with Air:

- Attach an Air Stream Light Node to the light.
- In the Air Stream Light Node Attributes section set the Shadow Map Depth to a value greater than 1.
- Render shadow maps from the Air Stream menu, then render a preview image.

Larger values for the shadow map depth produce more accurate but larger and slower shadow maps.

Shadows for Custom Light Shaders

Air Stream will automatically use the shadow mode specified in the Maya light controls for a custom light shader if the shader has a `shadowname` parameter and its value is the empty string. This automatic translation allows shadow maps to be easily used with custom light shaders.

Note that only the `shadowname` value is set; other shadow parameters such as bias and samples are not transferred from the Maya settings.

See Also:

Air Stream Light Node

10 Global Illumination

In addition to simulating direct illumination from lights placed in the scene, Air can simulate illumination from other sources, such as the environment surrounding the scene or light that has been reflected from a surface. The Air Stream plug-in groups the controls for these lighting effects under the Global Illumination tab in the Render Settings dialog.

Air offers two methods of computing global illumination:

- Indirect Diffuse Illumination

  In this mode Air computes the indirect diffuse illumination incident at the current shading location by

tracing rays to sample nearby surfaces and the global environment. The illumination computed with this method includes color bleeding effects.

- Ambient Occlusion

  Ambient occlusion is a faster alternative to indirect diffuse illumination. Ambient occlusion is computed by tracing shadow rays to see how much the current shading location is blocked or occluded by nearby surfaces. The occlusion value is then combined with the light contribution from the global environment to produce an illumination result that is generally similar to indirect diffuse illumination but without color bleeding effects.

For more information on these two methods, see the detailed sections that follow.

See Also:

Global Illumination Tab

10.1 Indirect Diffuse Illumination

Air will compute indirect diffuse illumination when the Global Illumination Mode on the Global Illumination tab is set to Indirect.

Air computes indirect diffuse illumination by casting indirect rays from the current shading location. If a ray hits a surface, the surface's shading is evaluated and added to the indirect lighting contribution. For rays that miss all objects, the global environment shaders are evaluted for their contribution.

Sample Quality

The Sample Quality section of the Global Illumination tab has controls for tuning the quality of the indirect diffuse illumination estimate.

The quality of indirect illumination results depends on the number of **Rays per Sample**. More rays produce smoother results. Using too few rays can produce a grainy appearance.

Indirect computations can be accelerated in a couple ways. The **Max trace distance** gives the maximum distance Air will trace a ray looking for intersecting objects. Using a lower distance will reduce rendering time.

When **Adaptive ray casting** is enabled, Air casts fewer rays initially (1/4 the total number) and additional rays only where there are significant differences in the results of the first set of rays.

Irradiance Caching

To accelerate indirect illumination computations, Air supports irradiance caching - a technique in which irradiance samples are stored and re-used at nearby shading locations.

When rendering with an irradiance cache, Air first attempts to estimate the indirect diffuse illumination incident at the current location by re-using values already in the cache. Whether a given cache value can be re-used is determined by the **Max error allowed**. A higher maximum error allows cached values to be re-used more often. If Air cannot find enough usable values in the cache, it will generate a new estimate by casting rays and store the new result in the cache for possible re-use.

To ensure that all areas of the image receive some indirect samples, the **Max sample spacing** setting gives the maximum distance in pixels within which Air will look for usable samples in the

cache.

Indirect Pre-pass

Air has a special pre-pass capability that generally produces smoother final results when rendering with an irradiance cache. When **Uses pre-pass** is enabled, Air first renders a pre-pass evaluating the indirect diffuse illumination for each surface and building the irradiance cache. The pre-pass will use higher quality settings than the final pass by applying the **Pre-pass multiplier** to the **Max error allowed** and **Max sample spacing** values. The result of the pre-pass is a denser set of irradiance cache samples that produces a smoother result in the final rendering (because each shading location averages more irradiance cache samples).

Recycling Irradiance Samples

The irradiance cache can be saved to a file and re-used in subsequent renderings by providing a **Cache file** name and checking the **Read** or **Write** boxes.

A separate cache can be referenced for each frame by including $F*n* in the file name, where n is the number of digits for the frame number.

A cache file can accumulate samples from multiple renderings by enabling read and write at the same time. This approach can be used to generate a single cache file with all values required for an animated sequence.

Optimizing Indirect Shading

In many scenes a significant portion of the time needed to compute indirect diffuse illumination is spent evaluating the shading for surfaces hit by indirect rays. The **Indirect Shading** mode in the Indirect Only section of the GI tab determines how objects are shaded when hit by indirect rays. The choices are:

Shader
Air evaluates the surface shader assigned to the object to compute the indirect illumination contribution. This is the slowest and most accurate mode.

Lambert
Air estimates the indirect contribution by computing a Lambert (diffuse) illumination result and multiplying it by the object's color attribute. This mode is faster than the Shader mode because the object's surface shader does not have to be evaluated.

Constant
The object's color attribute value is used as the indirect shading result. This is the fastest mode since it does not require evaluating any shaders or illumination.

The indirect shading mode can be set on a per-object basis using an Air Stream node or using the Advanced Attributes on an Air Stream surface shader node.

Optimizing Indirect Environment Queries

Air 11.06 introduces a new caching mechanism for global illumination environment queries, which can be enabled in the Environment shaders section of the Shading tab. The caching mechanism can greatly accelerate evaluation of the environment for indirect rays that miss all objects. Note that when the cache is enabled, the environment is queried based only on the ray direction; the current

shading location is ignored.

IPR with Indirect Diffuse Illumination

Indirect diffuse illumination can be used while interactively shading and lighting a scene with TweakAir. The IPR Settings section of the Global Illumination tab allows many Global Illumination settings to be customized for IPR rendering. By default most of the settings are inherited from the values used for normal rendering with Air, with the following exception:

The irradiance cache is enabled. The cache provides a faster initial rendering and also permits samples to be re-used when the IPR view updates. If you do not wish to use the cache for IPR, lowering the number of rays per sample can help keep IPR refresh rates up.

See Also:

Global Illumination Tab
Air User Manual: Lighting -> Indirect Lighting

10.2 Ambient Occlusion

Air will estimate global illumination using ambient occlusion when the Global Illumination Mode on the Global Illumination tab is set to Occlusion.

Air computes ambient occlusion casting shadow rays from the current shading location to estimate the extent to which the region being shaded is blocked or occluded by nearby objects. The occlusion result is then used to estimate the light from the global environment visible at the current shading location.

Computing occlusion is faster than computing indirect illumination because a ray cast for occlusion only has to determine the extent to which it is blocked by an object, whereas a ray cast for indirect illumination also has to compute the illumination at an intersection point. Air also has special optimizations for tracing shadow rays that are not available when tracing indirect rays.

Sample Quality

The Sample Quality section of the Global Illumination tab has controls for tuning the quality of the ambient occlusion estimate.

The quality of the occlusion estimate depends on the number of **Rays per Sample**. More rays produce smoother results. Using too few rays can produce a grainy appearance.

Occlusion computations can be accelerated in a couple ways. The **Max trace distance** gives the maximum distance Air will trace a ray looking for intersecting objects. Using a lower distance will reduce rendering time.

When **Adaptive ray casting** is enabled, Air casts fewer rays initially (1/4 the total number) and additional rays only where there are significant differences in the results of the first set of rays.

Occlusion Caching

To accelerate occlusion rendeing, Air supports occlusion caching - a technique in which occlusion samples are stored and re-used at nearby shading locations.

When rendering with an occlusion cache, Air first attempts to estimate the occlusion at the current location by re-using values already in the cache. Whether a given cache value can be re-used is determined by the **Max error allowed**. A higher maximum error allows cached values to be re-used more often. If Air cannot find enough usable values in the cache, it will generate a new estimate by casting rays and store the new result in the cache for possible re-use.

To ensure that all areas of the image receive some indirect samples, the **Max sample spacing** setting gives the maximum distance in pixels within which Air will look for usable samples in the cache.

Occlusion Pre-pass

Air has a special pre-pass capability that generally produces smoother final results when rendering with an occlusion cache. When **Uses pre-pass** is enabled, Air first renders a pre-pass evaluating the occlusion for each surface and building the occlusion cache. The pre-pass will use higher quality settings than the final pass by applying the **Pre-pass multiplier** to the **Max error allowed** and **Max sample spacing** values. The result of the pre-pass is a denser set of occlusion samples that produces a smoother result in the final rendering (because each shading location averages more occlusion cache samples).

Recycling Occlusion Samples

The occlusion cache can be saved to a file and re-used in subsequent renderings by providing a **Cache file** name and checking the **Read** or **Write** boxes.

A separate cache can be referenced for each frame by including $F*n* in the file name, where n is the number of digits for the frame number.

A cache file can accumulate samples from multiple renderings by enabling read and write at the same time. This approach can be used to generate a single cache file with all values required for an animated sequence.

Environment Sampling

Air has two ways of using the occlusion information to filter the light contribution from the global environment. The Occlusion Only section of the Global Illumination tab has an **Environment Sampling** control for selecting between the following methods:

Accurate (once per unoccluded ray)
The environment is sampled at each unoccluded ray. This method produces the most accurate results. It will be slower to render when the occlusion cache is not used, but it can be faster than the approximate method when the cache is used because the environment result will be saved with the cache.

Approximate (once per sample)
The environment is sampled only once for each global illumination sample. The environment is sampled over a hemispherical region centered around the average unoccluded direction.

Note that if you change the environment sampling method, any cache files will need to be re-generated to include the correct information.

Baking Occlusion to Texture Maps

BakeAir can be used to bake occlusion information to texture maps for accelerated re-rendering. The main requirement for baking occlusion is that each object must have a set of texture coordinates that maps the surface one-to-one into the unit square in texture space. (In other words, no two points on the surface should share the same texture coordinates.) See the following tutorial for detailed information:

Tutorial: Baking Global Illumination

IPR with Ambient Occlusion

Ambient occlusion can be used while interactively shading and lighting a scene with TweakAir. The IPR Settings section of the Global Illumination tab allows many Global Illumination settings to be customized for IPR rendering. By default most of the settings are inherited from the values used for normal rendering with Air, with the following exceptions:

The occlusion cache is enabled. The cache provides a faster initial rendering and also permits occlusion samples to be re-used when the IPR view updates. If you do not wish to use the cache for IPR, lowering the number of rays per sample can help keep IPR refresh rates up.

The environment sampling mode is set to Approximate. In this mode TweakAir can automatically update the global illumination results whenever environment shader parameters are modified.

See Also:

Global Illumination Tab
Air User Manual: Lighting -> Ambient Occlusion

11 IPR - Interactive Rendering with TweakAir

Air Stream provides an interactive rendering option using TweakAir, a special version of Air designed for fast preview of shading and lighting changes.

Starting an Interactive Preview

- To start an IPR session, choose Start IPR from the Air Stream menu or click the green IPR button in the Air shelf.

Air Stream will export the current scene and start TweakAir as a separate process. The preview image will appear in the Air Show framebuffer.

Interactive Preview Updates

Once an IPR session is started, Air Stream automatically sends changes made to the following shading features:

- Light shader parameter values
- Light transformations
- Surface shader parameters, including input parameters to any networked nodes
- Displacement shader parameters, except when true displacement is used
- Environment and imager shaders
- Camera moves when the relevant option is enabled in the IPR section of the Render tab

Stopping an Interactive Preview

- To stop an IPR session, select Stop IPR fromthe Air Stream menu or click the red IPR button in the Air shelf.

IPR Settings

IPR settings can be changed in the IPR section of the globals Render Tab.

Optimizing Shading for IPR

You can select the shading mode for IPR in the IPR section of the Render tab. The default shading method uses a fast re-lighting technique that de-couples the evaluation of surface and displacement shaders from the evaluation of light sources. In this mode TweakAir can re-shaded an object when lighting changes without re-evaluating the assigned surface or displacement shaders, and it can re-shade objects when a surface shader is tweaked without re-evaluating the light sources. This method is typically much faster than fully reshading an object when any change is made. This mode works well for most surface shaders that have a basic illumination model:

diffuse + specular (phong,blinn,anisotropic) + reflections + incandescence

For more complex shading models, use the full shading mode for IPR that reshades an object whenever any changes are made to its assigned shaders.

TweakAir also provides a fast diffuse shading mode that shades all objects as simple Lambertian diffuse reflectors. The diffuse mode may be useful for quickly roughing in the lighting for a scene.

The IPR shading mode can be set on a per-object basis using an Air Stream node.

See Also:

Air User Manual: TweakAir for Interactive Rendering

12 Baking Maps with BakeAir

BakeAir is a special version of Air designed for baking shading and lighting results to texture maps. For detailed instructions on baking, you may wish to look at one of the following tutorials:

Introduction to Baking with BakeAir
Baking Global Illumination
Baking a Procedural Pattern

Bake Map Pre-Requisites

- Air Stream can bake maps for polygon meshes, subdivision surfaces, and NURB surfaces.
- Each object to be baked must have a UV set that maps the surface into the unit square in texture space. The mapping should be one-to-one; ie., each point on the surface should map to a unique u,v coordinate.
- Most of the standard shaders used for baking will bake only the front side of a surface (the side that is visible when rendering the object as single-sided). It is possible to bake the back side or even both sides using a custom shader.

Bake Map Settings

The Bake tab in the Render Settings dialog has controls for configuring the baking process. Many of the controls can be left at their default values, but you'll likely want to select the channels to bake and tune the bake map size:

Bake Channels

The **Bake Channels** control has a list of standard output values to bake. The GI bake modes are designed to be used in conjunction with Air Stream's Global Illumination features.

An arbitrary set of user-defined channels can be baked by setting the **Bake Channels** selection to `Custom` and entering a comma-separated list of channels in the **Custom Channels** control. Each channel should include a full inline type declaration.

Bake Map Size

Air Stream computes a resolution for each baked map based on the **Pixel Size** value. Smaller pixel size values produce larger and more detailed maps. **Max Map Size** gives the maximum width and height of any baked map. The size of the baked map for a particular object can be set explicitly in the Bake section of an attached Air Stream node.

Baking

To start BakeAir and bake the selected texture maps, choose *Bake Maps* from the Air Stream menu.

Using Baked Maps

To enable baked maps to be used when rendering, Air Stream can attach a user attribute to each baked object with the map name. A shader can query the user attribute for the map name, and then query the map in whatever manner makes sense.

For all bake channels except the GI bake modes, enable **Export map name as user attribute** under Map File on the Bake tab to tell Air Stream to include the map names as user attributes in the exported RIB file. The user attribute name will be the same as bake map base name.

For the GI bake modes, enable use of baked maps in the Baked Maps section of the Global Illumination tab. Set the **Use Baked Maps** selection to the entry corresponding to the GI bake mode used for baking.

Baked Map Location

Baked maps will be created in a scene-specific subdirectory of the Textures directory specified on the Air Export page. If you rename the current scene, you'll need to rename the directory with the baked maps to have access to any existing baked maps.

See Also:

Render Globals: Bake Tab
Tutorial: Introduction to Baking with BakeAir
Tutorial: Baking Global Illumination
Tutorial: Baking a Procedural Pattern

12.1 Tutorial: Introduction to Baking with BakeAir

For this tutorial you will need the shaders available in Air 11.05 and later.

This tutorial describes how to bake the standard color output from a surface shader and re-render using the baked maps.

Begin by creating a simple scene in Maya:

- Start a new scene in Maya
- Create a simple sphere on a plane with a shadow-casting point light
- Assign a Marble texture to the Color input of the default lambert surface shader
- Render a preview image with Air

The next step will render a texture map for each object storing the standard color output from the surface shader at each texel:

- From the Air Stream menu select *Bake Maps*

Air Stream will export the current scene and start a bakeair process which will render a separate texture map for each object. Wait for the baking process to finish.

Now we can use the baked texture maps for re-rendering. First we tell Air Stream to include a reference for each object's texture map in the exported scene file:

- Display the Render Settings dialog.
- Select Air Stream as the current renderer.
- Click on the Bake tab
- Check **Export map name as user attribute**

Each object will now be assigned a user attribute with the file name of its corresponding baked texture map. A surface shader can query that attribute to pick up the file name and reference that when rendering:

- Display the Render Layer editor (menu *Window -> Rendering Editors -> Render Layer Editor*)
- Select the sphere and plane objects in the Maya scene
- In the Render Layer Editor, from the Layers menu choose *Create Layer from Selected*
- Right click the new layer (layer1), and from the pop-up menu select

  Overrides -> Create New Material Override -> Air Stream Surface Shader

- Select the sphere object. From the Window menu select *Attribute Editor*.
- In the Attribute Editor click on the new `airstreamSurfaceShader1` tab.
- Click the file browser button next to the Shader control, and select the BakedSurface shader from the shaders directory of the Air installation.

Because the BakedSurface shader defaults are designed to work with Air Stream, the scene is ready to render using the baked maps.

- In the Render Settings dialog make sure the current Render Layer is set to `layer1`.
- Render a preview with Air to see the results of rendering using the baked texture maps.

Map Size and Detail

The size of each texture map (and thus the amount of detail it represents) is determined by two settings on the Bake tab in the Render Settings dialog. The Pixel Size value gives the approximate size of each texel in the map in world space units. Air Stream uses this value to compute a map size for each object based on it's surface area and texture mapping. Smaller values produce larger maps with more detail. The maximum map size is given by the Max Map Size parameter.

To see the effect of increasing the Pixel Size value:

- Set the Render Layer back to `masterLayer` so the maps can be re-baked with the original shaders.
- On the Bake tab change the **Pixel Size** value from the default of 0.1 to 0.2
- Choose Bake Maps from the Air Stream menu

- Set the Render Layer to `layer1`
- Render a preview. The texture detail should be coarser than the result from the first set of baked maps.

Diffuse, Unshadowed Diffuse, and Shadow Passes

BakeAir can render an arbitrary number of channels to a texture map. The follow steps outline how to save separate channels with information for diffuse, unshadowed diffuse, and shadow output images in a final rendering.

- Set the current Render Layer to `masterLayer`
- Change the Pixel Size value back to 0.1
- Select the Images tab.
- Under Additional Passes select the `Diffuse` render pass. Click the Create button.
- Under Additional Passes select the `Unshadowed` render pass. Click the Create button.
- Under Additional Passes select the `Shadow` render pass. Click the Create button.
- Render a preview to see the original rendering with multiple passes.
- Select the Bake tab.
- Set Bake Channels to `Diffuse + Unshadowed`
- From the Air Stream menu choose *Bake Maps*. Wait for the baking process to finish.

Each baked image now has 6 channels, 3 with the diffuse output from the surface shader, and 3 with the unshadowed diffuse value. The BakedSurface shader can use those values to compute a shadow output, but we first need to tell the shader where to find the unshadowed diffuse values:

- Set the current Render Layer to layer1
- Select the sphere object
- In the Attribute Editor select the airstreamSurfaceShader1 tab
- Set the **Channel with Unshadowed** parameter to 3, the starting channel of the unshadowed values in the baked texture maps.
- Render

Adding Global Illumination

By default the global illumination contribution to the final color is emitted in a separate __indirect output variable from each surface shader. We can add that to the baked texture to provide an indirect pass in the final rendering:

- Set the current Render Layer to `masterLayer`
- Select the Images tab.
- Under Additional Passes select the `Indirect` render pass. Click the Create button.
- Select the Global Illumination tab
- Set the **Global Illumination Mode** to `Occlusion`
- Expand the Cache Properties section, and check **Use cache**.
- Render a preview image
- Select the Bake tab
- Set Bake Channels to `Diffuse + Unshadowed + Indirect`
- From the Air Stream menu choose *Bake Maps*. Wait for the baking process to finish.
- Select the Global Illumination tab and set **Global Illumination Mode** to `Off`
- Set the current Render Layer to `layer1`
- Select the sphere object
- In the Attribute Editor select the airstreamSurfaceShader1 tab
- Set the **Channel with Indirect** parameter to 6, the starting channel of the indirect values in the baked texture maps.

- Render

12.2 Tutorial: Baking Global Illumination

For this tutorial you will need the shaders available in Air 11.05 and later.

This tutorial covers the Air Stream options for creating and using baked maps to accelerate rendering with global illumination.

Here's an overview of the process:

- Select and configure the desired GI mode and parameters on the Global Illumination tab in the render globals.
- On the Bake tab choose an appropriate GI bake mode.
- Bake the maps (by selecting Bake Maps from the Air Stream menu)
- On the Global Illumination tab, expand the Baked Maps section and set Use Baked Maps For to the mode corresponding to the bake mode selected above.
- Render normally

Prologue: Create a Simple Test Scene

- Start a new scene in Maya.
- Create a simple sphere on a plane model.
- Open the render globals window, and select Air Stream as the renderer to use.
- Select the Shading tab. Expand the Environment Shaders section. Click *Create Environment*.
- In the Attribute Editor window for the environment shader, click the file browser button next to the Shader control.
- Choose the `envPhysicalSky` shader from the `shaders` directory of the Air installation.

Baking Indirect Illumination

- Return to the Render Settings window and select the Global Illumination tab.
- Set the **Global Illumination Mode** to `Indirect`.
- Render a preview image for later comparison.

To bake the indirect lighting information to a texture map for each object:

- Select the Bake tab in the Render Settings window.
- Set **Bake Channels** to `GI: Indirect Illumination`.
- Choose *Bake Maps* from the Air Stream menu.
- Wait for the baking process to finish.

To use the baked maps for final rendering:

- Go to the Global Illumination tab in the Render Settings window.
- Expand the Baked Maps section and set **Use Baked Maps For** to `Indirect`.
- Render

The baked maps include the indirect illumination contributions from both surfaces and the environment. The maps need to be re-baked to reflect any changes made to the shading of objects or the environment.

Baking Occlusion

- If continuing from the above section on Baking Indirect Illumination, return to the Global Illumination tab. Under Baked Maps set **Use Baked Maps For** to `Nothing`.
- Set the **Global Illumination Mode** to `Occlusion`.
- Render a test image.

Air Stream provides three ways to bake occlusion. The method that produces the greatest acceleration of final rendering stored occlusion and the sampled environment results:

- Select the Bake tab in the Render Settings window.
- Set **Bake Channels** to `GI: Occlusion + Environment`.
- Choose *Bake Maps* from the Air Stream menu.
- Wait for the baking process to finish.

To use the baked maps for final rendering:

- Go to the Global Illumination tab in the Render Settings window.
- Expand the Baked Maps section and set Use Baked Maps For to Occlusion + Environment.
- Render

The second method of baking occlusion stores occlusion and a "bent normal" vector. The bent normal is the average unoccluded direction. The occlusion light shader uses that information to sample the environment more accurately. Because the environment is still sampled during final rendering, changes to the environment shader do not require the maps to be baked again.

- On the Global Illumination tab, set **Use Baked Maps For** to `Nothing`.
- Under Occlusion Only, set **Environment Sampling** to `Approximate`.
- Render a preview image for later comparison.
- On the Bake tab set **Bake Channels** to `GI: Occlusion + Bent Normal`.
- Choose *Bake Maps* from the Air Stream menu.
- Wait for the baking process to finish.

To use the baked maps for final rendering:

- On the Global Illumination tab, set **Use Baked Maps For** to `Occlusion + Bent Normal`.
- Render

The third method of baking occlusion stores only a single channel occlusion value. This is the least accurate method, but it only requires one channel per map. The baked occlusion channel can even use 8-bit precision for smaller map size and faster texture sampling.

- On the Global Illumination tab, set **Use Baked Maps For** to `Nothing`.
- On the Bake tab set **Bake Channels** to `GI: Occlusion Only` Set **Quantization** to `8-bit`.
- Choose *Bake Maps* from the Air Stream menu.
- Wait for the baking process to finish.

To use the baked maps:

- On the Global Illumination tab, set **Use Baked Maps For** to `Occlusion Only`.
- Render

See Also:

Baking Maps with BakeAir
Render Globals: Bake Tab

12.3 Tutorial: Baking a Procedural Pattern

For this tutorial you will need the shaders available in Air 11.05 and later.

This tutorial shows how to bake a procedural pattern to a map and how to use the baked map for re-rendering.

- Start a new scene in Maya
- Create a torus.
- In the Attribute Editor select the lambert1 surface shader. Assign a Marble texture to the Color attribute.
- Render a preview image

To bake the procedural pattern:

- Open the Render Settings window.
- Set Air Stream as the renderer to use.
- Select the Bake tab.
- Set Bake Channels to Unlit Color.
- From the Air Stream menu choose Bake Maps.
- Wait for the baking process to finish.

The procedural pattern has now been baked to a texture map that we can use to accelerate re-rendering. First we tell Air Stream to include a reference for each object's texture map in the exported scene file:

- On the Bake tab check **Export map name as user attribute**

Each object will now be assigned a user attribute with the file name of its corresponding baked texture map. The final step is to configure a shading network with a node to query the baked map.

- Select the torus.
- Assign a new Lambert material.
- Open the Hypershade window.
- Middle-click and drag the `lambert2` material to the work area.
- From the Create menu, choose *General Utilities -> Air Stream Generic Shader*.
- The new generic shader should be displayed in the Attribute Editor. Click the file browser next to the Shader entry. Select the `genQueryBakedMap` shader from the `shaders` directory of the Air installation.

Note that the **Attribute Name** value in the shader defaults to `user:bakedmap`. This identifies the user attribute the node will query for the baked map name. The name after `user:` should match the **Base Name** specified in the Bake tab when baking the texture. The default name of `bakedmap` matches the default base name on the Bake tab, so we don't have to change anything in this case.

- Back in the Hypershade window, connect the `airs_OutputColor` output from the `airstreamGenericShader` node to the `color` input of the `lambert2` node.
- Render

See Also:

Baking Maps with BakeAir
Render Globals: Bake Tab

13 Particles

Air Stream exports particles for rendering based on the particle shape rendering type and associated parameters.

Air's instancer shaders provide a simple, flexible mechanism for replacing particles with geometry at render time. Per-particle attributes can be used to control the shading and the shape of replacement objects. See the separate section on Particle Instancing for more information.

Particle Render Types

This section describes the details of particle export for each render type:

Points, MultiPoint

Because Air treats points as true 3D entities, Air Stream uses the particle radius defined for the Sphere render type as the particles size for point and multipoint modes (ignoring the integral **Point Size** parameter in Maya). To set the particle size, temporarily set the render type to Sphere, add the attributes for the current render type, set **Radius** to the desired value, then restore the render type to Point or MultiPoint.

By default Air renders point primitives as hexagons, designed for fast rendering of large numbers of tiny (~ pixel size) points. You can change the representation to a sphere or disk instead on the Render tab in the render globals:

- Expand the Global Defaults panel.
- Expand the Particles sub-panel.
- Set the Point Render Type to the desired value.

The MultiPoint render mode is implemented using an Air instancer shader which creates a cluster of points for each particle on-demand at render time.

Streak, MultiStreak

Because Air treats points as true 3D entities, Air Stream uses the particle radius defined for the Sphere render type as the particles size for Streak and MultiStreak modes (ignoring the integral **Line Width** parameter in Maya). To set the particle size, temporarily set the render type to Sphere, add the attributes for the current render type, set **Radius** to the desired value, then restore the render type to Streak or MultiStreak.

The tail fade option is exported using a per-vertex opacity attribute. For tail fading to be rendered, you will need to either use an Air surface shader that modulates it's output by opacity (most do), or use a ParticleSamplerInfo node to connect per-particle transparency to the Transparency input of a Maya shader.

The MultiStreak render mode is implemented using an Air instancer shader that creates multiple streaks for each particle at render time.

Blobby Surface

Air can render a particle system as a blobby primitive, including linear motion blur. The `Threshold` Maya parameter is currently ignored.

The time and memory used by the tessellator for blobby primitives is very sensitive to the Surface tolerance setting, which can be tuned on a per object basis using an Air Stream node. Even a small change in the tolerance setting can have a big impact on render time and memory use.

E.g., increasing the tolerance from the default of 0.5 to 0.7 will typically cut memory use by a factor of 2.

Cloud

When the particle render type is set to Cloud, Air Stream exports the particle system as an Air volume primitive. Although a very basic ParticleCloud shader is provided, in general the appearance of the Air volume primitive will not be similar to a Maya cloud render. You may wish to use the Air particle surface shader when rendering particles as volumes.

If you wish to use ray-traced shadows, you will need to enable shader evaluation for shadow rays when using Air 11.01 or earlier. This attribute can be set in the Attributes section of an Air Stream Surface shader node or by adding the following line in the RIB Box section of an Air Stream node:

```
Attribute "shade" "transmissionhitmode" "shader"
```

Sprites

Particle sprites are rendered using Air's native 'patch' point type which creates sprite patches from point data at render time. Sprite size in X and Y are taken from the corresponding sprite attributes. Sprite twist is supported with Air 11.05 or later.

Texture sequence cycling is supported when hardware texture cycling is enabled in a file texture node.

Spheres

Air supports spherical particles as a native primitive type.

Tube

The tube render type is currently not supported.

Particle Data

Air Stream currently exports the following per-particle data if it is available:

Maya name	RIB primitive variable
rgbPP	color Cs
opacityPP	color Os
radius, radiusPP	float width
particleId	float id
velocity	vector velocity

For sprites, Air Stream also exports:

Maya name	RIB primitive variable
spriteTwist, spriteTwistPP	float patchangle
spriteNum, spriteNumPP	float SpriteNum

Air Stream also looks for ParticleSamplerInfo nodes in the shading network, and exports any of the following particle data that is used as an input to another shading node:

age, birthTime, lifespanPP, userScalar*PP, userVector*PP

Air Stream also exports particle data referenced by name in the parameter list of any instancer shader attached to the particles.

Particle Instancing

Air's instancer shaders provide a powerful and flexible mechanism for replacing particles with instanced geometry at render time. An instancer shader can query any per-particle attribute, and use the resulting value to generate or reference replacement geometry in whatever manner is desired.

Air ships with two instancer shaders that provide basic object instancing capabilities for single and multiple archives:

`VInstanceArchive`

The VInstanceArchive instancer creates instances of a single archive file at render time. A per-particle size value can be applied to create instances of different sizes. Instances can be rotated based on a per-particle attribute. Any vector or scalar particle attribute can also be exported as a user attribute, which can be queried by a shading network. This mechanism allows per-particle attributes to be used to vary the shading of each instance.

`instMultipleArchives`

The instMultipleArchives adds the ability to select an archive from a numbered set for each particle based on a user-controlled index value.

Both of these shaders use pre-generated RIB archive files for the referenced geometry. You can create a RIB archive file by selecting objects in Maya and then choosing *Export -> Export Selection to Archive* from the Air Stream menu.

See Also:

Particles
Tutorial: Instancing Multiple Archives
Tutorial: Customizing Shading for Instanced Geometry

13.1 Particle Instancing

Air's instancer shaders provide a powerful and flexible mechanism for replacing particles with instanced geometry at render time. An instancer shader can query any per-particle attribute, and use the resulting value to generate replacement geometry in whatever manner is desired.

Air ships with two instancer shaders that provide basic object instancing capabilities for single and multiple archives:

`VInstanceArchive`

The VInstanceArchive instancer creates instances of a single archive file at render time. A per-particle size value can be applied to create instances of different sizes. Instances can be rotated based on a per-particle attribute. Any vector or scalar particle attribute can also be exported as a

user attribute, which can be queried by a shading network. This mechanism allows per-particle attributes to be used to vary the shading of each instance.

`instMultipleArchives`

The instMultipleArchives shader extends the functionality of the VInstanceArchive shader by providing the ability to select an archive from a numbered set for each particle based on a user-controlled index.

Both of these shaders use pre-generated RIB archive files for the referenced geometry. You can create a RIB archive file by selecting objects in Maya and then choosing *Export -> Export Selection to Archive* from the Air Stream menu.

For detailed instructions on using these instancer shaders, see the following tutorials:

Tutorial: Custom Shading for Instanced Geometry
Tutorial: Instancing Multiple Archives

13.2 Tutorial: Custom Shading for Instanced Geometry

For this tutorial you will need the shaders available in Air 11.05 and later.

This tutorial describes how to use per-particle attributes to customize the shading of each copy of an instanced archive.

Here's an outline of the two part process:

Phase 1: Archive creation

- Load or create the objects that are to be instanced at each particle location.
- For objects whose shading is to be customized on a per-particle basis, configure the shading network to look for a user attribute for each per-particle property.
- Create an archive file with the objects that are to be instanced.

Phase 2: Particle instancing

- Load or create a Maya particle system.
- Create and configure the per-particle attributes that are to drive the instanced shading network.
- Assign an Air instancer shader node to the particle system and make sure the VInstanceArchive instancer shader is selected.
- Select the archive created in phase 1 as the archive to use for instancing.
- Tell the instancer shader which particle attributes to translate into Air user attributes.
- Render.

The remainder of this tutorial provides step-by-step instructions for instancing a simple object with a different color for each instance.

Phase 1: Archive creation

- Start a new scene in Maya.
- Create a simple shape such as a torus centered at the origin.

Now we're going to modify the object's shading network so it can detect and use a per-object color attribute.

- Open the Hypershade window.
- Drag the lambert1 shader to the work area.
- From the Create menu, look under General Utilities and select Airstream Generic Shader.
- The new generic shader node should be displayed in the Attribute Editor. In the Attribute Editor, click the file browser button next to the Shader control, and choose the genUserAttributes shader from `$AIRHOME/shaders`.
- In the shader parameter list, find the **Color 1 Name** parameter and set it's value to `UserVector1PP`
- Return to the Hypershade window. Connect the **airs_OutputColor1** output from the airStreamGenericShader1 node to the **color** input of the lambert1 shader.
- Return to the main Maya window. Select the torus.
- From the Air Stream menu, choose Export to RIB -> Export Selection to Archive.
- Pick a location and file name for the archive, say `torus.rib`

Air Stream will export the selected object and its shading network to the specified file.

That completes phase 1, creating the archive for instancing.

Phase 2: Particle instancing

- Start a new scene in Maya.
- In the Dynamics menu set, find the Particles menu and select Create Emitter.
- In the Attribute Editor for the emitter, under Basic Emission Speed Attributes change the speed from 1 to 10.
- Play the animation and stop around frame 20.
- Zoom in so the particles fill most of the display.
- In the Attribute Editor click on the particle shape tab.
- Under Render Attributes, change the Particle Render Type to Sphere.
- Test render (in the AirStream menu, choose Render Preview).

The next step is to apply an instancer shader which will create the archive instances at render-time.

- Select the particle system.
- From the Air Stream menu, choose Instancer Shader -> Create New Instancer Shader
- In the Attribute Editor select the new instancer node, airstreamInstancerShader1.
- We will use the default instancer shader, VInstanceArchive.
- In the Attributes section, check **Hide base object** to hide the particles at render time.
- Set the **Archive** parameter value to full path for the torus archive created earlier.
- Test render.

The particle system should render with each particle replaced by a copy of the torus archive. All instances will have the same color at this point.

Air needs to know the region occupied by instanced geometry in order to efficiently and accurately apply the instancer shader. As long as the instanced geometry lies within the bounding box of the base object, the instanced geometry should appear properly. The simplest way to meet this condition is to ensure that the archive geometry fits inside the bounding sphere for each particle. You can use the **Scale** parameter in the VInstanceArchive shader to reduce the size of the archive if necessary.

Another way of meeting the bounding box condition is to use the **Region bound** attribute on the instancer shader to expand the bounding box for the base object so all instances are encompassed.

Next we'll create a per-particle attribute with a different color for each particle:

- Select the particle system and click on the particleShape tab.
- Under Add Dynamic Attributes click General.

- In the Add Attributes window, choose the Particle tab.
- Select userVector1PP and click Add.
- Click OK to close the window.

In the Attribute Editor:

- On the particleShape tab, find the **User Vector 1 PP** entry under Per-Particle (Array) Attributes.
- Right-click the box next to **User Vector 1 PP** and choose Create Ramp from the drop-down menu.
- Re-run the particle sim to initialize the new user attribute with the color ramp values.

The last step is to tell the instancer shader to convert the per-particle values to a user attribute for each instance. The user attribute will be queried by the torus' shading network, and the result will be used as the object color.

- In the Attribute Editor select the tab for the instancer shader, airstreamInstancerShader1.
- Set the value for **Prim Var To Attribute 1** to `UserVector1PP`
- Render the particle system to see instanced archives of different colors.

This technique can be used to vary any color or float (scalar) shading property for instanced geometry using a per-particle attribute.

Motion Blur

When the Motion Blur parameter of the instancer shader is set to 1, the shader will apply translational motion blur to the instanced geometry. In order for motion blur to appear, the global motion blur mode on the Render tab must be set to include defomation blur (so the particle motion data is exported).

See Also:

Particles
Tutorial: Instancing Multiple Archives

13.3 Tutorial: Instancing Multiple Archives

For this tutorial you will need the shaders available in Air 11.05 and later.

This tutorial describes how to use an instancer shader to replace each particle by one of a numbered set of pre-generated object archives.

The overall process has two phases:

1. Create a numbered set of object archives

2. Apply the instMultipleArchives instancer shader to the particle system and configure it to use the pre-generated archives

We'll now discuss each phase in detail.

Creating a Numbered Set of Archives

The instancer shader we will be using works with a numbered sequence of RIB archive files. E.g.,

```
tree0.rib
tree1.rib
tree2.rib
...
```

To create an archive file with Air Stream:

- Select the objects to go in the archive
- From the Air Stream menu, select Export -> Export Selection to Archive.
- Select a path and enter a file name.

Create an archive for each object or group of objects that will be instanced, numbering the archives consecutively beginning with 0. Archived geometry should usually be placed near the origin - any transformation applied to the object(s) will be included in the archive file.

Configuring the Instancer for Multiple Archives

- Load or create the particle system to be used for instancing.
- Set the particle render type to Sphere.
- With the particle system selected, go to the Air Stream menu and choose Instancer Shader -> Create New Instancer Shader.
- In the Attribute Editor select the tab with the new instancer shader node, airstreamInstancerShader1
- In the Attributes section of that node, check **Hide base object** so the particles will be hidden when rendering
- Click the file browser button next to the Shader control. Select the `instMultipleArchives` shader from the shaders directory of your Air installation.
- Set the shader's **Archive Name** parameter to the full path and file name of one of the archive files created above.
- Depending on the size of the objects in the archive files, set the **Scale** shader parameter so that each instance will be contained in the sphere region around each particle.
- Render a test image

The particle system should render with each particle replaced by the first (0-th) archive in the file sequence.

The next step is create a per-particle attribute to be used as the archive index so each particle can reference a different archive file.

- With the particle system selected, click on the particle shape tab in the Attribute Editor.
- Under Add Dynamic Attributes click on General.
- In the Add Attribute window select the Particle tab.
- Select `userScalar1PP` in the list of attributes, and click the Add button. Then click the close button.
- Find the Per-Particle Attributes section of the particle shape tab in the Attribute Editor.
- Right-click the box next to **User Scalar 1 PP** and choose Create Ramp from the drop-down menu.
- Re-run the particle sim to initialize the new ramp attributes.
- Right-click the box next to **User Scalar 1 PP**, select the array listed, and choose Edit Array Mapper from the side menu.
- Set the Max Value to slightly less than (maximum archive index+1). E.g., if the maximum index is 4, use a value like 4.999

The final step is to tell the instancer to use the new attribute as the archive index:

- In the Attribute Editor, select the tab for the instancer shader.
- Set the **Archive Index Prim Var** parameter value to `UserScalar1PP`
- Render

The particle system should render with each particle replaced by one of the archive files. We've used a simple linear ramp for the index value, so the distribution of archives should fairly even. You can of course edit the ramp or use an expression or other mechanism to alter the distribution of archives to suit the project.

Motion Blur

When the Apply Motion Blur parameter of the instancer shader is set to 1, the shader will apply translational motion blur to the instanced geometry. In order for motion blur to appear, the global motion blur mode on the Render tab must be set to include defomation blur (so the particle motion data is exported).

See Also:

Particles
Tutorial: Customizing Shading for Instanced Geometry

14 Fur, Hair, Curves

In this section:

14.1 Maya Fur

(Fur rendering requires Air 11.07 or later.)

Air Stream can export Maya fur descriptions for rendering with Air.

Fur Attributes

Air Stream translates most of the standard fur description attributes. Extra attributes can be applied to the fur description to customize fur rendering with Air. To add these extra attributes:

- Select the fur description or a connected fur feedback node.
- Choose Fur, Hair, Curves -> Add Fur Attributes from the Air Stream menu.

Fur Creation

Fur curves are generated at render time using a pair of Air instancer shaders. The first instancer shader samples the base object for the fur, generating a particle for each curve storing the surface parameters for that curve (root location, surface normal, tangent vectors, and texture coordinates). The particles are grouped into clusters and assigned a second instancer shader that converts the particles into curves if and when a given cluster is visible in the rendered image.

By default the curve vertices are connected using a sequence of linear line segments. The **Air Spline Type** extra attribute can be used to select a different interpolation method for the vertices:

`b-spline`: render each curve as a cubic b-spline
`catmull-rom`: render each curve as a cubic catmull-rom spline

Either of the cubic spline types will produce a smoother curve; they may also permit the curve to be represented with fewer segments, saving memory and render time.

By default Air renders each curve as a set of line segments, which is very efficient for thin curves used to represent fur. For wider curves, use the **Air Curve Type** extra attribute to select a different representation:

`ribbon`: render each curve as a quad strip
`tube`: render each curve as a tube

By default each curve will always face the camera, which works fine with the default fur shader used by Air Stream. If you wish the curves to represent something other than fur, you can specify a fixed orientation using the **Air Curve Orientation** extra attribute:

`face u tangent`: the curve will be oriented perpendicular to the u tangent vector
`face v tangent`: the curve will be oriented perpendicular to the v tangent vector

With Air 12.10 and later, ribbon and tube curves can optionally be tapered at either end. The **Air Tip Taper** and **Air Root Taper** attributes determine where along the curve tapering begins. The interpretation of the taper value is influenced by the corresponding taper type:

`width`: the taper value is multiplied by the curve width
`length`: the taper value is multiplied by the curve length
`absolute`: the taper value is the distance along the curve at which tapering begins

Fur Shading

Fur is shaded by treating the fur description as a shader, passing all shading-related parameters to a surface shader. Because the surface shader for the fur is tied to the fur description, the shading parameters can be treated like those of other surface shaders:

- Shading parameters such as Base Color can be driven by Maya shading nodes without the need to bake maps.
- Changes made to fur shading parameters will update in an IPR view.

By default Air Stream will use the `ASM_FurSurface` shader included with Air Stream. You can specify a custom shader using the Air Custom Surface extra attribute. The compiled shader file for a custom fur surface shader must be placed in the `shaders` directory of the Air Stream installation.

Air Stream includes one extra fur surface shader - `ASM_FurPhongSurface` - which computes a standard phong illumination model for the fur.

A custom shader used for fur may modify the transparency of the fur curve, in which case the shader needs to be evaluated when the curve is intersected by a shadow ray. Enable shader evaluation for shadow rays in this case by enabling the **Air Surface Modifies Transparency** extra attribute.

For more control over fur shading, you can assign a custom Air shader. To do that, select the fur node, then from the Air Stream menu choose Fur, Hair, Curves -> Assign Fur Surface.

Global Illumination for Fur

Because computing global illumination for large numbers of fur curves can be very slow, Air Stream and the default fur surface shader provide an efficient shortcut that approximates the effects of global illuminaton on the fur. Here's how it works:

First, the global illumination result is computed for the root of the curve being shaded (which will be a location on the base object surface) excluding the fur itself from the global illumination query. The GI result is then modulated by a self-occlusion value based on the distance along the curve from the

curve root, approximating the effect that locations nearer to the curve root are more shadowed by nearby curves.

In effect this approach uses the global illumination result of the underlying surface for the curves. If the base surface is relatively smooth, the global illumination cache can be used for fast and smooth GI rendering.

The self-occlusion approximation can be tuned by adjusting the **Air Self Occlusion** extra attribute, which gives a power applied to the interpolation value along the curve length. Higher values produce darker fur.

To exclude the fur curves from the global illumination computations, Air Stream and the fur instancer shaders do the following:

- Make the fur curves invisible to indirect rays
- Modify the parameter list for the light shader used for occlusion to only sample members of an "occluder" group, and exclude the fur curves from that group. All other objects are included in the occluder group.

When a custom surface shader is used for fur, this global illumination optimization may be undesirable. To disable the optimization, uncheck the **Air Optimize GI** extra attribute.

Texturing Fur Curves

Fur curves can be textured to represent entities other than fur such feathers. The ribbon curve type works best for textured curves.

Each curve has a 2D parametric coordinate system that exactly covers the curve, but this coordinate system cannot be referenced using the standard texture coordinates used by most Maya and Air shaders. When a fur curve is created by the instancer shader, it is assigned a constant pair of standard texture coordinates taken from the underlying surface. The coordinate system within a curve can be referenced using the shading language u and v global variables (instead of the s and t values used for standard texture coordinates).

Air 11.07 and later include some new shaders that work with u,v coordinates:

- The `genTextureUV` generic shader can be used to apply a texture map to a curve using the u,v coordinate system.
- The `genUV` generic shader emits the u and v coordinates in an OutUV parameter that can be connected as the input texture coordinates to any 2D Maya texture, whose pattern will then be applied to the curve surface.

You can add either shader to a Hypershade network using the Air Generic Shader node.

Fur Groups

Fur curves are automatically assigned to a `fur` group and excluded from a `nonfur` group. These group names can be used to control whether fur or non-fur entities are included in a particular output image. E.g., to render only fur to an image, include the `fur` group name in the render pass **Groups** list.

To exclude fur from an image, assign `nonfur` as a global group (in the *Groups and Sets* sub-section of the *Global Defaults* section of the Render tab). Then set the **Groups** parameter for the render pass to `nonfur`.

14.2 Maya Hair

Air Stream can export Maya's paint fx hair for rendering with Air.

The hair geometry is exported as linear or cubic curves. The export process can take a noticable amount of time if there are many hairs or the hairs have many control vertices. Once the hair grooming and animation are locked, you can accelerate the rendering of hair by using Air Stream's archive facility to save the hair geo to an archive for re-use on subsequent renders. See Optimizing Scene Export for more information.

Hair Attributes

Air Stream translates most of the standard hair attributes for rendering. Extra attributes can be applied to customize how the hair is rendered by Air. To add these extra attributes:

- Select the hair node.
- Choose Fur, Hair, Curves -> Add Hair Attributes from the Air Stream menu.

Hair Spline & Curve Type

By default hair curves are rendered as cubic b-spline curves, which apply a smooth interpolation to the control vertices. The **Air Spline Type** extra attribute can be used to select a different interpolation method for the vertices:

`linear`: render each curve a linear curve
`b-spline`: render each curve as a cubic b-spline
`catmull-rom`: render each curve as a cubic catmull-rom spline

Either of the cubic spline types will produce a smoother curve; they may also permit the curve to be represented with fewer segments, saving memory and render time.

By default Air renders each curve as a set of line segments, which is very efficient for thin curves used to represent hair. For wider curves, use the **Air Curve Type** extra attribute to select a different representation:

`ribbon`: render each curve as a quad strip
`tube`: render each curve as a tube

Hair Shading

Air Stream uses the shading attributes on the hair system attached to the hair shape node. The base shading color is exported as a color value for each control vertex. If the hair color is modified, the hair geometry must be re-exported.

The translucence, specular, and specular power shading attributes are applied using the ASM_Hair shader included with Air Stream. Changes made to those shading parameters will not update in an IPR view.

For more control over hair shading, you can assign a custom Air shader. To do that, select the hair shape, then from the Air Stream menu choose Fur, Hair, Curves -> Assign Hair Surface. Changes to a custom surface shader assigned to hair will update in an IPR view.

The Cast Shadows attribute in the Shading section determines whether the hair will cast ray-traced shadows.

Hair Groups

Hair curves are automatically assigned to a `hair` group and excluded from a `nonhair` group. These group names can be used to control whether hair or non-hair entities are included in a particular output image. E.g., to render only hair to an image, include the `hair` group name in the render pass **Groups** list.

To exclude hair from an image, assign `nonhair` as a global group (in the *Groups and Sets* sub-section of the *Global Defaults* section of the Render tab). Then set the **Groups** parameter for the render pass to `nonhair`.

14.3 Curves

Air Stream can export Maya curves as NURB curves primitives for rendering with Air.

Enable Curve Rendering

By default curves will not be rendered. To enable curve rendering, check the **Render curves** control in the Geometry section of the global Render tab.

Curve Shading

Maya curves are not assigned to a shading group. By default they will be shaded using the defaultsurface surface shader when rendered with Air. You can assign an Air surface shader to a curve by selecting the curve and choosing *Fur, Hair, Curves -> Assign Curve Surface* from the Air Stream menu.

Curve Attributes

You can customize how a curve is rendered by Air by assigning extra attributes:

- Select the curve
- Choose Fur, Hair, Curves -> Add Curve Attributes from the Air Stream menu.

The following sections describe the extra attributes for curves recognized by Air Stream.

Curve Rendering

In the Geometry section of the global Render tab you can select the default representation for all curves:

`polyline`: render each curve as a set of line segments
`ribbon`: render each curve as a quad strip
`tube`: render each curve as a tube
`default`: use Air's default curve representation (currently same as polyline)

The render type for an individual curve can be overridden using the **Air Curve Type** extra attribute attached to the curve.

Air renders curves as true 3D entities. Maya curves do not have an assigned width, so Air Stream provides an attribute to set the default width for curves in the Geometry section of the global Render tab. You can override the default width using the **Air Base Width** and **Air Tip Width** extra attributes. If the tip and base widths are different, the width is linearly interpolated along the curve

length.

Maya curves are not visible to reflection rays by default. Use the **Air Reflection Visibility** attribute to enable curves to be seen in reflections and refraction.

Curves are also tagged as single-sided in Maya by default. Use the **Air Sides** extra attribute to enable double-sided curves.

By default each curve will always face the camera. You can specify a fixed orientation using the **Air Curve Orientation** extra attribute:

`face x`: face the X axis
`face y`: face the Y axis
`face z`: face the Z axis

Note that you may need to make a curve 2-sided with the **Air Sides** attribute for the curve to be visible with a fixed orientation.

With Air 12.10 and later, ribbon and tube curves can optionally be tapered at either end. The **Air Tip Taper** and **Air Root Taper** attributes determine where along the curve length the tapering begins. The interpretation of the taper value is influenced by the corresponding taper type:

`width`: the taper value is multiplied by the curve width
`length`: the taper value is multiplied by the curve length
`absolute`: the taper value is the distance along the curve at which tapering begins

Curve Groups

Curves are automatically assigned to a `curves` group and excluded from a `noncurves` group. These group names can be used to control whether curves are included in a particular output image. E.g., to render only curves to an image, include the `curves` group name in the render pass **Groups** list.

14.4 Shave and a Haircut

Air Stream 1.1 and later versions provide native support for rendering hair or fur created with Joe Alter's Shave and a Haircut.

Shave Node Attributes

Air Stream translates most of the relevant Shave node attributes for rendering. Extra attributes can be applied to customize how Shave hair is exported and rendered by Air. To add these extra attributes:

- Select the Shave node.
- Choose Fur, Hair, Curves -> Add Shave Attributes from the Air Stream menu.

Shave Export

Air Stream automatically recognizes Shave nodes and calls the Shave export command to produce an archive for each Shave node that is used for rendering. These archives can be re-used as long as none of the information in the archive has changed. To enable re-use of archives for all nodes in a scene:

- Go to the Export page of the Render Settings window (with Air Stream selected as the current

renderer).
- Under Shave and a Haircut, set the **Shave Export Mode** to `Read archives`.

The export mode can be overridden for an individual Shave node using the **Air Export Mode** extra attribute.

The archive for a node contains curve primitives with the basic geometric information required for rendering. The Renderman section of the Shave globals can be used to add extra data to the curves. By default in Shave 6, the following extra values are exported:

- Per-curve root and tip colors
- Per vertex opacity values

The Renderman section of the Shave globals has three controls for optimizing the export and rendering process:

Binary?

When enabled, curve data is written in a more compact binary format instead of the default text format.

Compress?

When enabled, the output archive is compressed.

Enable voxels?

When enabled, the hair is divided into a 3D grid of voxels (whose size is determined by the Resolution setting), and the curves in each voxel are written to a separate archive. The main archive references each voxel archive as a "delayed read archive" so that a voxel's data is loaded only when and if it is required by the renderer. This option takes longer to export, but it will in general reduce both memory use and render time.

Shave Shading

If a Shave node has a Maya or Air surface shader assigned, Air Stream will use that shader for rendering.

By default a Shave node does not have a surface shader assigned, and the shading attributes on the Shave node are used for rendering. Air Stream applies the ASM_Shave shader to the node, which has the following tuneable parameters:

Self Shadow
Specular
Specular Tint
Secondary Specular
Gloss
Amb/diff
Second Gloss
Bend Tangent

These attributes can all be tweaked in an IPR rendering.

The root and tip colors are written as part of the archive data by Shave (one per curve). The archive must be re-written If changes are made that affect those colors.

The Air distribution includes two surface shaders for hair and fur that may be useful when rendering shave nodes: VFur and VHair. The VFur shader is based on a SIGGRAPH 2000 presentation on fur rendering in the movie Stuart Little. The lighting model used in VFur was developed to allow the shading of fur to closely mimic that of the underlying surface. To do that, the normal vector used for shading is based on the normal at the current location on the curve and the surface normal at the curve root. For that computation to work correctly, the curve archive exported by Shave must included the surface normal at the root location of each curve. You can enable that export using the **Export Root Normal** extra attribute.

Note: The Shave documentation describes a default shading mechanism for rendering with a RenderMan renderer that relies on writing a simple shader declaration into the exported archive file. Air Stream does not use that mechanism (and in fact uses a RIB filter to disable any Surface shader in the archive).

Global Illumination for Shave Fur

Because computing global illumination for large numbers of fur curves can be very slow, Air Stream provides an optimized global illumination solution for short fur. To enable this optimization, apply the Air Stream extra attributes to the Shave node, and check the **Air Optimize GI** control. Here's how the optimization works:

First, the global illumination result is computed for the root of the curve being shaded (which will be a location on the base object surface) excluding the fur itself from the global illumination query. The GI result is then modulated by a self-occlusion value based on the distance along the curve from the curve root, approximating the effect that locations nearer to the curve root are more shadowed by nearby curves.

In effect this approach uses the global illumination result of the underlying surface for the curves. If the base surface is relatively smooth, the global illumination cache can be used for fast and smooth GI rendering.

The self-occlusion approximation can be tuned by adjusting the **Air Self Occlusion** extra attribute, which gives a power applied to the interpolation value along the curve length. Higher values produce darker fur.

To exclude the fur curves from the global illumination computations, Air Stream and the fur instancer shaders do the following:

- Make the fur curves invisible to indirect rays
- Modify the parameter list for the light shader used for occlusion to only sample members of an "occluder" group, and exclude the fur curves from that group. All other objects are included in the occluder group.

Shave Motion Blur

To render motion blurred hair or fur:

- Make sure motion blur is enabled on the Render tab of the Render Settings window.
- Go to the Renderman section of the Shave globals.
- In the motion blur section, set **Inherit Settings** to `Off`.
- Check **Enable Motion Blur**.
- Set **Shutter Times** to `Relative`.
- Set the **Shutter Open** and **Close Offsets** to the same values as the **Shutter Open / Close** attributes in the Air motion blur settings on the Render tab.

Shave Curve Type

The default representation for curves is set in the Geometry section of the Render tab. Use the **Air Curve Type** extra attribute to select a different representation:

`polyline`: render each curve as a set of line segments
`ribbon`: render each curve as a quad strip
`tube`: render each curve as a tube

Shave Groups

Shave curves are automatically assigned to a `shave` group and excluded from a `nonshave` group. These group names can be used to control whether Shave or non-Shave entities are included in a particular output image. E.g., to render only Shave hair to an image, include the `shave` group name in the render pass **Groups** list.

To exclude Shave hair from an image, assign `nonshave` as a global group (in the *Groups and Sets* sub-section of the *Global Defaults* section of the Render tab). Then set the **Groups** parameter for the render pass to `nonshave`.

See Also:

Shave and a Haircut web site: www.joealter.com

15 Fluids

Air can render Maya fluids as volumes or surfaces.

Fluids as Volumes

Air Stream exports a volumetric fluid shape as a true volume primtive for rendering with Air.

Global volume controls can be found in the Volume Rendering section of the Render tab. Here is a brief discussion of the volume controls:

Air supports two methods of volume rendering:

Voxels

In the voxel method the volume space is shaded using an adaptive 3D shading grid that decouples shading from visibility calculations. The maximum size of each voxel (in pixels) can be set with the **Voxel Width** attribute. Larger voxels render more quickly but produce a coarser final result. The 3D shading cache can use a considerable amount of memory, but it generally renders faster than the ray marching method described below.

Ray marching

The volume is evaulated by "marching" along the query ray through the volume, sampling the volume at discrete intervals. The distance between samples is set with the **Ray Marching Step Size** attribute. Larger step sizes render more quickly.

Air Stream allows a volume's opacity to be tuned independently for viewing rays (from the camera or reflections) and shadow rays using the **View Density** and **Shadow Density** controls.

Fluids as Surfaces

If the outMesh plug of a fluid shape is connected, Air Stream assumes the fluid is represented by a separate mesh entity for rendering and the fluid shape itself is not exported. Otherwise, Air Stream exports the fluid shape as an Air implicit surface primitive. The fluid shape must have a density grid - static or dynamic - for the implicit primitive to render properly.

Fluid Primitive Data

Air Stream attaches the following primitive data (when available) to an exported Volume or Implicit primitive:

Maya data	RIB variable name
density	`fluid_density` (volume) or `density` (surface)
temperature	`fluid_temperature`
fuel	`fluid_fuel`
pressure	`fluid_pressure` (only exported if used by the color, opacity, or incandescence ramps in the fluid shape)
color	`Cs`

See Also:

Air User Manual: Volume Rendering
Air User Manual: Primitives -> Implicit Surfaces

16 Massive Crowds

Crowds created with Massive can easily be added to a scene using the MassiveAgents instancer shader included with Air.

Preliminary

Massive uses a procedural primitive to generate agents at render-time. Air must be able to find the procedural primitive in order to render the agents. If you have Massive installed on the machine running Maya, the procedural primitive will be located in `$MASSIVE_HOME/bin`. Air Stream automatically adds that directory to the search path for procedural primitives. If the Massive procedural is located elsewhere, you will need to add that directory to the procedural search path on the Export tab of the Air Stream Render Globals.

There are two similar methods for adding agents:

Method 1: Agents on terrain

If all agents are all in the vicinity of a single terrain object, you can include agents in the scene by attaching the MassiveAgents instancer shader to the terrain object:

- Select the terrain object.
- In the Air Stream menu, select Instancer Shader -> Create New Instancer Shader.
- In the new Air Stream Instancer window, click the button next to the **Shader** name, and load the MassiveAgents instancer from the shaders directory of your Air installation.
- In the Attributes section, set the **Region bound** to the maximum distance of any agent from the

terrain object. (E.g., if the agents are walking on the terrain and are 2 units high, set the region bound to 2.)
- In the Shader Parameters list, set the Agent Archive Prefix string to the full path to the individual agent archives, minus the trailing frame number and .rib extension. The agent archives must have a 4-digit frame number.
- Set the RenderPass parameter to the name of the desired Massive render pass.
- If motion blur is required, set the Motion Blur parameter to 1. You will also need to enable one of the motion blur options on the Air Stream Render tab in the Render Globals dialog.
- The Frame Offset parameter can be used to offset the agent archive number relative to the Air Stream frame number. E.g., a Frame Offset of 10 would load the agent archive #11 for render frame 1.
- When the Inherit Transform parameter is enabled with a value of 1, the agents will be affected by any translation, rotation, or scaling applied to the terrain.
- Render with Air Stream to see the scene with Massive agents.

Method 2: Agents in a box

This method uses a standin box for the agents as the basis for the MassiveAgents instancer shader:

- Create a polygonal box big enough to contain all the agents in the scene.
- Select the new box.
- In the Air Stream menu, select Instancer Shader -> Create New Instancer Shader.
- In the new Air Stream Instancer window, click the button next to the **Shader** name, and load the MassiveAgents instancer from the shaders directory of your Air installation.
- In the Attributes section, check **Hide base object** to make the box invisible in the rendering.
- In the Shader Parameters list, set the Agent Archive Prefix string to the full path to the individual agent archives, minus the trailing frame number and .rib extension. The agent archives must have a 4-digit frame number.
- Set the RenderPass parameter to the name of the desired Massive render pass.
- If motion blur is required, set the Motion Blur parameter to 1. You will also need to enable one of the motion blur options on the Air Stream Render tab in the Render Globals dialog.
- The Frame Offset parameter can be used to offset the agent archive number relative to the Air Stream frame number. E.g., a Frame Offset of 10 would load the agent archive #11 for render frame 1.
- When the Inherit Transform parameter is enabled with a value of 1, the agents will be affected by any translation, rotation, or scaling applied to the standin box.
- Render with Air Stream to see the scene with Massive agents.

Customizing the MassiveAgents Shader

Source code to the MassiveAgents instancer shader is included in the Air distribution in:

```
$AIRHOME/shaders/src/instancer
```

If you are familiar with shading language programming, you can adapt the shader to your workflow by changing default values, adding new parameters to apply additional attributes to the agents, etc.

See Also:

Air Stream Instancer Shader node
Air User Manual: Massive
Air User Manual: Shaders -> Instancers -> MassiveAgents

17 Render Globals

When Air is selected as the current Maya renderer, the following pages appear in the Maya Render Globals window:

17.1 Common Tab

The controls in this tab mirror those of the Maya software renderer.

Color Management

Air Stream does not use the controls in this section.

File Output

Air Stream does not use the controls in this section. Use the Images tab to specify the images to render with Air.

Frame Range

Use these controls to set the range of frames to render for an animated sequence (using the Render Sequence option in the Air Stream menu).

To enable the controls in this section, set the **Frame/Animation ext** control in the File Output section to an option that includes a frame number.

Air Stream does not support the option to renumber frames.

Renderable Cameras

Renderable Camera

Choose the camera to use for rendering.

Image Size

Use these controls to specify the size of the rendered image.

Render Options

Enable Default Light

When enabled, Air Stream automatically adds a default light when exporting a scene with no lights.

MEL scripts

Specify MEL scripts to run before and after rendering.

17.2 Render Tab

Image Quality

Primary Hider (Scanline 'Scanline + micopolygons' 'Ray trace - no shading cache')

Sets the rendering and shading methods used for primary visibility in the scene (camera rays):

`Scanline`

This select's Air's default highly efficient scanline visible surface determination. An image-space based shading cache allows shading samples to be shared by different pixel samples. Memory used by an object that does not need to be ray traced is freed as soon as the object has been rendered.

`Scanline + micropolygons`

The same basic visibility algorithm as the above scanline option, but instead of an image-space shading cache, polygons are diced into small "micropolygons", and each micropolygon is shaded and then rasterized. This mode may produce smoother shading particularly for highly reflective objects.

`Ray trace`

In this mode ray tracing is used to determine visibility from the camera. There is no shading cache; each pixel sample is shaded separately. The lack of a shading cache makes this method slower than the above two options, but it can produce higher quality images and capture effects such as motion-blurred shading that are not possible with the scanline methods.

Shading Samples (default 1)

The frequency with which an object's shading is computed, given as the number of shading samples per pixels. This setting is the inverse of the RIB shading rate attribute.

Pixel Samples (default 4)

The Pixel Samples setting specifies the number of times the scene geometry is sampled at each pixel as a M x M grid. For example, a Pixel Samples value of 4 specifies 4x4=16 samples per pixel. Use more pixel samples to smooth jagged edges. Because geometric sampling is independent of shader sampling, increasing the number of pixel samples will not usually result in a large increase in rendering time.

Edge mask (default off)

When enabled, Air computes a high-resolution sub-pixel edge mask to provide better smoothing of surface edges and small or thin objects.

Filter name and width (default gaussian, 2)

The pixel filter and support width determine how the pixel samples are combined to form the final output values at each pixel. The default gaussian filter with a support width of 2 produces a "soft" filtered result. For a sharper image, try using the mitchell filter with a width of 4.

Clamp to alpha before filtering (default off)

When checked shading results for each pixel sample are clamped to be no greater than the alpha/transparency value (and no greater than 1 in any case). This option can help to smooth shading in images using HDR (high-dynamic range) textures or environment maps which may result in shading values that are much greater than 1. Note that when this option is enabled, the resulting output values for every pixel will be less than or equal to 1.

Geometry

Surface Tolerance (default 0.5)

Specifies the tolerance for surface tessellation as the maximum allowed difference (in pixels) between the tessellated surface and the true surface. Higher values use less memory and render more quickly.

Tolerance Measure (Pixels 'Object space')

Defines how the surface tolerance for mesh tessellation is measured. The default Pixels value measures tolerance based on the view-dependent projection of the difference between the true surface and the tessellated mesh. When object space is selected, the tolerance value gives an absolute maximum allowed difference in object space, and the tessellation is not view dependent.

Render curves (default off)

Whether to render NURB curves.

Curve Render Type (default polyline ribbon tube)

`polyline`: render each curve as a set of line segments

`ribbon`: render each curve as a quad strip

`tube`: render each curve as a tube

`default`: use Air's default curve representation (currently same as polyline)

Point Render Type (Hexagon Disk Sphere)

This is the geometric representation used to render particles with render type Points or MultiPoint.

Points per Cluster (5000)

Air can automatically subdivide large point sets into smaller clusters for more efficient rendering. This attribute sets the maximum number of points in a single cluster.

Motion Blur

Motion Blur Mode ('Motion Blur Off' 'Camera Only' Transform Deform 'Transform & Deform')

Specifies the types of motion blur (if any).

Motion Samples (default 2)

Number of time positions to use to specify the blurred motion.

Shutter Open/Close (default 0, 0.5)

Specifies the shutter open and close times as a fraction of the unit interval.

Include motion blur when ray tracing (default off)

When enabled, the ray tracer includes the effects of motion blur when tracing rays, producing more accurate results at the expense of longer render times.

Reduce quality for blurred objects (default off)

When enabled, Air uses a coarser tessellation for objects that are heavily motion blurred, reducing render time and memory use.

Quality Reduction Factor (default 1)

Quality control used when the above option is enabled. Larger values produce a faster, but lower quality result.

Volume Rendering

Volume Method (Voxels 'Ray marching')

Specifies the method used to evaluate volume primitives. The default voxel mode creates a 3D grid of shading samples that are analytically integrated to determine visibility and color. This mode effectively decouples shading from visibility determination. The shading cache can use a lot of memory. Ray marching does not use any additional memory but may be slower.

Voxel Width (pixels) (default 1)

Approximate size of each voxel in pixels used for the voxel method of volume rendering. Larger values render more quickly but produce a coarser result.

Ray Marching Step Size (default 1)

Step size (in world units) for ray marching volume primitives.

View Density (default 1)

Multiplier for volume opacity applied when evaluating camera and reflection rays.

Shadow Density (default 1)

Mulitplier for volume opacity applied when tracing shadow rays.

Optimization

Cull backfaces (default off)

When enabled objects are rendered as single-sided entities and back-facing polygons are hidden.

Optimize memory for unreflected objects (default off)

When enable, Air attempts to reduce the data retained for objects that need to be ray traced but do not need to have their shaders evaluated by traced rays. This optimization can reduce the memory used by objects that are visible to shadow rays (unless the surface shader modifies the surface opacity) and objects visible to indirect rays (as long as the indirect shade mode does not require shader evaluation).

Render Control

Render Command (air)

Command to execute when rendering. Under Windows, set this value to `air64` to render with 64-bit Air.

Bake Command (bakeair)

Command to execute when baking maps. Under Windows, set this value to `bakeair64` to bake with 64-bit BakeAir.

Threads (default 0=all)

Specifies the number of threads to use for rendering. When set to 0, all detected CPUs will be used.

Run at low priority (Windows only) (default on)

When enabled, the Air Stream render preview and render current frame commands will run at lower priority under Windows, allowing Maya and other foreground applications to continue to run unimpeded. The render sequence and render shadow map commands are always started at low priority under Windows.

Tile Order (default columns rows spiral)

Specifies the order in which image tiles are rendered. The default order is the same as the row order.

Tile Size (default 32)

Air renders an image in small tiles, and this property sets the width and height of each tile. Larger tile sizes usually render more quickly but also use more memory.

Custom Options and Attributes

Use the edit box in this section to enter RIB commands for custom options and attributes. These custom commands will be included in the exported scene file just prior to the WorldBegin statement.

Groups

Global Groups

A space-separated list of group names to which all objects belong by default. Group

membership for an object can be modified in the Groups and Sets section of an Air Stream node.

See Also:

Air User Manual: Image Quality
Air User Manual: Options
Air User Manual: Attributes

17.3 Images Tab

Primary Image

Use the controls in this section to specify the properties of the primary output image.

Image File Name (default: use scene name)

Base name of the rendered image.

File Format (default Open EXR)

Specifies the file format for the rendered image.

Channels (default rgba)

Specifies the data to be saved in the output image.

Quantization (8-bit 16-bit float)

Specifies the precision of the output data that is saved.

Groups (default empty)

If not empty, a list of groups (separated by spaces) to include in the output image. Any objects not in any of the groups will be treated as matte objects for the rendered image.

Extra Parameters

Enter any additional parameters to pass to the display driver. Additional parameters should include a full type declaration for the parameter name and a corresponding value. E.g.,

```
"string description" "test 1"
```

Additional Images

Use the **Create** button to make additional images to be included in the renderer's output. Additional images can be saved to separate files or embedded in the primary image (in some file formats).

Show additional images in preview (default on)

When checked, additional passes appear in a preview rendering (embedded in the primary image, they show up as extra layers in the Air Show framebuffer).

Exposure

Custom exposure (default off)

When checked, the following gain and gamma values are exported as a RIB `Exposure` statement.

Gain (default 1)

Specifies a custom gain (multiplier) applied to the rendered data values prior to quantization.

Gamma (default 1)

Specifies a custom gamma value applied to the rendered data values prior to quantization.

Crop Window

Use crop window (default off)

When enabled, only the rectangular area defined by the following crop window is rendered.

Crop Window (xmin xmax ymin ymax)

Defines a sub-rectangle of the output image. Each parameter should be in the range 0 to 1. Air Show provides a simple method of selecting a crop window: click and drag a rectangle in an image in Air Show, then select Copy Crop Coordinates from the Air Show Edit menu.

Depth

Remap depth range (default off)

When enabled, depth values (z) are remapped to the unit interval based on the min and max values supplied below.

Z min (default 0)

Depth value to be remapped to 0.

Z max (default 10000)

Depth value to be mapped to 1.

Light Channels

These options affect light output channels:

Reflect light channels (default off)

When enabled, a light channel includes the light's contribution to reflected and refracted surfaces. When saving light channels that include reflections, you may also wish to save the Environment channel which holds the contribution of the background environment to reflections and refraction.

Bounce light channels (default off)

When enabled, a light channel includes the indirect contribution of the light to the scene. When

this mode is enabled, the indirect light source channel (channel 0) will hold the environment contribution to indirect lighting.

See Also:

Air User Manual: Output

17.4 Export Tab

RIB Export Options

Archive Mode ('Off' 'Write' 'Read')

The archive mode selects how geometric shape data is exported for rendering:

Off (Shapes in Main File)
All shape data is written to the main scene file.

Write Shapes to Archives
Each shape is written to a separate file, which is then referenced by the main scene file. Existing archives, if any, are overwritten.

By default Air Stream assumes all objects are non-deforming and writes only a single archive which is used for all frames in a sequence render. For shapes that change during an animation, assign an Air Stream Node and enable per frame archiving by setting the Archive Frequency attribute to Per Frame.

Read Shapes from Archives
The main scene file references shape archives that have been created by the Write archive mode. If an archive is missing for a particular shape, the archive will be created. Existing archives are assumed to be valid and are used as-is.

The archive mode for a particular object can be overridden by applying an Air Stream Node and selecting an archive mode in the RIB Export section.

Archive Read Method (Read always 'Read when object bound is hit')

The archive read method determines how archives are referenced by the main scene file:

Read always
Archives are always read into the scene before rendering begins. Archives are referenced using the RIB `ReadArchive` command.

Read when object bound is hit
The archive for a shape is read when and only if the bounding box for the shape is encountered by the renderer. This mode can greatly increase the efficiency of rendering for complex scenes by delaying the inclusion of geometry until it is actually needed. Objects that are invisible - either off-screen or obscured by other objects - need not be processed at all. Time-to-first-pixel is also reduced because archives are read when they are needed instead of before rendering begins. Archives are referenced using the RIB `DelayedReadArchive` procedural primitive.

The read method for a particular object can be overridden by applying an Air Stream Node and overriding the read method in the RIB Export section.

File name (default scene name)

Specifies the name of the exported RIB file or base name for multiple RIB files. This field recognizes two variables:

`$SCENE` will be replaced by the current scene name.
`$Fn` will be replaced by the current frame number padded to *n* digits.

Output root

The path to the Air Stream export directory for the current file. By default, this directory points to a directory called AIRStream inside the current Maya Workspace.

Inherit from output root (default on)

If checked, all the subsequent directories will be subdirectories inside the output root. If unchecked, the paths can be set to any location that's writable for the current user.

RIB

Path to the directory where AIR Stream will write RIB files. By default it is a `ribs` sub-directory in the output root directory.

Archive

Path to the directory where AIR Stream will write Archive RIB files. By default it is a `archives` sub-directory in the output root directory.

Temp

Path to the directory where AIR Stream will write temporary files. By default it is a temp sub-directory in the output root directory. Currently this directory is used for writing the RIB files for scene and shader previews.

Search Paths

Use the controls in this section to add directories to the search paths used by Air to find files. Directories should be separated by colons (:) and should use Unix-style forward slashes to separate path components.

Network Cache

Enable local cache of network files (default off)

When enabled and a cache directory is provided below, Air will cache resource files such as texture maps and point clouds in a local directory for faster access.

Cache Size (1024 mb)

Maximum size of the network file cache in megabytes.

Cache Directory

Local directory to use to cache network files. The directory should be dedicated to this purpose.

Path Mapping

Enable render-time path mapping (default off)

When enabled and prefixes are provided below, Air will translate file reference path prefixes depending on the environment of the machine used for rendering.

Windows Prefix

Path prefix for Windows machines.

Linux Prefix

Path prefix for Linux machines corresponding to the Windows prefix provided above.

Shave and a Haircut

Shave Export Mode (Off Update archives 'Read archives')

Global control for how Shave nodes are exported for rendering. When set to `Off`, no Shave nodes are rendered. When set to `Update archives`, a separate archive is created for each Shave node with the curve data required for rendering, and those archives are used for rendering. If the mode is `Read archives`, any existing archives are re-used when rendering.

The export mode can be overridden for an individual Shave node by using the Shave Export Mode extra attribute on the Shave node. (The extra attribute can be added from the Air Stream menu using Fur, Hair, Curves -> Add Shave Attributes.)

17.5 Shading Tab

Ray Tracing

Trace Offset (default 0.05)

The trace offset is a small distance used to shift the starting location when rays are traced for reflections or shadows. Increase this setting to prevent rendering artifacts due to incorrect self-intersection of a ray with the surface from which it is cast.

Enable traced shadows (default on)

When enabled, objects are retained for possible intersection with shadow rays.

Enable traced reflections (default on)

When enabled, objects are retained for ray tracing by reflection rays.

Max Trace Depth (default 6)

This setting gives the maximum number of bounces through which to trace reflection and refraction rays.

Maya Reflection Rays (default 1)

This controls specifies the default number of rays to trace for Maya surface shaders with reflections.

Maya Refraction Rays (default 1)

This controls specifies the default number of rays to trace for Maya surface shaders with refraction.

Displacement

Displacement Mode (Bump only 'True displacement' 'Mesh displacement')

This property sets the default method of applying displacement shaders. The displacement mode can be overridden on a per-object basis using an Air Stream node.

Displacement Bound (default 0.1)

Sets the maximum distance a displacement shader moves the surface in object space. The displacement bound can be overridden for a particular object using the corresponding displacement bound control in the Displacement section of an Air Stream node.

Environment Shaders

Environment shaders shade rays that miss all objects in the scene. Use the Create Environment button to add an environment shader node to the scene. To display an environment node for editing, double-click the node name in the list window. Use the Delete Environment button to remove a selected environment node.

Mode (Enabled 'Disabled for preview & IPR' Disabled)

Whether to export environment shaders when rendering.

Use cache for faster global illumination (default on)

When enabled Air 11.06 and later will use an internal cache to accelerate environment shader queries for global illumination. The cache generally produces smoother results with faster rendering times.

Imager Shaders

Imager shaders can be used to modify the final pixel values in a rendered image. Use the Create Imager button to add an imager shader node to a scene. To display an imager node for editing, double-click the node name in the list window. Use the Delete Imager button to remove an unwanted imager node.

Mode (Enabled 'Disabled for preview & IPR' Disabled)

Whether to export imager shaders when rendering.

Atmosphere Shaders

Atmosphere shaders can be used to add volumetric effects such as smoke or fog to a rendered image. Use the Create Atmosphere button to add a volume shader as the atmosphere for all objects in a scene. To display a volume node for editing, double-click the node name in the list window. Use the Delete Atmosphere button to remove an unwanted imager node.

Mode (Enabled 'Disabled for preview & IPR' Disabled)

Whether to export atmosphere shaders when rendering.

Subsurface Scattering

SSS File Mode (Off Write Read)

The default file access mode for subsurface scattering point sets. This setting can overridden with the SSS File Mode control in an Air Stream node.

Textures

Convert Maya images to Air textures (default on)

When enabled Air Stream converts any image files referenced in a Maya File Texture node to an Air texture map at export time. Air texture files store multiple resolutions of an image in a tiled format that allows Air to provide high quality filtering and management of large numbers of images efficiently. Converted files are stored in the textures directory specified on the Export tab. If the source image is modified, the image will be reconverted on the next export. Note that using this option with the following option to convert from sRGB color space requires Air 13.02 for accurate results.

Convert textures from sRGB color space (default off)

When enabled, Air will automatically convert raw texture files from sRGB color space to linear color space. This option requires Air 12.10 or later. It does not affect textures that have already been converted to Air texture files using mktex or mktexui.

Resize Mode (Nearest Power of 2 'Up to Power of 2' 'Down to Power of 2' 'None')

This option determines how a source image is resized when it is converted to an Air texture map.

Automatic MIP Mapping (default off)

When enabled, Air automatically generates mip-maps on demand for image files used as textures that have not been converted to Air texture maps. Requires Air 13.02 or later.

Texture Cache (32 mb)

Texture cache size for Air's tiled texture files. In Air 13.01 and later a single cache is shared by all Threads. In earlier releases of Air, a separate cache is used for each rendering thread.

Auto Shadow Maps (40 mb)

Cache size for Air's automatic shadow maps (used for non-disk-based Maya shadow maps).

Ptex Cache (100 mb)

Cache size for Ptex textures.

Spectral Lighting

Enable Spectral Prefiltering (default off)

When enabled, spectral color data is pre-multiplied by the dominant illuminant spectrum and the white point is adjusted for the display illuminant during conversion to rgb values for rendering.

Dominant Illuminant (default incandescent tungsten)

The file name of a spectral profile for the dominant source(s) of illumination in a scene.

Display Illuminant (default CIE D65 standard)

The file name of a spectral profile corresponding to the display conditions for the resulting image. For a normal image viewed on an sRGB device, the default D65 profile is appropriate.

Filter RGB colors (default on)

When enabled, spectral prefiltering is also applied to rgb color shader parameters and to color textures by shaders that convert rgb color texture values to "current" color space.

See Also:

Air User Manual: Reflections -> Ray Trace Reflections
Air User Manual: Shaders -> Environments
Air User Manual: Shaders -> Imagers
Air User Manual: Shaders -> 2D Texture Mapping
Air User Manual: Lighting -> Spectral Prefiltering and Chromatic Adaptation

17.6 Global Illumination Tab

Use the controls on this tab to enable and configure the global illumination rendering options for Air.

Global Illumination Mode (Off Indirect Occlusion)

The GI mode detemines the method used to compute global illumination effects:

Off
Global illumination effects are not computed.

Indirect
Air casts indirect rays to sample the incoming light at a surface.

Occlusion
Air casts shadow rays to estimate the extent to which the current shading location is blocked or occluded from the surrounding environment. The occlusion result can be combined with an environment map to simulate illumination due to the surrounding environment.

Use cache (default off)

Check this control to enable the use of an irradiance cache for indirect illumination or an occlusion sample cache for occlusion.

Sample Quality

Rays per Sample (default 64)

Number of rays to cast to compute each global illumination sample.

Max Trace Distance (default 10000)

Maximum distance to trace rays used to compute global illumination. Shorter distances render more quickly.

Adaptive ray casting (default off)

When enabled, Air casts fewer rays initially, and casts additional rays only in regions where it detects high variation in the incoming illumination or occlusion.

Cache Properties

To accelerate global illumination, Air allows GI samples to be saved to an internal cache and re-used at nearby shading locations. The properties in this section allow you to configure a cache for indirect illumination or occlusion.

Max Error Allowed (default 0.2)

Sets a maximum error threshold when re-using samples in the cache. If a result cannot be obtained from cached values within the given threshold, a new sample will be generated.

Max Sample Spacing (default 20 pixels)

Gives a maximum distance in pixels between samples in the cache. This parameter ensures that even flat sections in an image receive some GI samples. This setting should be increased for larger image sizes.

Cache File

Air allows the contents of the GI cache to be saved to a file and re-used to accelerate future renderings. The cache file name is relative to the directory containing the main rib file. Check the Read or Write boxes to, respectively, read from or write to the cache file.

Indirect and occlusion GI modes use different cache formats.

Use pre-pass (default on)

When enabled, performs a separate pre-pass to generate GI samples before final rendering begins. Using the pre-pass generally improves the quality of rendering with the GI cache by smoothing out boundaries between regions covered by each cache sample.

Pre-pass Multiplier (default 0.5)

When rendering a pre-pass, Air applies this multiplier to the **Max error allowed** and **Max sample spacing** values. Using a multiplier less than 1 causes more samples to be taken during the pre-pass, generally producing a smoother appearance in the final image.

Tweak Result

Tint (default white)

Color tint applied to the computed global illumination result.

Scale Intensity (default 1)

Multiplier for the computed global illumination result.

Baked Maps

Use Baked Maps For (Nothing Indirect Occlusion 'Occlusion + Bent Normal' 'Occlusion + Environment')

BakeAir can be used to bake per-object maps with global illumination information. To use such baked maps, select the appropriate entry corresponding to the values that were baked.

Baked Map File Type (Air Texture Map)

Select the file format used for the baked maps with global illumination data.

IPR Settings

Use the controls in this section to customize global illumination settings for Interactive Preview Rendering.

Global Illumination Mode (Inherit Off Indirect Occlusion)

Rays per Sample (Inherit 4 16 32 64 256 512 1024)

Adaptive Sampling (Inherit Enable Disable)

Use Cache (Inherit Enable Disable)

Cache Pre-pass (Inherit Enable Disable)

Cache File Mode (Inherit Disable Read Write 'Read and Write')

Occlusion Environment (Inherit Accurate Approximate)

Indirect Only

The properties in this section only affect the behavior of the indirect global illumination mode.

Max bounces (default 1)

Gives the maximum number of "bounces" to follow indirect light rays. More bounces take longer to render. When using indirect lighting without a cache, rendering with more than 1 bounce may take a long time.

Indirect shading (Constant Lambert Shader)

The indirect shading mode determines how Air computes the contribution from surfaces that are intersected by indirect rays:

Constant
The object's color attribute value is used as the indirect shading result. This is the fastest mode since it does not require evaluating any shaders or illumination.

Lambert
Air estimates the indirect contribution by computing a Lambert (diffuse) illumination result and

multiplying it by the object's color attribute. This mode is faster than the Shader mode because the object's surface shader does not have to be evaluated.

Shader
Air evaluates the surface shader assigned to the object to compute the indirect illumination contribution.

Re-use shading (default off)

When enabled, Air caches the shading result computed when an indirect ray hits a surface. If that surface is intersected again, the shading result may be re-used instead of computing a new shading sample.

Radiosity Size (default 0.0)

Sets the maximum area (in world space units) for any one shading sample in the radiosity cache. Surfaces that are larger than this maximum size will store a grid of shading samples instead. Smaller values produce more accurate results at the expense of longer rendering time and more memory use by the cache. The special value 0 results in exactly 1 radiosity sample per polygon face.

Occlusion Only

Environment Sampling (Accurate Approximate)

Selects the method used to sample the surrounding environment in occlusion mode:

Accurate (once per unoccluded ray)
The environment is sampled at each unoccluded ray. This method produces the most accurate results. It will be slower to render when the occlusion cache is not used, but it can be faster than the approximate method when the cache is used because the environment result will be saved with the cache.

Approximate (once per sample)
The environment is sampled only once for each global illumination query. The environment is sampled over a hemispherical region centered around the average unoccluded direction.

Shadow Color (default black)

Shadow color used to tint occluded areas.

See Also:

Global Illumination
Air User Manual: Lighting -> Indirect Lighting
Air User Manual: Lighting -> Ambient Occlusion

17.7 IPR Tab

Use the controls on this tab to configure Interactive Preview Rendering (IPR) with TweakAir.

IPR Quality

IPR Shading Mode ('Diffuse only' 'Fast re-lighting' 'Full shading')

This property determines how objects are shaded in the interactive preview. The shading mode for an individual object can be set by attaching an Air Stream node and selecting a mode under the IPR section of that node.

Diffuse only
Objects are shaded using a simple diffuse (lambert) surface shader. The fastest shading option.

Fast re-lighting
The fast re-lighting mode provides fast re-rendering of most surface shaders that fit a basic shading model: a diffuse component, a specular component, and possibly reflections or incandescence. TweakAir is able to decouple lighting computations from surface shader evaluation for these shaders. After the initial image generation, TweakAir can compute the effects of lighting changes without re-evaluating the surface shader for an object.

Full shading
Shaders are re-evaluated in full after any change.

Smooth edges (default off)

When enabled, TweakAir computes coverage information at each pixel to produce a smoother final image, using more memory and taking longer to refresh.

Max Shadow Rays (default 4)

Maximum number of shadow rays to trace for each shadow sample, regardless of the samples setting in the light shader.

Max Reflection Rays (default 4)

Maximum number of reflection rays to trace for each reflection or refraction sample, regardless of the samples setting in the shader. Allowing more rays produces more accurate results at the expense of longer refresh times.

IPR Process

Update reflections (default on)

When enabled, an object with reflections will be updated whenever a reflected object's shading changes.

Update camera (default on)

When enabled, changes to the (first) Maya renderable camera are sent to TweakAir. When the camera view is changed, the entire preview region must be re-computed.

Update materials (default on)

When enabled, changes made to Maya and Air surface and displacement shaders are sent to TweakAir.

IPR Process

Image Quantization (8-bit 16-bit float)

Specifies the precision of the image data.

Threads (default 0=all)

Number of rendering threads to use for the IPR process.

Run at low priority (Windows only) (default off)

When enabled, the IPR process run at lower priority under Windows, allowing Maya and other foreground applications to continue to run unimpeded.

Show command window (default off)

Whether to show the command shell executing the separate TweakAir process for IPR.

See Also:

IPR - Interactive Preview Rendering with TweakAir

17.8 Bake Tab

The Bake Tab settings control the baking of texture maps using BakeAir.

Bake Mode ('Update maps' 're-use existing maps')

Selects whether bakeair should re-bake all maps or preserve those that exist and only bake missing maps.

Export map name as user attribute (off)

When enabled, Air Stream exports a user attribute with the baked map name for each object. The user attribute is a string attribute with the same name as the base name specified below. A shader can query the user attribute to retrieve the map name so that the texture can be used in a shading network.

Note that this export switch does NOT need to be enabled when using maps baked with one of the GI bake modes. Instead, select the appropriate map mode in the Baked Map section of the Global Illumination tab.

Map File

Base Name (bakedmap)

Base name for the baked maps. The map name for each object has the DAG path appended to the base name, followed by a file extension based on the file type selected below. The maps are generated in a per-project subdirectory of the texture output path specified on the Export tab.

The GI bake modes ignore the base name specified here and instead use a fixed name for each mode:

GI Bake Mode	Base Name
Indirect Illumination	`indirectillum`
Occlusion Only	`occlusion`
Occlusion + Bent Normal	`occnrm`
Occlusion + Environment	`occenv`

File Type (Air Texture Map)

Selects the file format for the baked texture maps. The Air Texture Map format is recommended for maps that will be used for re-rendering with Air.

Quantization (8-bit 16-bit float)

Selects the precision for the saved data.

Bake Channels (rgb)

Select from a list of standard bake channels. To bake a custom set of channels, select Custom and enter the channel names in the Custom Channels control below.

Here are the standard channels and the corresponding variables that are baked:

Channel Preset	Channel Count	Display Channels
rgb	3	`rgb`
Unlit Color	3	`color __constant`
Diffuse	3	`color __diffuse`
Diffuse + Unshadowed	6	`color __diffuse,` `color __diffuse_unshadowed`

Diffuse + Unshadowed + Indirect	9	`color __diffuse,`
		`color __diffuse_unshadowed,`
		`color __indirect`

GI: Indirect Illumination	3	`color __indirect_illum`
GI: Occlusion Only	1	`r`
GI: Occlusion + Bent Normal	4	`r,normal Nunoccl`
GI: Occlusion + Environment	4	`r,color __occenv`

The GI occlusion modes use the occlusionpass surface shader for each baked object.

Custom Channels

Comma-separated list of channels to bake when the Bake Channels preset is set to Custom. Each channel should include an inline type declaration.

Max Map Size (512)

Maximum width and height of any one bake map. The actual map size for each object is computed from the object geometry and texture mapping based on the Pixel Size value.

Pixel Size (0.1)

Approximate width of one pixel in a baked map when applied to the object's surface. Air Stream uses this value to compute a width and height for each baked map based on the surface area and texture coordinates of the object. Smaller values produce larger maps with more detail.

Border Fill (1)

A baked map may have empty pixels when the texture mapping does not cover the unit square in texture space. Texture interpolation and filtering may cause artifacts when sampling near empty

pixels in the baked map. To help prevent such artifacts, BakeAir can fill in empty pixels along the edge of the rendered region in a baked map. This setting gives the width of the filled border region in pixels.

Map Quality

Shading Samples (1)

Number of shading samples taken per pixel when baking each map.

Pixel Samples (2)

Number of sub-pixel samples (in texture space) used when baking maps. This value can usually be quite low.

Filter Name (gaussian)

Pixel filter to apply to the baked map.

Filter Width (2)

Width of the filter region for the above filter.

See Also:

Baking Maps with BakeAir

17.9 Outlines Tab

Air can draw outlines on a rendered image at full sub-pixel resolution. Use the controls in this tab to enable and fine tune Air's outline capability.

Draw Outlines (default off)

Check to enable outline drawing.

Line Color (default black)

Default color for outlines.

Line Width (default 1.0)

Default line width in pixels.

Image-based Edge Detection

Outline regions (default on)

When enabled, Air assigns a unique id to each primitive, and boundaries between region ids are detected as edges.

Silhouette Threshold (default 0.5)

The silhouette threshold gives a tolerance for edge detection based on surface orientation. Larger values detect fewer edges. Set the threshold to 0 or 1 to disable image-based silhouette edge detection.

Geometry-based Edge Detection

Minimum Angle (default 0)

When greater than 0, this setting gives the minimum angle in degrees between adjacent faces for marking edges. Larger values detect fewer edges.

Silhouette edge detection (default off)

When enabled, Air draws lines for silhouette edges in the mesh used for rendering.

Include single edges (default off)

When enabled, single edges are marked as outlines.

Max Edge Length (default 0)

When greater than 0, this value sets the maximum edge length in pixels when rendering geometry-based edges using a stroke shader. Edges that are longer than the specified maximum are split until the criterion is satisfied. Requires Air 13.20 or later.

Fade with Distance

Fade with distance (default off)

Check to enable outline attenuation with distance.

Start Distance (defautl 10000)

The distance at which to begin narrowing outline width based on distance to the camera.

Min Line Width (default 0)

The minimum line width in pixels for outline attenuation with distance.

Stroke Shaders

Stroke shaders can be used to customize how outlines are drawn. Stroke shaders only work with edges detected using one or more of the geometry-based edge detection methods. Use the Create Stroke button to add a stroke shader node to a scene. To display a stroke node for editing, double-click the node name in the list window. Use the Delete Stroke button to remove an unwanted stroke node.

Mode (Enabled Disabled)

Whether to export stroke shaders when rendering.

See Also:

Air User Manual: Outlines for Toon Rendering and Illustration

18 Node Reference

The following additional nodes are available when Air Stream is installed:

Air Stream Node
Air Stream Displacement Shader
Air Stream Environment Shader

Air Stream Generic Shader
Air Stream Imager Shader
Air Stream Instancer Shader
Air Stream Light Node

Air Stream Prim Var Node
Air Stream Stroke Shader
Air Stream Surface Shader
Air Stream Volume Shader

18.1 Air Stream Node

The Air Stream Node holds controls for assigning custom Air properties to individual objects.

To apply an Air Stream Node to a selected object, go to the Air Stream menu and choose:

Air Stream Node -> Create New Air Stream Node

RIB Export

Archive Mode (Inherit 'Do not archive' 'Update archive' 'Read only' 'Read mesh file')

The archive mode determines how the associated geometry is handled during the RIB export process.

Inherit
Shape data is processed based on the global archive mode in the Export tab.

Do not archive
Shape data is never written to an archive and is always included in the main scene file.

Update archive
Write the shape data to an external archive if the global archive mode is read or write, overwriting any pre-existing archive.

Read only
Use the shape data in the associated archive for rendering; never overwrite the archive, even when the global archive mode is write.

Read mesh file
Read the shape data from the file specified in the Mesh File Name attribute below. The mesh file is normally generated from an earlier invocation of Air or BakeAir with the 'Write render mesh' option enabled.

Read Method (Inherit 'Read always' 'Read when object bound is hit')

The archive read method determines how an archive is referenced by the main scene file:

Inherit
Use the read method specifed in the Export tab of the render globals.

Read always
The archive is always read into the scene before rendering begins.

Read when object bound is hit
The archive is read when and only if the bounding box for the shape is encountered by the renderer. If the archive has been generated from another source, be sure the bounding box of the shape encompasses all objects contained in the archive.

Archive Frequency (Automatic 'Per Scene' 'Per Frame')

The archive frequency tells Air Stream how often the shape needs to be archived, or when a custom archive is used, whether the archive name varies per frame. Shapes whose vertex positions, shading normals, or texture coordinates change during an animation require a separate archive for each frame.

Automatic
Let Air Stream determine whether the shape requires an archive per frame.

Per Scene
Force a single archive to be used for all frames.

Per Frame
Use a separate archive file for every frame.

Custom Archive Name (default empty)

Optional full path name for the archive used during RIB export. If no custom name is provided, Air Stream generates a path using the export archive path, the current scene name, and the DAG name of the associated shape node.

If you wish to substitute a pre-baked custom archive for the current shape, set the Archive Mode (above) to 'Read only'.

For per-frame archives, include the frame number with `$F`*n*, where *n* is the number of digits (zero-padded).

Apply Transform (default on)

When enabled, the transformation of the associated primitive is applied to the archive.

Write render mesh to file (default off)

When enabled, Air will save the render mesh for the associated primitives to the file specified below. This feature can be used to bake displaced geometry for faster re-rendering with Air or for use with another application.

Mesh File Name

Name of the file name for the render mesh.

Mesh Coordinate Space (Object World)

Coordinate space in which to save the render mesh geometry.

Export color sets (default off)

When enabled, exported Maya primitives will include any attached Maya color sets. The primitive variable name will be the same as the color set name.

Quality

Override shading samples (default off)

When checked , the number of shading samples (set on the Render tab of the Render Settings dialog) can be overridden by the companion **Shading Samples** value.

Override surface tolerance (default off)

When checked, the tolerance for tessellation (set on the Render tab of the Render Settings dialog) can be overridden by the companion **Surface tolerance** value.

Groups and Sets

Include Groups

Add the attached objects to the listed groups (separated by spaces or commas).

Exclude Groups

Exclude the attached objects from the listed groups (separated by spaces or commas).

SSS Point Set

Point set name for use with Air's subsurface scattering capability. The name should be a valid file handle if you wish to write the set to a file. The set name can include the frame number using the sequence $Fn where n gives the number of digits in the frame number.

SSS File Mode (Inherit Write Read)

Set the SSS file mode to `Write` to save the subsurface scattering point set to the file name specified in the SSS point set attribute. Set the mode to `Read` to load a previously created SSS point file. Saving and re-using a point set file can accelerate re-rendering.

IPR

IPR Shading Mode ('Inherit' 'Diffuse only' 'Fast re-lighting' 'Full shading')

Override for the global IPR shading mode set on the Render tab.

Global Illumination

Receive global illumination (default on)

When checked, the object receives light from the global illumination light source.

Contribute indirect illumination (default on)

When checked, the object will contribute indirect illumination when the global illumination mode is set to Indirect.

Indirect Shading (Inherit Constant Lambert Shader)

The indirect shading mode determines how Air computes the contribution from surfaces that are intersected by indirect rays:

Inherit
Use the shading mode set on the Global Illumination tab in the render globals dialog.

Constant
The object's color attribute value is used as the indirect shading result. This is the fastest mode since it does not require evaluating any shaders or illumination.

Lambert
Air estimates the indirect contribution by computing a Lambert (diffuse) illumination result and multiplying it by the object's color attribute. This mode is faster than the Shader mode because the object's surface shader does not have to be evaluated.

Shader
Air evaluates the surface shader assigned to the object to compute the indirect illumination contribution.

Use average color (default off)

When enabled, Air uses the average color specified below when computing the indirect contribution in Constant and Lambert shading modes.

Average Color (default middle grey)

When the above 'use average color' control is enabled, Air uses this average color as the surface color for the fast Constant and Lambert indirect shading modes. By setting the average color to an appropriate value, you can use the fast Constant or Lambert shading modes even with complex shading networks.

Bake Maps

Bake Mode (Inherit 'Do not bake' 'Update map' 'Reuse map')

The bake mode control can be used to override how map baking is performed by BakeAir:

Inherit
Use the bake mode set on the Bake tab in the render globals dialog.

Do not bake
Do not bake a map for the attached objects.

Update map
Bake a new map for each object even if the global bake mode is set to re-use.

Reuse map
Reuse a baked map if it is available regardless of the global bake mode.

Set Map Size (default off)

When enabled, the bake map width and height are taken from the following two attributes, instead of being computed by Air Stream.

Bake Map Width (512)

Bake map width override when Set Map Size is enabled.

Bake Map Height (512)

Bake map height override when Set Map Size is enabled.

Subdivision Surfaces

Split subdivision surface (default off)

When checked, Air splits a subdivision mesh into smaller pieces for tessellation.

Export polygon mesh as subdivision surface (default off)

When checked, any attached polygon meshes are exported as subdivision surfaces instead.

Interpolate Boundary (Off 'Sharp corners' 'Smooth edges')

Use this control to apply a boundary tag to a subdivision surface.

Displacement

Displacement Mode (Inherit 'Bump displacement' 'True displacement' 'Mesh displacement' Off)

Selects the method used for evaluating displacement shaders.

Displacement Bound (default 0)

The maximum distance in object space that a displacement shader will move the surface.

Triangulate displaced mesh (default off)

When checked, all quads are converted to triangles for displacement.

RIB Box

Use the text box in this section to add custom RIB statements.

Output to RIB (None Before Replace After)

Selects how the custom RIB is included in the exported RIB file.

See Also

Air User Manual: Shaders -> Displacements
Air User Manual: Lighting -> Subsurface Scattering

18.2 Air Stream Displacement Shader

The Air Stream Displacement Shader node allows a custom Air displacement shader to be assigned to an object.

To apply an Air Stream Displacement Shader:

- Select the surface shader associated with the target object.
- Click on the companion shading group tab.
- Click on the Displacement navigation button.
- In the Materials tab, choose Air Stream Displacement from the Displacement Materials list.

Click the button next to the **Shader** text box to display a file browser for choosing an Air displacement shader. The shader selection dialog lists all Air displacement shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

The top section of the displacement shader node displays a small preview image of a sphere with the displacement shader. When the **Preview** box is checked, the preview image will be automatically updated whenever a shader parameter values is changed.

A displacement shader may be evaluated as a bump effect, altering the surface normal only, or as true displacement, moving the surface position. The default displacement mode is set in the Displacement section of the Shading tab. You can override the displacement mode on a per-object basis by assigning an Air Stream node and using the controls in the Displacement section.

Attributes

Preview color (default middle grey)

Sets the color for the preview image.

Shader Parameters

This section lists the parameters for the currently loaded shader.

Shader parameters may be driven by a Maya shading node or network.

See Also

Air User Manual: Shaders -> Displacement

18.3 Air Stream Environment Shader

An Air environment shader is used to shade rays that miss all objects in a scene.

To create a new environment shader:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Environment Shaders section.
- Click Create Environment to create a new environment shader node.
- The shader selection dialog appears with a list of all Air environment shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. Choose an Air environment shader from the list or pick `[Select File]` to load a compiled shader file from a different location.

Air Stream allows multiple environment shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

Environment shader parameters can be driven by a Maya shading network.

See Also:

Environment Shaders
Air User Manual: Shaders -> Environments

18.4 Air Stream Generic Shader

The Air Stream Generic Shader node allows an Air generic shader to be used in a Maya shading network.

To add an Air generic shader to a network:

- Open the Hypershade window in Maya.
- Drag the shading network you'd like to work on to the Work Area.
- Under the Create tab, find the General Utilities section and click on the Airstream Generic Shader entry to add that node.
- In the attribute editor for the Airstream Generic Shader node, click on the file browser button next to the **Shader** name to choose an Air generic shader. The shader selection dialog lists all Air generic shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.
- Once the Air generic shader is loaded, you can modify its input parameter, and connect its inputs or outputs to other nodes in the shading network.

See Also

Generic Shaders
Air User Manual: Shaders -> Generics

18.5 Air Stream Imager Shader

An Air imager shader modifies the final rendered pixel values in the rendered image.

To add an imager shader to a scene:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Imager Shaders section.

- Click Create Imager to create a new imager shader node.
- In the Air Stream Imager Shader window, click on the browse button next to the **Shader** text box to choose an Air imager shader. The shader selection dialog lists all Air imager shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Air Stream allows multiple imager shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

Imager shader parameters can be driven by a Maya shading network. Texture coordinates are defined for an imager shader such that the unit square maps to the output image, allowing standard 2D texture nodes to be used.

See Also

Imager Shaders
Air User Manual: Shaders -> Imagers

18.6 Air Stream Instancer Shader

The Air Stream Instancer Shader node allows an Air instancer shader to be assigned to an object.

An Air Stream instancer shader can be assigned to an object using the Air Stream menu:

- Select the desired object.
- From the Air Stream menu, choose Instancer Shader -> Create New Instancer Shader.

Click the button next to the **Shader** text box to choose an Air instancer shader. The shader selection dialog lists all Air instancer shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Attributes

Region bound (default 1.0)

The region bound defines the region within which the instancer shader will create new objects. The bound is given as a distance used to expand the bounding box for the base object to generate a bounding box for the instancer.

Hide base object (default off)

When checked, the base object for the instancer will not be visible in the rendering.

Shader Parameters

This section lists the parameters for the currently loaded shader.

See Also

Instancer Shaders
Air User Manual: Shaders -> Instancers

18.7 Air Stream Light Node

Attach an Air Stream Light Node to a Maya light to use an Air custom light shader or assign other custom light attributes for rendering with Air.

To apply an Air Stream Light Node:

- Select the Maya light
- From the Air Stream menu, choose Light Node -> Create New Light Node.

Click the button next to the **Shader** text box to select an Air light shader. The shader selection dialog lists all Air light shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Attributes

Light Channel (default -1, range -1 to 9)

Selects a channel for storing the light's contribution to the scene. Per-light channels can be saved as extra output images on the Images tab of the Render Globals dialog. A channel value of -1 disabled per-channel output. Note that the global illumination light source (if present) is assigned to channel 0.

Shadow Map Depth (default 1, range 1 to 16)

The shadow map depth sets the maximum number of depth fragments to store at each pixel in the shadow map. A value of 1 produces a normal shadow map. Values greater than 1 cause Air to render a deep/fragment shadow map that stores opacity/coverage information as well as extra depth values at each pixel.

Use Custom Shader (default off)

Check this control to enable the custom Air light shader when rendering.

See Also

Air User Manual: Shaders -> Lights

18.8 Air Stream Prim Var Node

An Air Stream Prim Var Node allows Maya attribute data attached to a shape node to be exported as primitive variable data for rendering with Air. Air will automatically feed prim var data to a shader input parameter with the same name.

Export primitive variable (default on)

Whether to include this prim var in the exported RIB file

Maya Attribute Name

Name of the Maya attribute to be exported.

RIB Variable Name

Name of the variable in the exported RIB file. If no name is provided. the Maya Attribute name will

be used.

Class (Constant Uniform Vertex Varying 'Face Varying')

Variable class, which determines how the values are assigned to the associated primitive (and hence how many values the attribute should have).

Type (String Float Color Point Vector Normal Matrix)

Type of data.

Array Size (1)

Number of array elements in each entry.

18.9 Air Stream Stroke Shader

An Air stroke shader can be used to customize how outlines are drawn. Stroke shaders are only used for lines drawn using one of the geometry-based edge detection methods. Stroke shaders were introduced in Air 13.18.

To add a stroke shader to a scene:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Outlines tab and expand the Stroke Shaders section.
- Click Create Stroke to create a new stroke shader node.
- Use the shader selection dialog to choose a stroke shader. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Air Stream allows multiple stroke shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

18.10 Air Stream Surface Shader

The Air Stream Surface Shader node allows a custom Air surface shader to be assigned to an object.

The Air Stream surface shader node can be assigned to an object just like any other Maya surface shader:

- Select the desired object.
- From the Lighting/Shading menu, select Assign New Material, Air Stream Surface Shader.

Click the button next to the **Shader** text box to choose an Air surface shader. The shader selection dialog will list all Air surface shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

The top section of the surface shader node displays a small preview image of the surface shader. When the **Preview** box is checked, the preview image will be automatically updated whenever a shader parameter value is changed.

Basic Attributes

Color (default middle grey)

Sets the base color for the object, which is exported as the RIB Color attribute. The color attribute can be connected to a Maya shading network.

Transparency (default black)

Sets the base transparency for the object. The value (1-Transparency) is exported as the RIB Opacity attribute.

Surface modifies transparency (default off)

Use this control to indicate that the surface shader modifies the transparency of the object, which forces evaluation of the surface shader when the object is intersected by shadow rays.

Advanced Attributes

Shading Samples (default inherit)

Use this control to customize the number of shading samples for this surface shader. By default the number of shading samples is inherited from the setting on the global Render tab.

IPR Shading (default inherit)

Use this control to override the default IPR shading mode (set on the global IPR tab) for this shader.

Indirect Shading (default inherit)

Use this control to override the default indirect shading mode set in the Indirect Only section of the Global Illumination tab.

Spectrum

The base surface color may also be specified as a spectrum using a spectral data file. Spectral data is converted to rgb values for rendering, including the effects of spectral prefiltering and chromatic adaptation if that option is enabled.

Enable Ptex (default off)

Enables per-face texture mapping for use with Ptex textures.

Shader Parameters

This section lists the parameters for the currently loaded shader.

Shader parameters may be driven by a Maya shading node or network.

See Also

Air User Manual: Shaders -> Surfaces

18.11 Air Stream Volume Shader

An Air volume shader computes volumetric effects such as fog or smoke.

To add a volume shader as the atmosphere shader for all objects in the scene:

- Open the Maya render globals window and set Air Stream as the current renderer.
- Click on the Shading tab and expand the Atmosphere Shaders section.
- Click Create Atmosphere to create a new imager shader node.
- In the Air Stream Volume Shader window, click on the browse button next to the **Shader** text box to choose an Air imager shader. The shader selection dialog lists all Air volume shaders in the `$AIRHOME/shaders` and `$AIRHOME/usershaders` directories. To select a compiled shader file from a different location, choose the `[Select File]` entry at the top of the list.

Air Stream allows multiple volume shaders to be assigned to a scene. The shaders will be evaluated in the order in which they are listed.

See Also

Air User Manual: Shaders -> Volumes

19 Scene Conversion

Air Stream exports Maya scene data to a RIB file for rendering. This section documents the Maya features that are currently supported.

Geometry

Polygons

A polygon mesh is exported as a RIB `PointsPolygon` primitive with shading normals and standard texture coordinates.

NURBS Surfaces

NURBS surfaces are translated to RIB `NuPatch` primitives. Trim curves are not yet supported.

Subdivision Surfaces

Subdivision surfaces are exported as RIB SubdivisionMesh primitives with texture coordinates. Boundary interpolation can be specified using an Air Stream attribute node. The same node can also be used to force export of a polygon mesh as subdivision surface.

Particles

All Maya particle render modes are supported except Numeric and Tube. Most per-particle attributes are supported.

Fur

Maya Fur is supported using Air instancer shaders which generate fur curves at render time.

Hair

Maya Paint Effects hair is supported.

Curves

Curves will be exported as `NuCurves` when the Render curves attribute is enabled in the Geometry section of the Render tab.

Fluids

Fluids can be rendered as volumes or implicit surfaces.

Lights

Standard Maya lights are translated to the nearest equivalent Air light shader:

Maya Light	Air Light Shader
Ambient light	ambientlight
Directional light	distantlight
Point light	pointlight
Spot light	spotlight
Area light	arealight
Volume light	ASM_VolumeLight

Ray-traced and shadow-mapped shadows are supported.

Object Properties

The following Render Stats properties are translated:

Primary Visibility
Casts Shadows
Receive Shadows
Visible in Reflections
Double Sided
Opposite

20 FAQ - Frequently Asked Questions

1. How can I render a surface with subsurface scattering (SSS)?

First, assign one of Air's surface shaders that utilizes subsurface scattering (such as VTranslucent or VSkin) using an Air Stream surface shader node. Then assign an Air Stream node, and in the attributes section, provide a name for the SSS point set.

2. How can I use particles to create instances of geometry?

Assign an Air Stream instancer shader node to the particle shape. Check the **Hide base object** control. Use the VInstanceArchive instancer shader or your own instancer shader to add archive references based on particle properties.

Also see the following tutorials:

Tutorial: Custom Shading for Instanced Geometry
Tutorial: Instancing Multiple Archives

21 History

21.1 Release 2

- Maya fluids
- Support for Joe Alter's Shave and a Haircut
- Plug-in builds for Maya 2014
- Support for the new curve tapering attributes in Air 12.10 for Maya fur and Maya curves
- Maya's Optimize Scene command no longer deletes environment, imager, or atmosphere shader nodes
- Render tab
  - New option to set the Primary Hider for rendering under Image Quality
- Images tab
  - New Crop Window option
  - New __diffuse_shadow output channel contains an alternate shadow pass value, computed as:

    diffuse_shadow = diffuse_unshadowed - (diffuse - indirect)

    so in a composite:

diffuse = diffuse_unshadowed - diffuse_shadow + indirect

- Global Illumination tab
  - Controls for the new radiosity cache in Air 13

- Stereo rendering improvements:
  - Air Stream detects a standard Maya stereo rig and automatically adds the left and right cameras to the list of renderable cameras.
  - When multiple cameras are rendered, the camera name is added to the image name to create a unique file name for each view.
  - IPR uses only the first renderable camera when there is more than one.

- The Textures section of the global Shading tab has a new control for automatically converting raw textures in sRGB color space to linear color space at render time. (Requires Air 12.10 or later)

- Shaders
  - Support for many Mental Ray shaders included with Maya. The exporter has been extended to recognize and process MR shading nodes, so you can add support for additional MR shaders by coding an equivalent Air shader (as with other Maya shading nodes).
  - Air volume shaders for atmospheric effects are now supported. The interface is in a new Atmosphere Shaders section on the Shading tab.
  - New Maya shaders: Crater, EnvSky, Mountain
  - Fix for grid in EnvChrome shader

- Dropped support for Maya 2009.

21.2 Release 1

- Maya instances are now supported

- Shaders
  - Ramp texture shader provides more color entries, and wave and noise parameters are supported
  - NoiseUV supported for Place2dtexture node

- Fixes
  - Alpha now appears in a Maya render preview image

21.3 Build 19

- Maya Fur
  - Multiple fur descriptions per object are supported (requires Air 11.08 or later)
  - Baked maps for most fur geometry attributes are supported
  - A custom Air surface shader can be assigned to fur (using the new Assign Fur Surface command in the Fur, Hair & Curves menu)
  - Fur attractors (requires Air 11.08 or later)

- Light linking

- Shading Tab
  - Imager and environment shaders can be disabled only for preview and IPR rendering

- Render Tab
  - Default width for curves can be set in the Geometry section

- Images Tab
  - The interface for controlling output images has been simplified; some of the changes are not backwards compatible with scenes saved from earlier builds.

- Output
  - Air Stream now appends the layer name of any non-default layer to the image name and rib name used for export (if the image and rib fields are left in their default state).

- Fixes
  - Restored inheritTransform attribute to the Air Stream node (inadvertently removed in last build)
  - Air Stream now uses the animation frame range set on the Common tab in the Render Settings dialog (in previous builds the time slider range was usually used).
  - Air Stream correctly handles objects whose visibility is animated

21.4 Build 18

- Maya 2013 support

- Maya Hair

- Curves

- Lighting
  - Changed default light source to a distant light shader to better match Maya
  - Air Stream now respects the Enable Default Light toggle in the Common render tab

- Maya Fur
  - Any instancer shader attached to the fur base object is now applied to the fur curves

- Rendering
  - Under Windows, rendering to the Maya render view now sends image data directly to Maya over a socket connection (like Air Show).
  - Air Stream now always calls 'air' to render a shader preview (instead of using the current render command)

- Miscellaneous
  - Simplified the output window messages for the render and export commands.
  - Removed unused Air Stream Node attributes.
  - Reorganized some global attributes. Surface tolerance, point attributes, and curve attributes are now in a new Geometry section in the Render tab. Subsurface scattering globals have their own section in the Shading tab.

21.5 Build 17

- Maya Fur

- Lighting
  - The Air Stream Light Node now has a control for setting the shadow map depth for deep/fragment shadow maps
  - The Maya light shadow mode setting is now transferred to an Air custom light shader if the shader has a 'shadowname' parameter that is empty. This permits custom light shaders to use shadow maps easily. Note that only the shadowname value is set; other shadow parameters such as bias and samples are not transferred from the Maya settings.

    - For Maya lights, shadow map blur is now set based on the Maya filter size and map resolution. Prior releases always used the default shadow blur of 0.01 for shadow maps, which precluded rendering sharp shadows.

- IPR
    - An active IPR session is closed when a new scene is opened or Maya is closed

- Maya Camera
    - Air Stream now uses the Maya rendering frustum for better translation of Maya camera parameter
    - Orthographic projection width is translated
    - Depth of field

- Shaders
    - Cloth and Grid shaders now set OutAlpha
    - Fix for pulse train function used by Grid, Checker, and Cloth shaders
    - Added scale parameter to DisplacementShader

- Displacement
    - Air Stream sets a displacement bound for the Maya DisplacementShader node based on the Scale parameter value.
    - Air Stream sets a displacement bound for Air displacement shaders based on the BumpMax parameter value.
    - Air Stream node attributes are now written to the RIB file after the displacement shader is declared, so the displacement bound setting on an Air Stream node can be used to override the computed displacement bound

- Shading Tab
    - New controls for setting cache sizes for tiled textures, automatic shadow maps, and Ptex textures
    - Default ray tracing offset changed from 0.2 to 0.05

- Fixes
    - Fixed camera transform export for TweakAir when motion blur is enabled

21.6 Build 16

- IPR
    - Global illumination will update when environment shaders are modified (using AIR 11.06 with Occlusion and Approximate environment sampling)
    - IPR global controls moved to a new IPR tab
    - New controls for quantization, number of threads, and process priority

- Shading Tab
    - New toggles to enable/disabled imager shaders and environment shaders
    - Control to enable the new cache for global illumination environment queries introduced in Air 11.06

- Global Illumination
    - New documentation
    - A separate cache file can now be referenced for each frame by including $Fn in the cache file name (where n is the number of digits for the frame number).
    - IPR override options for many global illumination settings on the Global Illumination tab

- Render Tab
    - New Optimization section

  - New Render Control option to render at low process priority under Windows, allowing foreground tasks (such as Maya) to run unimpeded.

- Surface Shader node
  - New controls for customizing the IPR shading mode, indirect shading mode, and number of shading samples

- Air Shelf
  - Surface assignment buttons now assign a node name based on the shader name
  - New `asuMakeAirSurfaceShaderEx` MEL procedure allows attributes for full IPR shading and shader-modified transparency to be set

- Documentation
  - New Image Quality section
  - New Shaders sections

- General
  - Preview and IPR rendering use the render view test resolution setting in Maya
  - Preview and IPR rendering respect the Render Selected Objects Only setting in the Render View
  - Air Stream will stop an IPR process if it is running when Maya shuts down
  - Shader selection presents a list of all shaders of the relevant type found in `$AIRHOME/shaders` and `$AIRHOME/usershaders`. Choose the [Select File] option at the top to pick a shader from a different location.
  - Air Stream now uses the shader parameter dictionary included with Air to display nicer parameter names (for light shaders in particular)

- Fixes:
  - Fixed bug in bounding box calculation for polygon meshes and subdivision surfaces
  - Pre-filter clamp to alpha option is now written to the RIB file
  - Air Stream now replaces colons with underscores in DAG path names when generating file names for archives

21.7 Build 15

Note: Some of the new baking and particle features in this release require shaders distributed with Air 11.05 and later.

- Texture baking with BakeAir
  - Tutorial: Introduction to Baking
  - Tutorial: Baking Global Illumination
  - Tutorial: Baking a Procedural Pattern

- Particles
  - Air Stream will export any per-particle attributes referenced by name in an instancer shader parameter list.
  - New tutorial on custom shading for instanced geometry
  - New tutorial on instancing multiple archives

- New Maya shaders:
  - Marble, Snow, Wood, Stucco

- Render Globals
  - Ray Trace tab replaced by a Shading tab

- Ray traced controls are now under the Shading tab
- Imager shader, displacement, and advanced (spectral) lighting controls moved from the Render tab to the Shading tab.

- Fixes
  - Fix for __shadow output calculation in Maya surface shaders
  - NURB surface u/v orientation now matches Maya
  - Fix for ambient light contribution in Maya surface shaders
  - Air 11.05 fixes a bug that could corrupt light shader output values for surface shader networks

21.8 Build 14

- IPR
  - The IPR view now automatically updates when changes are made to tweakable elements.
  - IPR support for tweaking surface, displacement, imager, and environment shaders and shader networks. Tweaking shading networks requires Air 11.04 or later.
  - New option to hide the command window for the TweakAir process.

- Particles
  - Per-particle scale X and scale Y values for sprites
  - Sprite twist support (requires Air 11.05 or later)
  - Sprite cycling of file textures when hardware texture cycling is enabled

- Export tab
  - Controls for local caching of resource files
  - Controls for render-time path mapping

- New Maya Shaders
  - Cloud, SolidFractal, Granite, VolumeNoise

- Air shelf
  - New menu item to create an Air shelf in Maya with a few convenience buttons for common shaders and commands

- Fixes
  - Fixed a bug that prevented animated camera attributes from being re-sampled per frame

21.9 Build 13

- Primitive Variables
  - A new Prim Var node allows Maya attributes to be exported as primitive variable data
  - A new control on the Air Stream node enables export of color sets attached to polygon meshes

- Render Tab
  - New Displacement controls to set global displacement mode and bounds

- Air Stream Node
  - New controls in the RIB Export section to configure Air's mesh export feature, which can be used to bake displaced meshes for accelerated re-rendering or use with another application
  - New displacement mode option to disable displacement

- Shaders

- Maya's volume light is now supported
- New SamplerInfo node shader
- Shader preview rib now includes the global shader search path (so custom shaders can be found)
- Tweak for Ramp shader to no longer reverse the second texture coordinate input when that value is provided as a single input

- Images Tab
  - New presets for Incandescence and unlit Color extra images

- Fixes:
  - Object visibility overrides on display layers are now respected.
  - Render previews and render/export current frame now set the current frame number properly
  - Render/export sequence now resets the current frame when done.

21.10 Build 12

- IPR - Interactive Preview Rendering with TweakAir

- Render Tab
  - New motion blur controls to reduce quality for blurred objects, reducing the time to render
  - New controls for IPR (interactive preview rendering) with TweakAir

- Global Illumination Tab
  - New Tweak Result section with controls for modifying the GI result
  - New Occlusion Only section with controls to set the shadow color and to select the environment sampling method
  - Old environment controls have been removed. Use environment shaders instead.

- Ray Trace Tab
  - Controls for the old reflection environment modes have been removed. Use environment shaders instead to set the properties for the traced environment.
  - The environment shader list is now in the Ray Trace tab (moved from the Render tab).

- Export Tab
  - The default global Archive Mode (Off) will use a custom archive specified on an Air Stream node if the Air Stream node archive mode is set to Read Only

- Output images and passes
  - The name of an additional render pass node is now based on the selected render pass
  - Extra passes may be embedded in the primary image for display drivers other than OpenEXR
  - An extra pass may have an alpha channel appended

- Particles
  - Support for additional per-particle attributes including birthtime, age, lifespanPP, userScalar*PP, and userVector*PP
  - Air Stream now looks for ParticleSamplerInfo nodes in the surface shader network to determine which additional attributes to export
  - Sprite width and height are now taken from the corresponding sprite attributes

- Default light source is no longer added when global illumination is enabled and no other lights are present.

21.11 Build 11

- New controls on the Export tab and Air Stream Node allow archives to be loaded on demand (using DelayedArchive procedural calls)
- Disk-based shadow maps for spotlights, point lights, and directional lights
- New menu option to export selected objects to an archive, including shaders and attributes
- Particles
  - Motion blur
  - Streak, blobby, cloud, multi-point, and mult-streak render types
  - Basic ParticleSamplerInfo node with color and transparency outputs
  - Very basic ParticleCloud shader
  - New documentation with details for each particle render type
- Air Stream node
  - New controls to override the number of shading samples
  - New control to set the SSS point set file mode
  - SSS point set handle may now include the frame number with `$F`*n* where *n* gives the number of digits
- Render tab in the render globals
  - The Extras section has been renamed Global Defaults
  - New control to set the SSS file mode globally
  - New controls to configure particle rendering
  - New control to enable motion blur for traced rays
- Fixes:
  - Air Stream now skips particles with no surface shader assigned (such as particles used only for simulation)
  - Menu items to assign existing Air Stream, instancer, and light nodes to objects now work

21.12 Build 10

- Shaders
  - New UseBackground shader
  - Added OutTransparency to the FileTexture node
  - FileTexture node now uses default texture coords when no UV coordinates are connected
  - Frame sequence support for file names
  - Projection node support for linked camera in perspective mode
  - The default environment shader is now envMap (instead of ConstantEnvironment)
- Basic render layer visibility is supported
- Basic support for an image plane used to add a background image
- Area lights are no longer visible to the camera
- The user guide now has a FAQ.
- Bug fixes:
  - Air Stream no longer caches the AIRHOME path internally, making scenes more portable

- PointsGeneralPolygons output now properly handles per-vertex normals
- Polygon meshes exported as subdivision surfaces now include texture coordinates when available
- Object export no longer skips connected meshes
- Polygonal objects with no visible vertices no longer throw an exception and halt rib export

21.13 Build 9

- Builds for Maya 2012

- New scene export modes that automatically create and reference archive files for primitive geometry data. These modes can greatly accelerate the export process for non-trivial scenes.

- Motion blur
  - Many bug fixes
  - Removed unused controls from the user interface
  - Deformation blur no longer tries to blur static shapes
  - Deformed polygon and subdivision meshes now write only position data when possible, reducing rib size

- Maya render region is now supported

- Spectral color support
  - New Advanced Lighting section on the Render tab with controls for spectral prefiltering and chromatic adaptation
  - The File Texture node now automatically converts color textures to current color space (so spectral prefiltering and chromatic adaptation apply automatically to color textures)

- Occlusion Pass
  - Air Stream can automatically add an occlusion value to the output from any render.

- Ray Trace Tab
  - New toggle for optimizing the memory used to store ray-traced objects that are invisible to reflection rays

- Render Tab
  - Changed shading rate control to shading samples (the inverse of the shading rate)
  - New Advanced Lighting section

- New Outlines tab in the render globals

- Transformation matrices for objects are now written using `ConcatTransform` instead of `Transform` (for compatibility with Air Space)

- Air Stream Surface Shader
  - New check box to enable per-face texture mapping
  - New Spectrum file attribute

- Air Stream Light Node
  - Node now shows up in the Maya attribute editor when attached
  - Use custom shader is now disabled by default
  - Falloff parameter set to 0 by default

- Particles

- Sphere and sprite render types are automatically recognized

21.14 Build 8

- Important! Ids for the Air Stream nodes have all been changed to new reserved numbers to avoid any possible conflict with other plug-ins. This one-time change is not backwards compatible. I.e., the plug-in will not recognize Air Stream nodes in files saved using prior releases of Air Stream.
- Imager shaders. Imager shader parameters may be connected to a Maya shading network.
- Environment shaders. With Air 10.20 and later, an Air environment network can be driven by a Maya shading network.
- Moved camera transformation placement in RIB to just after Projection (so coord sys declarations for GI and ray tracing work properly)
- New shaders: gammaCorrect, clampColor, HsvToRgb, Condition, envCube, Noise
- When an environment shader is used, the occlusion Gi mode uses the new sampled environment mode introduced in Air 10.20.
- Air shader preview now works under Linux
- envSphere and envChrome shaders updated to work with imager and environment shaders (requires Air 10.20)

21.15 Build 7

- Air surface and displacement shaders can now be driven by Maya shading nodes.
- A new Airstream Generic Shader node allows an Air generic shader to be used in a Maya shading network. Generic shaders provide a general mechanism for adding custom shading language functions to a Maya shading network.
- The shading network translator has been generalized to allow Air shaders and Maya shading nodes to be intermixed in a shading network.
- Support for per-light output channels.
- New pass presets for Environment (ray traced environment contribution) and Background (imager shader contribution).
- New documentation for output images and passes
- ReflectedColor supported in Maya surface shaders
- More efficient polygon mesh export - facevarying normal vectors are now converted to vertex values where possible
- Projection nodes now save and restore standard texture coordinates
- Fixed bug in custom shader initialization: string parameters with non-empty default values are now

saved in the Maya scene file.

- New shaders: contrast, envSphere, envChrome

21.16 Build 6

- Fix for render preview, render frame, and render sequence commands under Linux.

- Custom Air shaders
  - Parameters now load under Maya 2011 Linux.
  - Shader file dialog now defaults to the `$AIRHOME/shaders` directory and the .slb file extension.
  - The file browser for shader parameters remembers the last directory used.

- Shaders for additional Maya nodes: ramp shader, bump3d, brownian, cloth, fractal, leather, rock

- Render Tab
  - Image quality controls have been simplified.
  - New edit box for entering custom options and attributes.

- Under Windows vortex can be used for distributed rendering over multiple machines.

- Under Windows airq can be used to submit jobs to the Air Control render queue.

21.17 Build 5

- Maya shader networks for surface and displacement shaders are now translated to RIB shader networks for rendering with Air

- New documentation

- Render tab
  - Changed default pixel samples to 4x4 and default pixel filter to gaussian
  - New option to use all CPUs for rendering
  - New option to specify the render command
  - Global group membership controls moved to Render tab; fix for the RIB export of global groups

- Ray trace tab
  - New controls for default number of reflection and refraction rays for Maya shaders
  - Environment maps are now oriented with correct "up" direction by default
  - New Rotate control for orienting the environment map

- Global Illumination tab
  - Environment maps are now oriented with correct "up" direction by default
  - New Rotate control for orienting the environment map
  - Environment settings are now exported in occlusion mode
  - Cache file browser fixed

- Surface shader node
  - New preview scene including reflectable objects
  - Fix for shader preview to prevent infinite loop

- Support for shader parameters of type float[2] and float[3]
- Color shader parameters are now properly initialized

- Instancer shader node
  - New controls for instancer bound and instancer base visibility

- Air Stream node
  - New controls for tuning global illumination
  - Removed visibility controls that were redundant with Maya's Render Stats controls for primary visibility, reflection visibility, and shadow casting

- Render Sequence menu item has been implemented

- Lighting
  - Fix for the light transformation applied to distant light sources
  - Custom light sources attached to area lights are supported

- Output images
  - Fix to add frame numbers to image names
  - Multiple passes in a single file can now have different groups specified for each pass
  - Secondary passes now appear in render preview and render current frame