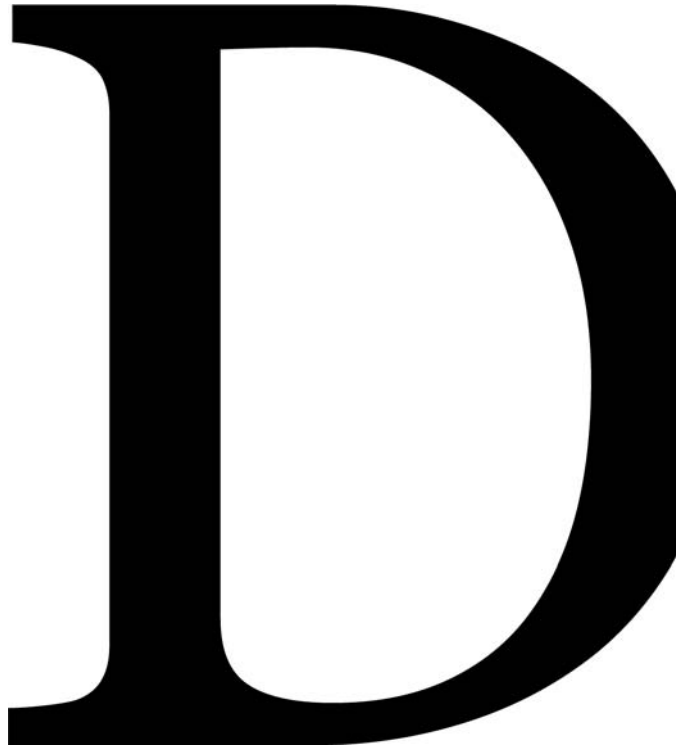


drUtilSuite\_v101



drUtilSuite\_v101 for mental ray.

**Suite** drUtilSuite\_v101

**Contact Information**

Duiker Research Corporation

Address

3450 Sacramento St. Suite 360  
San Francisco, CA  
94118

Phone

415 771 8991

Website

[www.duikerresearch.com](http://www.duikerresearch.com)

Email

Information and Questions [info@duikerresearch.com](mailto:info@duikerresearch.com)  
Support [support@duikerresearch.com](mailto:support@duikerresearch.com)

**Publication Date**

July 7, 2007

**Legal Information**

Copyright © 2006 - Duiker Research Corporation

**Suite** drUtilSuite\_v101

**Introduction**

The drUtil suite of mental ray shaders expands the set of basic shaders available to shader writers, lighters, technical directors, and 3d artists in general. The suite is primarily focused on use within Autodesk's Maya but can be used in any package that supports mental ray.

The drUtil suite consists of several groups of shaders. Those groups are listed here and described in the following pages.

- Logic Shaders
- Conversion Shaders
- Utility Shaders
- Layer Shaders
- Math Shaders
- State Shaders
- Lens Shaders
- Geometry Shaders
- Light Shaders
- Environment Shaders
- Debug Illumination Shaders
- Multipass Shaders
- Surface Component Shaders

The drUtil suite is available on Windows, Linux, and MacOSX for ray 3.4 and 3.5.

There is overlap between the functionality provided by the drUtil suite of shaders and the base shaders and utility nodes already available in mental ray and Maya. Where there was a question of implementing a feature due to pre-existing coverage, completeness of coverage was favored over minimizing redundancy.

## Installation Instructions

### **Maya 7.0 and mental ray 3.4**

First, copy the files in the mentalray\include directory into your Maya mentalray\include directory.

Second, copy the files in the mentalray\shader\platform (platform is win, linux, or mac) directory into your Maya mentalray\lib directory.

The drUtilSuite can be made to integrate better with Maya by installing the files in the mentalray\maya directory.

An example maya.rayrc is included in the mentalray\maya directory of the drUtilSuite\_v101 package.

Now, in a text editor, add the following lines to the mentalray\maya.rayrc file, and to the mentalray\ray3rc file if you have one.

```
link "drUtilShaders1.01.{DSO}"  
mi "drUtilShaders1.01.mi"
```

The files in the mentalray\maya\icons

The files in the mentalray\maya\others directory should be copied into your Maya scripts\others directory. These files will replace files of the same name, mentalrayCustomNodeClass.mel and mentalrayCustomNodeUI.mel. Copies of those files as originally distributed with Maya 7.0 are include in this directory as mentalrayCustomNodeClass.mel.bak0 and mentalrayCustomNodeUI.mel.

The files in the mentalray\maya\AETemplates directory should be copied into your Maya scripts\AETemplates directory.

**Changes**

2006.07.25

- Initial public release

2006.07.27

- Fixed shader export issue in Windows release.

2006.08.16

- Fixed missing symbol issue in Linux release.

2006.08.21

- Added new Conversion shaders
  - drInteger2Boolean\_v1
  - drInteger2Scalar\_v1
  - drInteger2Vector\_v1
  - drInteger2Color\_v1
  - drBoolean2Integer\_v1
  - drScalar2Integer\_v1
  - drVector2Integer\_v1
  - drColor2Integer\_v1
- Added new State Shaders
  - drStateReflectionLevelScalar\_v1
  - drStateRefractionLevelScalar\_v1
  - drStateLabelScalar\_v1
  - drStateInstanceScalar\_v1
  - drStateRayTypeScalar\_v1
- Added new Utility Phenomenon
  - drRayTypeSelect\_v1
- Moved shaders to the Deprecated list
  - drStateReflectionLevel\_v1
  - drStateRefractionLevel\_v1
  - drStateLabel\_v1
  - drStateInstance\_v1
  - drStateRayType\_v1

2006.11.28

- Added new Lens shaders
  - drLensPanoramic\_v1
  - drLensGamma\_v1
- Added new Utility shader
  - drSetOpacity\_v1
- Added new Geometry shader
  - drGeomSetAreaLightType\_v1
- Added example scenes demonstrating the use of
  - drLensPanoramic
  - drLensGamma
  - drGeomSetAreaLightType\_v1

2006.12.10

- Added new Environment shaders
  - drEnvironmentPanoramic\_v1
- Added new Math shaders
  - drModColor\_v2
  - drModScalar\_v1
  - drModVector\_v1
- Added debugging Illumination shaders

- drDebugIllumLambert\_v1
- drDebugLightPoint\_v1
- Includes source code and build projects for Win32, Linux, and OSX
- Added Multipass shaders
  - drMultipassGeometry\_v5
  - drMultipassGeometry8\_v5
  - drMultipassGeometry16\_v5
  - drMultipassSurface\_v5
  - drMultipassSurface8\_v5
  - drMultipassSurface16\_v5
  - drMultipassLens\_v5
  - drMultipassLens8\_v5
- Added example scenes demonstrating the use of
  - drModColor\_v2
  - drMultipassGeometry\_v5
  - drMultipassSurface\_v5
  - drMultipassGeometry8\_v5
  - drMultipassSurface8\_v5
  - drMultipassGeometry16\_v5
  - drMultipassSurface16\_v5

2007.01.09

- Redefined the different panoramic mappings used in the environment and lens shaders.

2007.01.11

- Added drLensBrightness\_v1
- Added node ids for Multipass shaders
- Updated drUtilShaders .mi declarations to include all new files
- Updated example scenes to be a bit more friendly out of the box

2007.05.21

Util Shaders

- Added the ShadowPass shader. This shader returns the shadowing percentage for the lights in the scene.
- Added Floor and Ceiling math shaders to the Color Suite
- Added the State Texture Coordinate shader
- Added Select Vector scalars
  
- Boost library is now statically linked into library on OSX and Linux. It's no longer necessary to install the library separately on machines using the shaders.

Multipass Shaders

- Exposed samples-based Multipass writing and merging shaders
- These have not been thoroughly tested.

Debug Illum Shaders

- Updated to .mi declaration to include min version. Ray was signaling a syntax error without one.
- Updated the XCode project to include Maya 8.5/ray 3.5 Universal Binary support
- Updated the XCode project to include separate configurations for Maya 7.0, 8.0, and 8.5

2007.07.07

- Added Windows x64 support.
- Added Linux x64 support.

**Group** Logic

**Suite** drUtilSuite\_v101

The Logic Shaders group provide shader level implementations of standard logic operations. The shaders names in this group should be fairly indicative of each shader's functionality.

The Logic Shaders are

- drAnd\_v1
- drAnd\_v2 - Array
- drOr\_v1
- drOr\_v2 - Array
- drXOr\_v1
- drNot\_v1
- drEqualScalar\_v1
- drNotEqualScalar\_v1
- drEqualColor\_v1
- drNotEqualColor\_v1
- drEqualVector\_v1
- drNotEqualVector\_v1
- drGreaterThanScalar\_v1
- drGreaterThanOrEqualScalar\_v1
- drLessThanScalar\_v1
- drLessThanOrEqualScalar\_v1
- drGreaterThanColor\_v1
- drGreaterThanOrEqualColor\_v1
- drLessThanColor\_v1
- drLessThanOrEqualColor\_v1
- drGreaterThanVector\_v1
- drGreaterThanOrEqualVector\_v1
- drLessThanVector\_v1
- drLessThanOrEqualVector\_v1
- drInRangeColor\_v1
- drInRangeScalar\_v1
- drInRangeVector\_v1
- drIsObject\_v1
- drInRasterPositionRange\_v1
- drInBetweenVector\_v1
- drIfThenElseScalar\_v1
- drIfThenElseColor\_v1
- drIfThenElseVector\_v1

**Group** Conversion

**Suite** drUtilSuite\_v101

The Conversion Shaders group provide shader-level implementations of conversions between the different standard data types within mental ray, namely Color, Vector, Scalar, and Boolean. The shaders names in this group should be fairly indicative of each shader's functionality.

The Conversion Shaders are

- drColor2Vector\_v1
- drColor2Scalar\_v1
- drColor2Scalars\_v1
- drColor2Boolean\_v1
- drVector2Color\_v1
- drVector2Scalar\_v1
- drVector2Scalars\_v1
- drVector2Boolean\_v1
- drScalar2Color\_v1
- drScalar2Vector\_v1
- drScalar2Boolean\_v1
- drScalars2Color\_v1
- drScalars2Vector\_v1
- drBoolean2Color\_v1
- drBoolean2Vector\_v1
- drBoolean2Scalar\_v1
- drInteger2Boolean\_v1
- drInteger2Scalar\_v1
- drInteger2Vector\_v1
- drInteger2Color\_v1
- drBoolean2Integer\_v1
- drScalar2Integer\_v1
- drVector2Integer\_v1
- drColor2Integer\_v1

**Group** Utility

**Suite** drUtilSuite\_v101

The Utility Shaders group provides an expanded set of shaders that perform commonly needed functions not provided for or not provided for cleanly already. The shaders names in this group should be fairly indicative of each shader's functionality.

The Utility Shaders are

- drDebugColor\_v1
- drDebugVector\_v1
- drDebugScalar\_v1
- drDebugBooleen\_v1
- drNullColor\_v2
- drNullScalar\_v1
- drNullVector\_v1
- drConstColor\_v2
- drConstScalar\_v1
- drConstVector\_v1
- drNormalizeVector\_v1
- drVectorNorm\_v1
- drSelectColor\_v1
- drSelectScalar\_v1
- drSelectColor2\_v1
- drSelectColor4\_v1
- drSelectColor8\_v1
- drSelectColor16\_v1
- drSelectScalar2\_v1
- drSelectScalar4\_v1
- drSelectScalar8\_v1
- drSelectScalar16\_v1
- drSwitchColor\_v4
- drRayTypeSelect\_v1
- drSetOpacity\_v1
- drSelectVector\_v1
- drSelectVector2\_v1
- drSelectVector4\_v1
- drSelectVector8\_v1
- drSelectVector16\_v1

**Group** Layer

**Suite** drUtilSuite\_v101

The Layer Shaders group provides allow for the manipulation of shader color channels in some of the same ways that compositing packages allow for the manipulation of image layers. The shaders names in this group should be fairly indicative of each shader's functionality.

The Layer Shaders are

- drOver\_v1
- drSwitchMatte\_v1
- drMatteMult\_v1
- drMatteDiv\_v1
- drSetAlpha\_v1
- drCopyChannelColor\_v1
- drCopyChannelVector\_v1
- drSetChannelColor\_v1
- drSetChannelVector\_v1
- drMixerColor2\_v1
- drMixerColor4\_v1
- drMixerColor8\_v1
- drMixerColor16\_v1
- drMixerScalar2\_v1
- drMixerScalar4\_v1
- drMixerScalar8\_v1
- drMixerScalar16\_v1
- drMixerVector2\_v1
- drMixerVector4\_v1
- drMixerVector8\_v1
- drMixerVector16\_v1
- drReorderColor\_v1
- drReorderVector\_v1

**Group** Math**Suite** drUtilSuite\_v101

The Math Shaders group provides shader-level implementations of common math operations for Colors, Vectors, and Scalars. The shaders names in this group should be fairly indicative of each shader's functionality.

The Math Shaders are

```

drAddColor_v2
drAddScalar_v1
drAddVector_v1
drSubColor_v2
drSubScalar_v1
drSubVector_v1
drSubAColor_v2
drSubAScalar_v1
drSubAVector_v1
drMultColor_v2
drMultScalar_v1
drMultVector_v1
drDivColor_v2
drDivScalar_v1
drDivVector_v1
drMinColor_v2
drMinScalar_v1
drMinVector_v1
drMaxColor_v2
drMaxScalar_v1
drMaxVector_v1
drMixColor_v2
drMixScalar_v1
drMixVector_v1
drClampColor_v3
drClampScalar_v1
drClampVector_v2
drThresholdColor_v2
drThresholdScalar_v1
drThresholdVector_v1
drCompressColor_v2
drCompressScalar_v1
drCompressVector_v1
drExpandColor_v2
drExpandScalar_v1
drExpandVector_v1
drInvertColor_v2
drInvertScalar_v1
drInvertVector_v1
drModColor_v2
drModScalar_v1
drModVector_v1
drFloorColor_v2
drFloorScalar_v1
drFloorVector_v1
drCeilColor_v2
drCeilScalar_v1
drCeilVector_v1

```

**Group** State

**Suite** drUtilSuite\_v101

The State Shaders group provides shader-level access to some of the variables that are defined as each point on a surface is shaders. The shaders names in this group should be fairly indicative of each shader's functionality.

The Math Shaders are

- drStateRasterPos\_v1
- drStateOrg\_v1
- drStateDir\_v1
- drStatePoint\_v1
- drStateNormal\_v1
- drStateGeometricNormal\_v1
- drStateMotion\_v1
- drStateRasterX\_v1
- drStateRasterY\_v1
- drStateRasterWidth\_v1
- drStateRasterHeight\_v1
- drStateDist\_v1
- drStateTime\_v1
- drStateDotND\_v1
- drStateInvNormal\_v1
- drStateReflectionLevel\_v1
- drStateRefractionLevel\_v1
- drStateLabel\_v1
- drStateInstance\_v1
- drStateReflectionLevelScalar\_v1
- drStateRefractionLevelScalar\_v1
- drStateLabelScalar\_v1
- drStateInstanceScalar\_v1
- drStateRayTypeScalar\_v1
- drStateTextureCoord\_v1

**Group** Lens

**Suite** drUtilSuite\_v101

The Lens Shaders group provides shader-level implementations of operations useful to apply to lenses. The shaders names in this group should be fairly indicative of each shader's functionality.

drLensPanoramic\_v1

drLensGamma\_v1

**Group** Geometry

**Suite** drUtilSuite\_v101

The Geometry Shaders group provides functionality not necessarily exposed in the mental integration provided by each application. The shaders names in this group should be fairly indicative of each shader's functionality.

drGeomSetAreaLightType\_v1

**Group** Light

**Suite** drUtilSuite\_v101

The Light Shaders group provides implementations of lights not covered in the standard mental ray distribution. The shaders names in this group should be fairly indicative of each shader's functionality.

drParticleLight\_v1

**Group** Environment  
**Suite** drUtilSuite\_v101

The Environment Shaders group provides implementations of environment map types not covered in the standard mental ray distribution.

drEnvironmentPanoramic\_v1

**Group** Debug Illumination

**Suite** drUtilSuite\_v101

The Debug Illumination Shaders group provides implementations of basic illumination nodes, lambert and point light, that will print out debug information. These can be useful in debugging mental ray scenes. The shaders names in this group should be fairly indicative of each shader's functionality.

drDebugIllumLambert\_v1

drDebugLightPoint\_v1

**Group** Multipass

**Suite** drUtilSuite\_v101

The Multipass Shaders group provides implementations of geometry, lens, and surface shaders that allow users to take advantage of mental ray's native support for rendering multiple frame buffers in a single render.

The lens shaders included in this group aren't needed, from an end-user perspective. The geometry shader will create the lens shader at render time, thereby removing one extra setup step. Example scenes in the examples\maya\scenes directory provide an example scene setup.

```
drMultipassGeometry_v5
drMultipassGeometry8_v5
drMultipassGeometry16_v5
drMultipassSurface_v5
drMultipassSurface8_v5
drMultipassSurface16_v5
drMultipassLens_v5
drMultipassLens8_v5
drMultipassSamplesMerge_v1
drMultipassGeometrySamplesWrite_v1
drMultipassGeometrySamplesMerge_v1
```

**Group** Surface Component

**Suite** drUtilSuite\_v101

The Surface Component Shaders group provides implementations of shaders that compute individual components of material properties.

drShadowPass\_v1