Appendix B – RenderMan Quick Reference

The reference material contained in this section is based largely on PIXAR's PhotoRealistic RenderMan Application Notes[1] #1 and #8 published in May 1990. The original "Quick Reference" was intended to be used by programmers and, therefore, contained a great deal of information that is irrelevant to those who wish to write RIB scripts directly by hand. Several RIB statements have been omitted because they relate to very advanced capabilities of the RenderMan interface. The information in this reference is of necessity very terse and is really only intended to act as a 'memory jogger'.

RIB Summary

Shape – geometric primitives

Cone
Cylinder
Disk
Hyperboloid
Paraboloid
Sphere
Torus
GeneralPolygon
PointsGeneralPolygons
PointsPolygons
Polygon

Camera

Clipping
DepthOfField
Display
Exposure
Format
FrameAspectRatio
FrameBegin/End
MotionBegin/End
Perspective
Projection
Shutter

Bookkeeping

#
Declare
Option

Space – transformations and grouping

Rotate
Scale
Skew
Translate
AttributeBegin/End
ObjectBegin/End
ObjectInstance
Sides
SolidBegin/End
TransformBegin/End
WorldBegin/End

Shading

AreaLightSource
Atmosphere
Color
LightSource
MakeCubeFaceEnvironment
MakeLatLongEnvironment
MakeShadow
MakeTexture
Opacity
ShadingInterpolation
ShadingRate
Surface
TextureCoordinates

[1] PhotoRealistic RenderMan Application Note #1 "A Brief Introduction to the RenderMan Interface"; PhotoRealistic RenderMan Application Note #8 "RenderMan Quick Reference"

[2] These include references to splines, trim curves, patches and patch meshes; transformation matrices; user defined coordinate systems; levels of detail; geometric approximation; bump mapping (not supported on the Macintosh platform); error handling; image filtering, sampling and quantization.

Shape – geometric primitives

Cone

Cone height radius thetamax parameters

Defines a partial or complete cone.

example

```
Cone 0.5 0.5 270 "Cs" [1 0 0   1 0 0   1 1 1   1 1 1]
```

"Cs" defines colours for the parameter space, which in this example provides the cone with a red base and a white apex.

Cylinder

Cylinder radius zmin zmax thetamax parameters

Defines a partial or complete cylinder.

example

```
Cylinder 0.5 0.2 1 360 "Os" [0 0 0   0 0 0   1 1 1   1 1 1]
```

"Os" defines opacity for the parameter space, which in this example provides the cylinder with a fully transparent base (opacity = 0,0,0) and a fully opaque top (opacity = 1,1,1).

Disk

Disk height radius thetamax parameters

Defines a partial or complete disk.

example

```
Disk 1.0 0.5 270 "Os" [0 0 0   0 0 0   1 1 1   1 1 1]
```

Opacity is used here to give the disk a fully transparent rim and a fully opaque centre.

Hyperboloid

Hyperboloid x1 y1 z1 x2 y2 z2 thetamax parameters

Defines a partial or complete hyperboloid.

example

```
Hyperboloid 1.0 -1.0 -1.0  1.0 1.0 1.0  360
```

Paraboloid

Paraboloid rmax zmin zmax thetamax parameters

Defines a partial or complete paraboloid.

example

```
Paraboloid 0.5 0.2 0.7 270
```

Sphere

Sphere radius zmin zmax thetamax parameters

Defines a partial or complete sphere.

example

```
Sphere 0.5 0.0 0.5 360
"Cs"  [1 0 0   1 0 0    0 0 1   0 0 1]
"Os" [0.7 0 0   0.7 0 0   1 1 1   1 1 1]
```

Both opacity and colour are used for the parameter space, which in this example provides the sphere with a semi-transparent red “base” and an opaque blue “top”.

Torus

Torus rmajor rmin phimin phimax thetamax parameters

Defines a partial or complete torus.

example

```
Torus 3.5 0.25 0.0 180 300
```

GeneralPolygon

GeneralPolygon nloops nvertices parameters

Defines a single convex or concave (general) planar polygon, with optional holes.

example

```
GeneralPolygon [3 3]
"P" [-1.0 -1.0 0.0  -1.0 1.0 0.0  1.0 -1.0 0.0  -0.5 -0.5 0.0   0.0 0.5 0.0
 0.5 -0.5 0.0]
```

PointsGeneralPolygons

PointsGeneralPolygons numLoops numVertices listVertices parameters

Defines several planar general polygons, with optional holes, that share vertices.

example

```
PointsGeneralPolygons [2 2][4 3 4 3][0 1 3 4 6 7 8 1 2 5 4 9 10 11]
"P" [0 0 1  0 1 1  0 2 1  0 0 0  0 1 0  0 2 0  0 0.2 0.5  0 0.7 0.7  0 1.7 0.2  0 1.2 0.5
0 1.7 0.7  0 1.7 0.2]
```

PointsPolygons

PointsPolygon numVertices listVertices parameters

Defines several non-concave polygons, without holes, that share vertices.

example

```
PointsPolygons [3 3 3][0 3 2 0 1 3 1 4 3]
"P" [0 1 1  0 3 1  0 0 0  0 2 0  0 4 0]
```

Polygon

Polygon listVertices parameters

Defines a single non-concave polygon with optional list of parameters that supplies information about vertex normals, colour, opacity and/or texture coordinates.

examples

–no additional parameters

```
Polygon "P" [0.0 1.0 0.0   0.0 1.0 1.0   0.0 0.0 1.0   0.0 0.0 0.0]
```

–plus additional parameters relating to vertex colours

```
Polygon "P" [0.0 1.0 0.0   0.0 1.0 1.0   0.0 0.0 1.0   0.0 0.0 0.0]
"Cs" [1 0 0   0 1 0   0 0 1  1 1 1]
```

–plus additional parameters relating to vertex opacity

```
Polygon "P" [0.0 1.0 0.0   0.0 1.0 1.0   0.0 0.0 1.0   0.0 0.0 0.0]
"Os" [1 1 1   0 0 0   0.5 0.5 0.5   0.2 0.5 0.8]
```

Shape – transformations and grouping

Rotate

Rotate angle dx dy dz

Turns the object space so that it is rotated by "angle" degrees around the given axis prior to a shape being defined.

example

```
Rotate 90 0 1 0
```

Scale

Scale sx sy sz

Stretches or compresses the object space so that it is scaled along the x, y and z axes prior to a shape being defined.

example

```
Scale 0.5 1.0 1.0
```

Skew

Skew angle dx1 dy1 dz1 dx2 dy2 dz2

Shears the object space so that it is skewed by "angle" degrees along the x, y and z axes prior to a shape being defined.

example

```
Skew 45 01 0  1 0 0
```

Translate

Translate dx dy dz

Shifts the origin of the object space so that it is translated along the x, y and z axes prior to a shape being defined.

example

```
Translate 0.0 1.5 0.0
```

AttributeBegin/End

AttributeBegin/AttributeEnd

Forms a group of shapes, transformations and surface attributes.

example

```
AttributeBegin
    some attributes such as color 1 0 0
AttributeEnd
```

ObjectBegin/End

ObjectBegin identifier/ObjectEnd

Defines a collection of shapes as a "retained" object that can be inserted, or instanced, within a scene.

example

```
ObjectBegin 4
    object 1
    object 2…
ObjectEnd
```

ObjectInstance

ObjectInstance identifier

Inserts, or instancies, a previously retained collection of shapes as a single object.

example

```
ObjectInstance 4
```

Sides

Sides sides

Defines subsequent shapes as single sided or double sided.

example

```
Sides 2
```

SolidBegin/End

SolidBegin operation/SolidEnd

Defines a collection of shapes as a "solid" object according to the rules of constructive solid modelling ie. union, intersection and difference.

example

```
SolidBegin "union"
    define two or more objects to be joined together as a single object
SolidEnd
```

TransformBegin/End

TransformBegin/TransformEnd

Forms a group of shapes and transformations but IGNORES surface attributes.

example

```
TransformBegin
    some transformations ex. Rotate
    some shapes
TransformEnd
```

WorldBegin/End

WorldBegin/WorldEnd

Freezes the characteristics of the camera and marks the beginning of a world description.

example

```
WorldBegin
    scene description
WorldEnd
```

Camera

Clipping

Clipping near far

Sets the near and far clipping planes along the direction of view.

example

```
Clipping 0.1 1000
```

DepthOfField

DepthOfField fstop focallength focaldistance

Parameters to simulate the depth of field.

example

```
DepthOfField 22 1 26.7
```

Display

Display name type mode parameters

Chooses a display by name and sets the type of output being generated.

examples

```
Display "filename" "file" "rgba"
Display "filename" "zfile" "z"
Display "windowname" "framebuffer" "rgba"
```

Exposure

Exposure gain gamma

Controls the sensitivity and non-linearity of the exposure process.

example

```
Exposure 1.5 2.3
```

Format

Format xresolution yresolution pixelaspectratio

Sets the horizontal and vertical resolution in pixels of the image to be rendered.

example

```
Format 400 300 1
```

FrameAspectRatio

FrameAspectRatio ratio

Ratio sets the ratio of the width to height of the desired image.

example

```
FrameAspectRatio 1.333
```

FrameBegin/End

FrameBegin/FrameEnd

Marks the beginning and end of a frame of animation.

example

```
FrameBegin 1
    scene description for this frame
FrameEnd
```

MotionBegin/End

MotionBegin t0 t1…tn-1/MotionEnd

Marks the beginning and end of motion

example

```
MotionBegin [0 1]
    transformation information at time 0
    transformation information at time 1
MotionEnd
```

Perspective

Perspective fov

Sets the camera to give a perspective view.

example

```
Perspective 90
```

Projection

Projection name parameters

Sets the type of projection and activates the camera coordinate system ie. the world coordinate system is only active between WorldBegin and WorldEnd.

example

```
Projection "perspective" "fov" 40
```

Shutter

Shutter opentime closetime

Sets the times at which the shutter opens and closes.

example

```
Shutter 0 1
```

Shading

AreaLightSource

AreaLightSource name int parameters

Creates an area light and makes it the current light source. Each subsequent object is added to the list of surfaces that define the area light.

example

```
AreaLightSource "finitelight" 1 "decayexponent" 0.5
AreaLightSource "glowlight" 2 "color" [0.5 0 0] "intensity" 0.6
```

Atmosphere

Atmosphere name parameters

Sets the currently active atmosphere shader.

examples

```
Atmosphere "fog" "background" [0.2 0.2 0.3] "distance" 39.4
Atmosphere "depthcue" "background" [0.2 0.2 0.3] "mindistance" ?
"maxdistance" ?
```

Color

Color red green blue

Sets the colour that will be applied to subsequent objects.

example

```
Color 0.2 0.3 0.9
```

LightSource

LightSource name sequencenumber parameters

Creates a non-area ie. infinitely small, light source, turns it on, and adds it to any other lights previously created.

example

```
LightSource "ambient" 2 "intensity" 10
```

MakeCubeFaceEnvironment

MakeCubeFaceEnvironment px nx py ny pz nz texturename fov filter swidth twidth parameters

Converts six images in a standard picture file (for example a TIFF file) representing six viewing directions into an environment map.

example

```
MakeCubeFaceEnvironment "foo.x" "foo.nx" "foo.y" "foo.ny" "foo.z" "foo.nz"
"foo.env" 95 "gaussian" 2.0 2.0
```

MakeLatLongEnvironment

MakeLatLongEnvironment picturename texturename filter swidth twidth parameters

Converts an image in a standard picture file (for example a TIFF file) representing a latitude-longitude map whose name is picturename into an environment map called texturename.

example

```
MakeLatLongEnvironment "long.tiff" "long.tx"  "gaussian"  2  2
```

MakeShadow

MakeShadow picturename texturename parameters

Convers a depth image file into a shadow map.

example

```
MakeShadow "shadow.tiff" "shadow.tx"
```

MakeTexture

MakeTexture picturename texturename swrap twrap filter swidth twidth

Converts an image in a standard picture file (eg. TIFF) into a texture file.

examples

```
MakeTexture "globe.tiff" "globe.tx" "periodic" "periodic" "gaussian" 2 2
MakeTexture "globe.tiff" "globe.tx" "black" "black" "gaussian" 2 2
MakeTexture "globe.tiff" "globe.tx" "clamp" "clamp" "gaussian" 2 2
```

In the first example the image will if necessay repeat horizontally and vertically. In the second example the image will be mapped once and will be surrounded by black. While in the last example, the colour of the pixels image at the extreme edge of the image will be "smeared" outward if there is enough space available on the object being texture mapped.

Opacity

Opacity c1 c2 c3

Sets the opacity to the colour channels c1, c2, c3, like the use of Color, subsequent objects are set to these levels of opacity.

example

```
Opacity 0.5 1.0 1.0
```

ShadingInterpolation

ShadingInterpolation type

Controls how the values are interpolated ie. estimated, between shading samples.

examples

```
ShadingInterpolation "constant"
ShadingInterpolation "smooth"
```

ShadingRate

ShadingRate size

Sets the number screen pixels, across and down the image, the renderer will skip between making its shading calculations – large numbers give fast but coarse images. The skipped pixels are either shaded with constant or "smoothed" colour – see above.

example

```
ShadingRate 10
```

Surface

Surface name parameters

Sets the current surface shader, subsequent surfaces acquire the 'look' of the chosen surface.

example

```
Surface "wood" "roughness" 0.3 "Kd" 1.0
```

TextureCoordinates

TextureCoordinates s1 t1 s2 t2 s3 t3 s4 t4

Sets the current set of texture coordinates.

example

```
TextureCoordinates 0.0 0.0    2.0 -0.5    -0.5 1.75    3.0 3.0
```

Bookkeeping

any text upto the **end of a line** is a comment

Enables notes to be included in a RIB file and ensures these will be ignored by the renderer.

examples

```
# this is a comment
Color 1 0 0   #this is another comment
```

Declare Declare name declaration

Declares a non-standard parameter.

example

```
Declare "centrepoint" "uniform float"
```

Option Option name parameterslist

Allows any pre-set option to be set from within a RIB file.

example

```
Option "limits" "bucketsize" [24  24]
Option "limits" "texturememory" [1024]
```