

3.3 Release Notes: 3D Graphics Kit

This file contains release notes for the 3.3 release of the 3D Graphics Kit. Release notes for 3.0 have been incorporated into the introduction of the printed and on-line Developer's Library (chapter 17 of *NEXTSTEP General Reference* or **/NextLibrary/Documentation/NextDev/GeneralRef/17_3DKit**).

Notes Specific to Release 3.2

Platform-specific Differences

1. The RenderMan software under NEXTSTEP does not recognize the token **int**. You have to use **integer**. **int** is not a valid RenderMan token, but it was allowed under some older implementations, and some versions of RenderMan running on other platforms. For example, RIB code like this:

```
Declare "foo" "uniform int"
```

will produce this error message under NEXTSTEP:

```
Warning in context RMContext00001 in command RiDeclare:  
UnrecognizedToken:  
    'type' == "int"
```

2. The Photoreal RenderMan software running under NEXTSTEP does not initialize a default camera space in the same way as PhotoReal RenderMan version on other platforms. If you attempt to make any transformations outside of world space without a Projection command, you will get an error like this:

```
Warning in context RMContext00001 in command RiWorldBegin:  
OutOfRange:  
    Non-identity world-to-screen transformation without projection  
    call.  
    Ill-defined camera space.
```

Bugs Fixed in Release 3.2

These bugs have been fixed in Release 3.2:

Reference	24246
Problem	RiLightSource() generated invalid <code>asequencenumbers</code> ^o
Description	The sequence numbers emitted by RiLightSource() when in 3.1-

compatibility mode weren't in the proper range.

Reference	31048
Problem	Memory leak in RenderMan
Description	There was a memory leak in the N3DShader object every time you used the setShader: method.

Reference	34628
Problem	Incorrect shading on NeXTdimension
Description	Solid shaded images on the NeXTdimension system were not correctly shaded—they were always shaded with a uniform color.

Reference	34741
Problem	Resurrection of NaN problem [Tracker #25553]
Description	With the default 3D graph shown in /NextDeveloper/Examples/Appkit/Graph , the equation is: $A * (u^2 + v^2) + C$ When changed to graph the square root of u, like this:

$$A * (u^{.5} + v^2) + C$$

the equation results in a NaN when trying to evaluate the square root of -2. This would work in the ^awire^o plot mode, but changing to one of the solid shaded modes would hang the Window Server.

Reference	34852
Problem	3DKit had classes with bad names
Description	Some of the internal 3Dkit classes did not adhere to the NEXTSTEP naming convention of using a proper prefix or an underscore at the beginning of the name.
Reference	34897
Problem	Crasher in NSDestroyContext() when exiting Backspace after using Teapot View
Description	If you managed to break up your 3D window into more than 64 rectangles, the Window Server would crash and you would be logged out.
Reference	35181

Problem	Polygons with normals drawn weren't rendered properly
Description	If you had a polygon with normals declared for all the vertices, it made a nice lovely drawing when drawn with N3D_SmoothSolids. However, if you drew it with N3D_FacetedSolids, all the polygons were rendered with the same color.
Reference	35386
Problem	N3DShader's setShader: method didn't check file format
Description	If you have an ASCII file or a generic file that is not a shader and you invoke setShader: with that file's name as the argument; your application would crash.
Reference	35894
Problem	zfile + RiMakeShadow() creates bad shadow texture on i386 architecture
Description	When creating a shadow by first rendering a depth file and then using RiMakeShadow() to create a show map, the shadow map is unusable. This happens on the Intel i386 architecture only. Rendering directly to ^a shadow ^o texture is a workaround but limits the RIB file's portability.

Reference	36516
Problem	Using light sources in a certain way caused logout on a NeXTdimension system
Description	On a NeXTdimension system, if you declared a light source outside of a world block (that is, before RiWorldBegin()), you would get logged out. All other architectures properly reported an error back to the client.

Notes Specific to Release 3.1

Bugs Fixed in Release 3.1

These bugs have been fixed in Release 3.1:

Reference	21211
Problem	.tx and .slo files should have an icon
Description	In release 3.0, neither .tx (texture) or .slo (shading language object) files have icons.

Reference	25374
Problem	Compiler incompatibility in the ri/ri.h header file
Description	There is an ambiguous function declaration in the ri/ri.h header file that causes the C++ compiler to crash. A workaround for 3.0 required editing the header file ri/ri.h . For Release 3.1, the edit is incorporated into the header file.
Reference	29630
Problem	NXImages don't work with loadFromFile: on RIB files
Description	A bug in N3DRIBImageRep's canLoadFromStream: method caused it to fail to return YES for the appropriate file types.
Reference	29471
Problem	QuickRenderMan crashes the Window Server if an application has two N3DCameras, one with lights, one without.
Description	In Release 3.0, there is an internal pointer to the light lists in the shading mechanism that's not properly synched with the real light list. You can get this problem when you have at least 2 N3DCameras, one with lights and one without. If you render the lighted one first, then the non-lighted one, you will get this crash.

Reference	29981
Problem	nupatch and big polygons can crash the Window Server.
Description	If you send a nupatch of almost any size, or a polygon with more than about 100 vertices, you will crash the Window Server.
Reference	29984
Problem	Intermittent DPS client errors from 3D programs
Description	Every so often, you will get "DPS client library error: Invalid port, DPSTContext 0, data 29" and your application will not work properly; sometimes it will just hang. Usually the only way to get going again is to quit the application. In release 3.1, the N3DCamera flushRib method is revised to prevent this bug.
Reference	30793
Problem	TrimCurve divide-by-zero problem causes photoreal renderer segmentation fault.
Description	The photoreal renderer may produce TrimCurve segmentation faults, caused by a divide-by-zero problem in the TrimCurve

routine. This bug affects N3DCamera's **renderAsTIFF** and **renderAsEPS** methods.