RenderWare Graphics

Viewer

RenderWare Visualizer

Contact Us

Criterion Software Ltd.

For general information about RenderWare Graphics e-mail info@csl.com.

Developer Relations

For information regarding Support please email devrels@csl.com.

Sales

For sales information contact: rw-sales@csl.com

Contributors

RenderWare Graphics development and documentation teams.

Table of Contents

1. Introduction

This Document

This document describes how to setup and use the RenderWare Visualizer.

The use of a standalone launcher to start the Visualizer with a specific piece of artwork is discussed.

The control options for varying the way in which the Visualizer renders artwork are detailed.

The Visualizer is also capable of displaying various performance metrics for your artwork. Section 4 contains a description of these metrics and discusses their interpretation.

RenderWare Visualizer

The RenderWare Visualizer is an application that can be used to view animated hierarchies and sections of static RenderWare Graphics geometry.

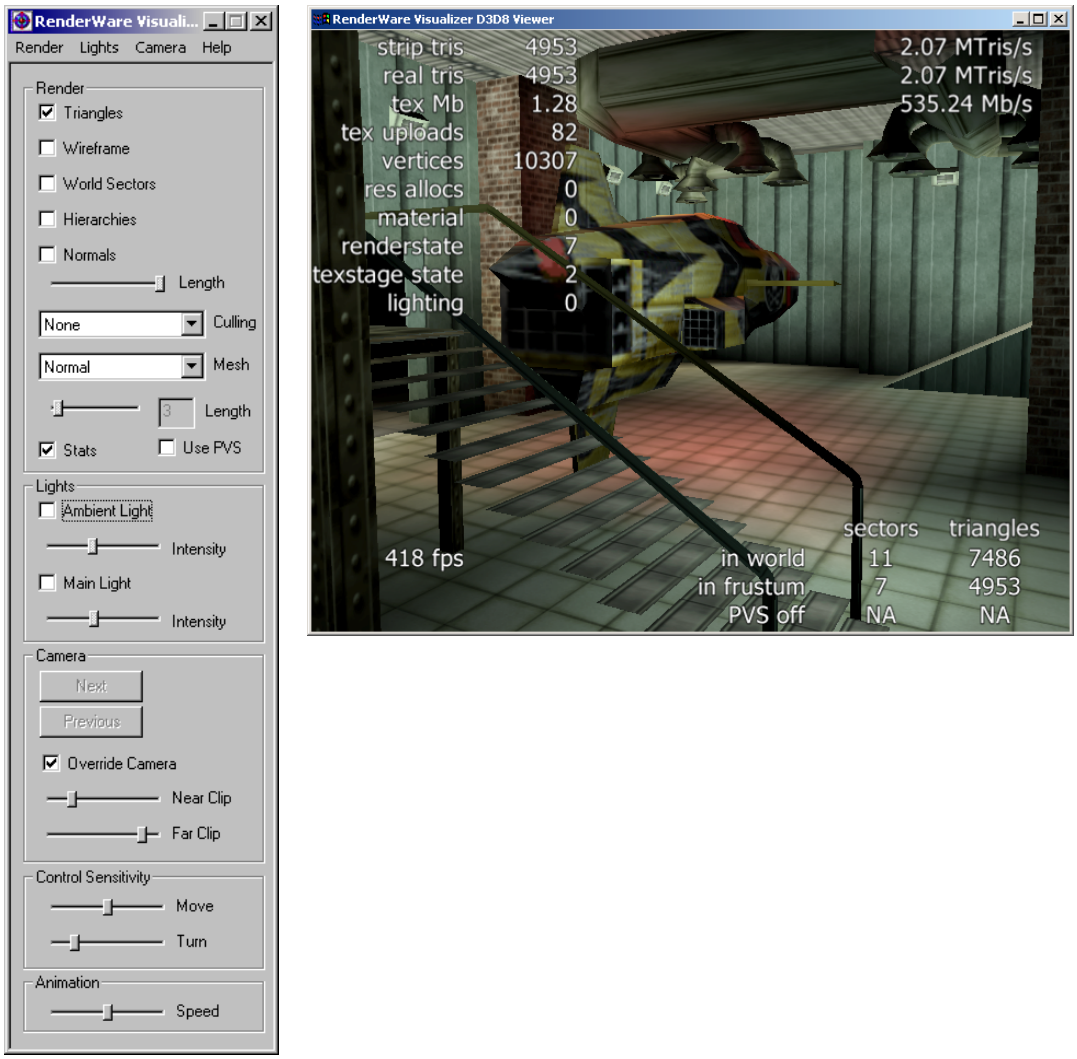
You can easily view your artwork in RenderWare Visualizer on any target hardware. RenderWare Visualizer also displays platform specific performance metrics that you can use to tune your artwork to get the most out of each platform.

RenderWare Graphics ships with exporters for 3ds max and Maya. The Visualizer enables you to preview your artwork from within the exporters, at once, at the click of a button.

The Visualizer can be used to view `.rws` files. These may contain static and animated graphics. These files also bundle in the textures used.

The Visualizer may also be used to export and view `.rf3` files. The exporter can create these text-based scene and animation files. They can later be converted to a platform specific form given an appropriate template file. See the exporter documentation for more details of the `.rf3` format.

`.rws` and `.rf3` files are the exporters' primary format. The visualizer can also display `.bsp`, `.dff`, `.dma` and `.anm` files, but these are now considered legacy formats.



Control Panel *Visualizer Viewer*

RenderWare Visualizer and Controls

RWS and RF3

RWS

An `.rws` file stores all export formats. This means that it can store worlds, animation, splines and clumps at the same time, which means that you're able to view your entire scene in one instance.

RF3

`.rf3` files are XML-based RenderWare scene files. Given an appropriate template file, the Visualizer can export `.rf3` files and view the resulting `.rws` file.

Legacy Formats

ANM

An .anm file stores only animation data and requires a .dff file of the same name to contain the geometry. To set up an .anm export format select an object or joint within the hierarchy you wish to export before running the exporter.

.anm is now considered a legacy format; .rws files can contain animations.

BSP

A .bsp file contains worlds that do not contain any hierarchy or animation information and are typically used for static level geometry in a game. They are automatically split into sections (BSP world sectors) that RenderWare Graphics uses to speed up the rendering process.

.bsp is now considered a legacy format; .rws files can contain worlds.

DFF

A .dff file contains a single hierarchy as a RenderWare Graphics container object, called RpClump (see the *Fundamental Types* and *Dynamic Models* chapters of the User Guide for more details on RpClump). Depending on the options you select in the DFF Exporter, the RpClump file saved will contain the object hierarchy, hierarchical animation, skinned animation, and morph target animation.

.dff is now considered a legacy format; .rws files can contain clumps.

DMA

A .dma file stores only DMorph animation data and requires a .dff file containing one or more atomics set up for DMorph rendering. .dma files can only be exported at the same time as a .dff export.

.dma is now considered a legacy format; .rws files can contain DMorph animations.

BSP world sectors should be viewed in the Visualizer.

Generating Artwork

RenderWare Graphics `.rws and .rf3` files can come from any source but are usually generated using one of the RenderWare Graphics export plugins for 3ds max and Maya. The RenderWare Graphics viewers form an integral part of the tool chain needed to get great looking artwork into RenderWare Graphics. The artist will usually run through a cycle of:

- Creating assets in an art package.
- Viewing the assets in RenderWare Visualizer by clicking on the view button or choosing the view menu option from the exporter plugins.
- Tweaking as necessary in the art package and viewing again.

The exporters are also capable of exporting the legacy `.bsp`, `.dff`, `.anm`, and `.dma` formats

Platforms

Just as RenderWare Graphics is available on many different platforms the RenderWare Visualizer works on many different platforms. RenderWare Visualizer views artwork on your PC and your target platform. This enables you to change the menu options on your PC and view the changes on your target platform.

Other Documentation

- 3ds max Reference Guide
- 3ds max Tutorials
- Maya Reference Guide
- Maya Tutorials
- Technical Artist Guide
- Optimize Static Geometry White Paper

2. Setup RenderWare Visualizer

RenderWare Visualizer is a suite of applications that can be run stand-alone or from 3ds max, Maya and RenderWare Graphics' FXEditors for GameCube and Xbox. It can be found in `viewers\visual` in the RenderWare Graphics directory.

The RenderWare Visualizer suite of applications consists of:

- `launcher.exe` – the launcher loads exported resources saved as `.rws`, or `.rf3` and sends them to viewers for viewing on different platforms. It also allows you to set up and edit connections to those platforms via the embedded Connection Editor.

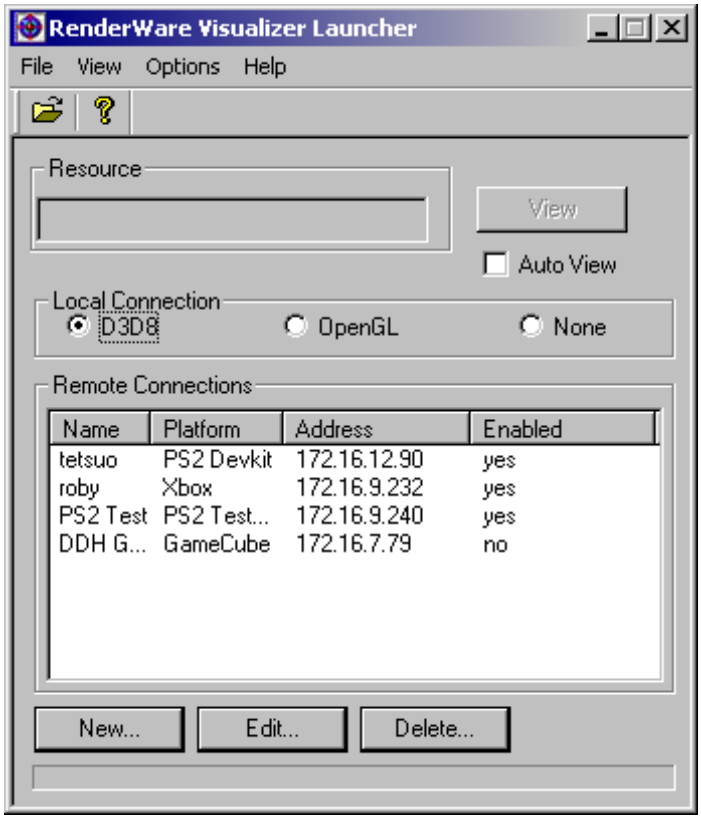
The launcher can also load the legacy formats `.dff` and `.bsp`. The launcher will attempt to locate textures or animations that are used by these files.

- `control.exe` – the control panel sets options for viewing `.rws`, `.dff` and `.bsp` files on all active viewers on all platforms simultaneously across a network. Viewing options include display of platform specific performance metrics, tri-strips and world sector visualization.

- `visualizer_d3d8.exe` / `visualizer_opengl.exe` / `visualizer_xbox.xbe` / `visualizer_sky2.elf` / `visualizer_gcn.elf` - the Visualizer is the viewer "host" application which accepts incoming resources and view options from `control.exe` over a network. The resources are rendered on the target platform.

To access the Launcher:

Start→Programs→RenderWare→Graphics→ Visualizer



3ds max, Maya and RenderWare Graphics' FXEditors also embed a connection editor and provide Launcher-like functionality from within the applications.

This section details the setup of connections on all supported platforms. Setting up from the launcher is described. Setting up connections within the modeling packages or FXEditors is very similar. The settings are saved in the same place but the interface may be different.

Requirements

RenderWare Visualizer requires Windows 2000 or later.

Windows 95/98/ME are not supported.

Target Connection Settings

RenderWare Visualizer connects to target platforms based on the settings entered in the connection editor. The connection editor interfaces for each application capable of launching the Visualizer are described in the sections following.

D3D8 & OpenGL Connections

1. Open Launcher.
   Start→Program Files→RenderWare→Graphics→Visualizer

2. Select a local connection for D3D8 or OpenGL. Local D3D8 and OpenGL connections will automatically spawn viewers and close them on disconnection.

3. Close *Launcher*.

Remote D3D8 and OpenGL connections are also possible. Simply specify the IP address of the remote machine in the connection editor and start the `visualizer.exe` on the remote machine. This instance of the viewer can be left running for multiple connections and disconnections.

GameCube Connections

The GameCube Visualizer Viewer comes with startup batch files for both GDEV and DDH development kits. The steps are:

1. Copy the batch files from `viewers\visual\GameCube` in the RenderWare Graphics directory to the host PC attached to the GameCube development kit.

2. Links to the `ddh_start_visualizer` and `gdev_start_visualizer` files should be placed in the startup folder on the GameCube development kits.

3. Use the appropriate 'start' file depending upon the variety of dev kit. These files start two programs:

   a. The first is the Visualizer host, which runs on the GameCube. It waits for resources to be sent via the USB link.

   b. The second is the `GameCubeCommsServer`. It forwards external connections and data from the network to the Visualizer host via the USB link.

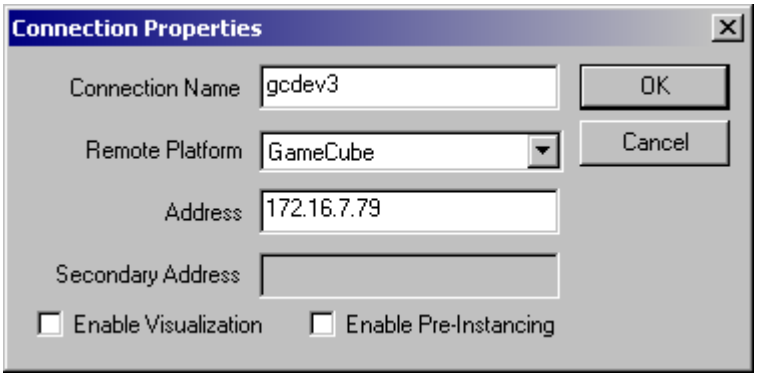
   Remote clients will connect to the GameCube Visualizer Viewer by talking to the `GameCubeCommsServer` running on the development kit.

4. The Visualizer splash screen is displayed with the IP address required.

5. To setup Visualizer
   Start→Programs→RenderWare→Graphics→SDK→Visualizer

   This runs the RenderWare Visualizer Launcher.

6. Click on the *New* button to create a new remote connection for GameCube. Enter a memorable *connection name*, choose the *remote platform* and enter the IP *address* from the splash screen.



7. To view artwork on DDH kits you will need to "Enable Visualization" in the Connection Properties dialog.

GDEV Notes

The Visualizer binary is kept in the 'bin' subdirectory as required by the GDEV kit. On this platform, the containing directory gets locked during execution of the binary.

Resources are transferred to the Visualizer via the network link, but the GDEV still needs an environment variable set up with a temporary path, for example:

```
setodenv dvdroot c:\temp\gdev
```

GameCube Trouble Shooting

Try rebooting the development box itself. Some applications may leave it in a state in which communication is impossible.

Bad USB connections are the cause of many communications problems. It is possible for connections to work briefly enough to allow artwork transfer, then fail before interaction with the Windows-based controls can take place.

Be sure to check all cables, and also try disabling then re-enabling the USB port from the Windows Control Panel. This facility is accessed through the 'System' item, then the 'Hardware' tab and finally the 'Device Manager' button. Options for the USB will appear near the bottom of the list.

In these conditions `GameCubeCommsServer` is often left running uselessly in the background. Therefore it hinders further connection attempts. The 'start' batch files attempt to shut this process down, but should they fail it must be terminated via the Task Manager.

PlayStation 2 Connections

Requirements

To connect to a PlayStation 2 Development Kit or Test Station you require either:

- a Sony broadband adaptor

or

- a supported USB Ethernet adapter

A list of supported adapters is available on the Sony Development Websites.

Note that you can not use both a broadband adapter and a USB Ethernet adapter simultaneously. This configuration will cause the Visualizer to lock up shortly after startup.

Setting up a PlayStation 2 Test Station

1. Burn the provided boot cd ISO image in `viewers\visual\sky2\cd_iso` in the RenderWare Graphics directory to CD-R media

In general to burn CDs for PS2 you should use Sony specific software and hardware.

2. Make sure you have the USB link connected.

3. Put the CD-R in the Test Station and reset it. After a few moments, the Visualizer splash screen will display with the PS2 Test Station's IP address.

4. Open *Launcher*.
Start→Programs→RenderWare→Graphics→Visualizer

5. Under Remote Connections, click *New...* The Connection Properties dialog opens.

6. Enter something memorable under *Connection Name*, copy the IP address into the *address* field, select *PS2 Test Station* from *the Remote Platform* drop down and tick the *Enabled* box. *OK* the dialog to close it.

7. Select a `.rws`, `.dff` or `.bsp` test resource and click *View.* Your test resource should now display on the PS2 Test Station.

Setting up a PlayStation 2 Devkit

The PlayStation 2 uses the ports as follows:

- The Ethernet port is used to transmit and run the Visualizer program. This port is used as the secondary static IP (or machine name) address.

- The USB port receives the scene data. This port is used as the primary IP address.

The instructions below should get Visualizer working for you.

1. Make sure you have the USB link connected before you start.

2. Open *Launcher*.
Start→Program Files→RenderWare→Graphics→Visualizer

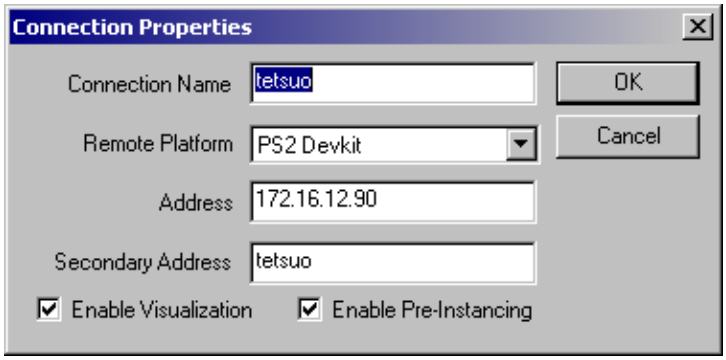
3. Under Remote Connections, click *New...*. The Connection Properties dialog opens.

4. Enter something memorable in *Connection Name*.

5. From the *Remote Platform* drop down select "PS2 Devkit".

6. The *Secondary Address* field is now enabled. Enter the name or the Ethernet address of the Devkit here. This address is used to automatically launch the Visualizer on the Devkit.

7. Tick the *Enabled Visualization* box, and click OK to close the dialog.



8. Select a `.rws`, `.dff` or `.bsp` test resource and click *View*. The Visualizer should launch and display a splash screen on the Devkit. The IP address is displayed on the splash screen.

   The IP address is used for communication over the USB link.

9. Control.exe will display a message stating "Unable to connect to *Your Connection Name Here*". Click OK on the message. Close the Control application.

10. Select the PS2 Devkit connection. Click *Edit...* to edit the connection properties.

11. Enter the USB link's IP address in the *Address* field.

12. Click OK to close the Connection Properties dialog.

13. Click *View* and your test resource should now display on the Devkit.

Specifying a fixed IP address for PlayStation2 USB adapter

By default Visualizer uses DHCP to obtain its IP addresses on the PlayStation2. If you prefer a fixed IP address instead, for a devkit, you can specify this by editing the file `IFC000.CNF` in viewers\visual\sky2\modules in the RenderWare Graphics directory. For example

```
#
#
# interface configuration sample for USB-Ethenet
#
# $Id: ifc000.cnf,v 1.1 2000/12/23 05:09:36 tetsu Exp $

type eth # only USB-Ethernet I/F

#dhcp # use dhcp
#-address # do not specify IP address
#-netmask # do not specify Subnet Mask

#-dhcp # do not use dhcp
address 172.16.11.36 # set IP address (your fixed IP here!)
#netmask 255.255.255.0 # set Subnet Mask
#route add default gw 0.0.0.1 # set default gateway
#nameserver add 0.0.0.1 # set primary name server
```

Next time you launch visualizer, the change should take effect and you'll need to repeat the set up process described above.

For a PlayStation2 Test Station, you need to perform the same edits to `IFC000.CNF`, in `viewers\visual\sky2\cdfiles` in the RenderWare Graphics directory and recreate the boot CD (see below).

Re-creating a PlayStation2 bootable CD-R image

After changing `IFC000.CNF`, you'll need to recreate the CD image and burn a new CD. All the necessary component files are provided in `viewers\visual\sky2\cdfiles` in the RenderWare Graphics directory.

You will need to obtain Sony's CD_DVD-ROM Generator (the version we use is 1.31). Look on the PlayStation2 developer network site https://www.ps2-pro.com/projects/cdgen for details of this product.

1. Start CD_DVD-ROM Generator.

2. Click the *Volume* tab and enter SLPS 66666 under *Disc Name*, CSL under *Producer Name* and CSL under *copyright holder*. *License Area* is ignored.

3. Click the *Directory* tab, then drag and drop the following files onto the window:

```
AN986.IRX
DEV000.CNF
DEV001.CNF
ICON.SYS
IFC000.CNF
IFC001.CNF
INET.IRX
INETCTL.IRX
IOPRP254.IMG
NET000.CNF
NETCNF.IRX
PADMAN.IRX
RWSCOMMS.IRX
SIO2MAN.IRX
SLPS_666.66
SYSTEM.CNF
SYS_NET.ICO
USBD.IRX
```

4. Drag and drop the following files onto the window. It's important these are added at the end of the image, or the disc will not work when used.

```
PADDING.1
PADDING.2
PADDING.3
```

5. If you have a Sony burner, you should be able to record as usual. If not, you can export an image that can be turned into an ISO image which can be burned with other burners. To do this, click *Record*, and save a temporary `.iml` file somewhere. Then a dialog called "CD/DVD-ROM Recording Unit Controller version 1.31" should open. Go to *File→Export* to get a "Export image file" dialog and give it a name and save it somewhere as an `.000` file. This `.000` file can be turned into an ISO image with the command line app `cdvd2iso.exe`.

Xbox Connections

Xbox development kits maintain two IP addresses labeled D and T in the development kit settings. These are displayed on the Xbox dashboard - you can cycle the display between them by pressing the black button on the Xbox controller. To setup up the Visualizer to run on Xbox:

1. Open *Launcher*.
   Start→Program Files→RenderWare→Graphics→ Visualizer

2. Under Remote Connections, click *New…*. The Connection Properties dialog opens.

3. Enter something memorable in *Connection Name*.

4. From the *Remote Platform* drop down select "Xbox".

5. Enter the "T" address in the *address* field.

6. Enter the "D" IP address in the *secondary address* field.

7. Tick the *Enabled* box, and click OK to close the dialog.

8. Select a `.rws` test resource and click *View*. The Visualizer should be automatically downloaded to the Xbox, launched and display the resource on the Xbox.

The Visualizer Viewer can also be started manually on the Xbox development kit by copying `visualizer.xbe` (found in `viewers\visual\Xbox` in the RenderWare Graphics directory) to `xe:\visualizer\visualizer.xbe`. It can then be started from the dashboard or a remote PC running `xbreboot`.

3. Using Launcher

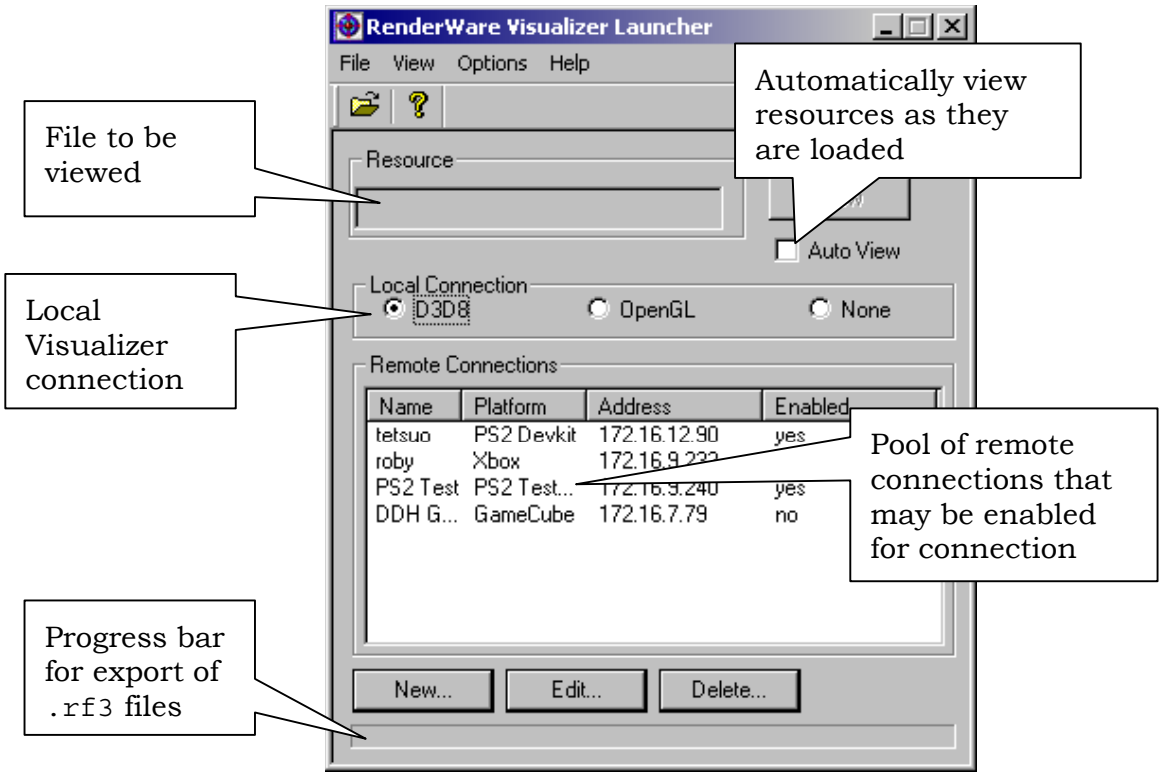
The Launcher loads exported resources saved as `.rws` files and sends them to viewers for viewing on different platforms. Given an appropriate template file, the Launcher can export `.rf3` files and view the resulting `.rws` file.

The launcher can also load the legacy formats `.dff` and `.bsp`. The launcher will attempt to locate textures or animations that are used by these files.

The Launcher also allows you to set up and edit connections to those platforms via the embedded Connection Editor.

Start→Programs→RenderWare→Graphics→Visualizer or 

Opens launcher.exe



Menus

The menus at the top of the Launcher provide a convenient way to access some common options.

File

Allows a specific file to be opened.

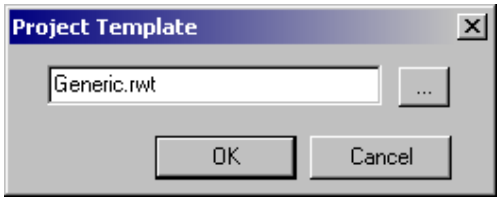
Provides a list of recently opened files.

View

Can show or hide the application toolbar.

Options

Displays up a dialog box that allows an `.rwt` template to be selected.



These files may be found in the `export/bin/template` directory in your RenderWare distribution. They are used by the Launcher (and the 3ds max and Maya exporters) to customize the export of `.rf3` files to `.rws` files.

For more information refer to the Technical Artist Guide.

Help

Can bring up the 'About' dialog box that lists the program's basic information.

Resource

File the `.rws` or `.rf3` file to view in RenderWare Visualizer

View views the file specified

Auto View automatically views a file when a file is opened in the launcher.

Connections

The Connection Editor is divided into *Local Connection* and *Remote Connections* so that you can easily view files on different platforms. Local Connection can be used to view D3D8 and OpenGL on your PC whereas Remote Connections can be used to view files on GameCube, PlayStation 2 Test Station, PlayStation 2 DevKit, Xbox, or a remote PC on D3D8 or OpenGL. Any number of remote targets can be configured, and even launched simultaneously.

Buttons

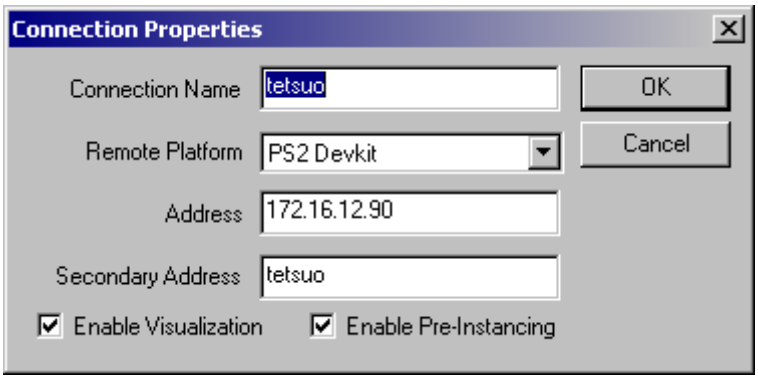
New... creates a new connection.

Edit... edits the selected connection. A connection is selected by clicking on the Connection Name.

Delete... deletes the selected connection. A connection is selected by clicking on the Connection Name.

Connection Properties

For a more detailed explanation about setting up connection properties refer to the *2 Setup RenderWare Visualizer* section.



Connection Name the name of the connection to be made.

Address the address for the target platform.

Secondary Address for the target platform.

Remote Platform the remote target platform.

Network Timeout the number of milliseconds to wait for timeouts. This option is currently unused.

Enable Visualization is used to state which platform viewers will display the RenderWare Export Nodes when the View button is pressed. When you click on a View button in RenderWare Visualizer Launcher, viewer.exe is run, the registry is accessed and connections are made to the enabled target platforms. The .rws, `.bsp` or `.dff` file is then displayed in RenderWare Visualizer.

Enable Pre-Instancing controls whether this connection will be used to generate platform specific data when an `.rws` file is exported with the 'Pre-Instance Data' option enabled. If assets are not being exported to `.rws` files or the 'Pre-Instance Data' option is not enabled in the exporters then this setting has no effect. See the `MayaReferenceGuide.pdf`, `3dsmaxReferenceGuide.pdf` or `TechnicalArtistGuide.pdf` documents in `doc/exporters/artists` in the RenderWare Graphics directory for more information on how pre-instancing works.

4. Using RenderWare Visualizer

The RenderWare Visualizer can be accessed from 3ds max, Maya and the stand-alone Launcher as mentioned above. This section describes the controls within the RenderWare Visualizer.

When the viewer starts it uses the information that the Connection Editor has saved into the registry about the local and remote controls and runs the viewer on all of them. This loads the Visualizer Viewer and the Control Panel on your workstation and sends your artwork to any targets that you have enabled in the Connection Editor.

The Control Panel contains the menus and control interface toolbar for the viewer, while the Viewer itself is a separate preview window that will display your selected artwork if you have selected a local connection.

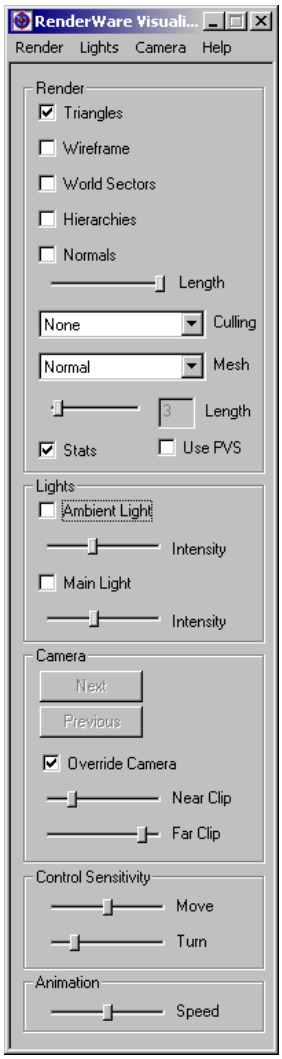
The RenderWare Visualizer

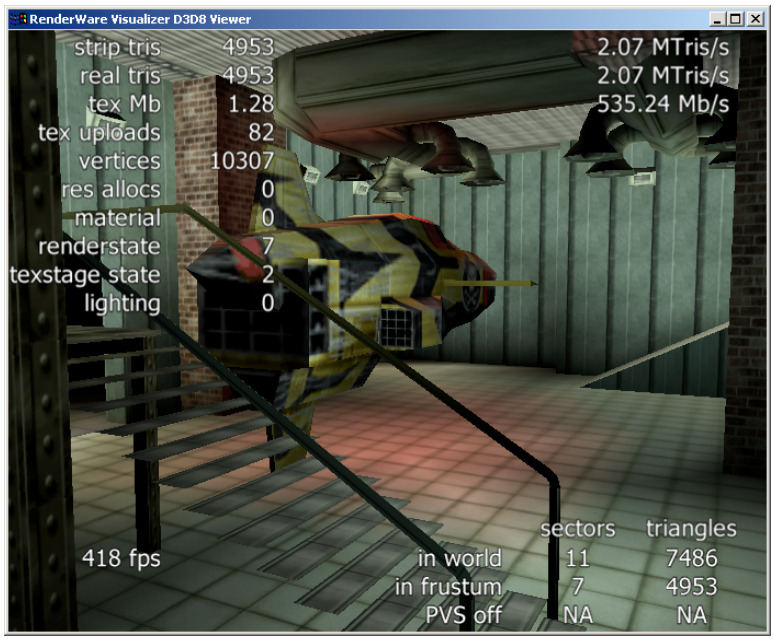
The RenderWare Visualizer loads `.rws` or `.rf3` files into the Viewer.

If files are in the legacy `.anm`, `.bsp`, `.dff`, or `.dma` format they are converted to temporary `.rws` files on load. The file format conversion takes place at view time.

Textures required by `.bsp` and `.dff` files are embedded within the `.rws` files if the textures are in a directory with the same name as the RenderWare Graphics file format or in a directory called textures. The embedded textures are automatically converted to platform independent textures on conversion to `.rws`.

Only one world can be viewed at a time, even though the exporters can create `.rws` files with more than one world in them. Attempting to view such a file will only display the first world in the Visualizer.



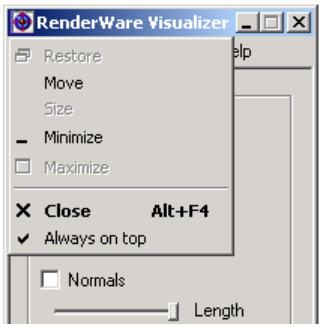


Control Panel *Visualizer Viewer*

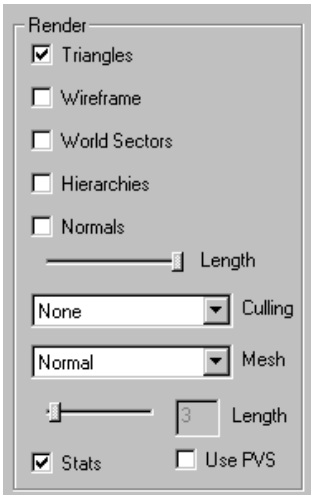
Control Panel

The RenderWare Visualizer Control Panel provides a number of controls that toggle key display information (triangles, wireframe, world sectors, hierarchies, normals, culling, mesh, statistics), and manipulate lighting (ambient, main) and cameras selection (free, near and far clip), to enable you to thoroughly review your content on your target platform(s). You may also override any cameras present in the scene, and use the mouse (for PC) or the appropriate console controller to fully preview your artwork.

The control panel has an "Always on top" option.



Control Panel Render Options



Triangles

Toggles rendering of world triangles (solid mode).

Wireframe

Toggles wireframe rendering of world triangles.

World Sectors

Toggles rendering of world sector bounding boxes.

Hierarchies

Toggles rendering of hierarchies.

Normals

Toggles rendering of vertex normals. The normals length can be scaled.

Rendering the world can be controlled via the control panel. It can be rendered in solid or wireframe, with vertex normals, with world sector bounding boxes, or in any combination of the above. For wireframe, vertex normals and world sector boxes, there is a coloring scheme.

Wireframe rendered in cyan.

Vertex Normals rendered in green.

World Sectors rendered in yellow.

Hierarchies rendered in red.

Wireframe and Vertex Normals only work on static geometry.

Culling

This control allows you to set whether *front*, *back* or *no faces* are culled out when RenderWare Graphics renders the scene. For most artwork the default value of *back* should be fine. If you're using a specialized tri-stripper which ignores winding order you'll need to set the face culling to *none*.

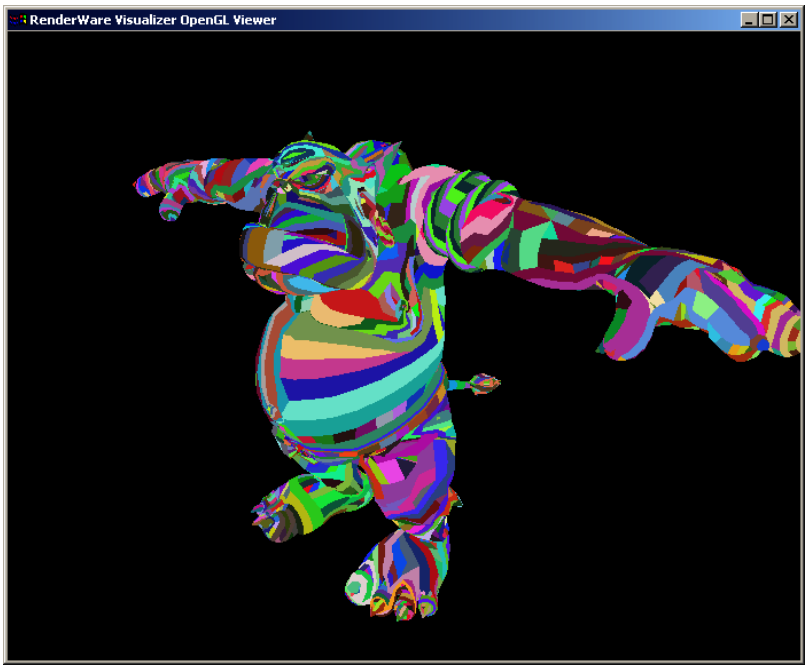
Some PS2 pipelines do not support backface culling and will ignore this setting.

Mesh

The mesh drop down lets you render your scene using a different color for each mesh. You should find there is a separate mesh for each material. Since the frequency of mesh changes affects the efficiency of tri-stripping and therefore your runtime performance, it is important to be aware of the meshes in your scenes and attempt to minimize the number that are generated.

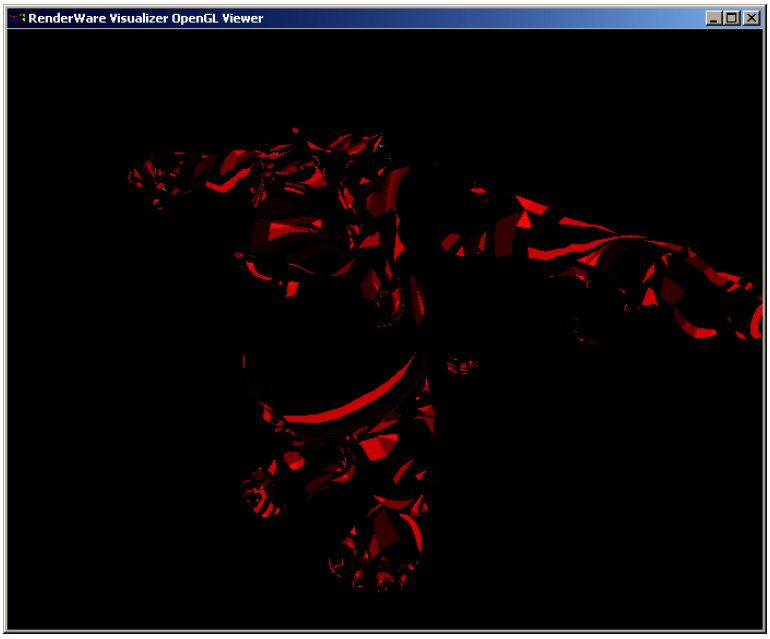
Tristrips

The number of tri-strips in your scene greatly affects the performance at run-time. The visualization tools built into the viewers enable you to assess the quality of the tri-stripping in your artwork and identity areas where improvement is possible. Each tri-strip is rendered in a different color. If you need guidance on improving the tri-stripping in your scenes please refer to the RenderWare Graphics Artist Guides for 3ds max and Maya.



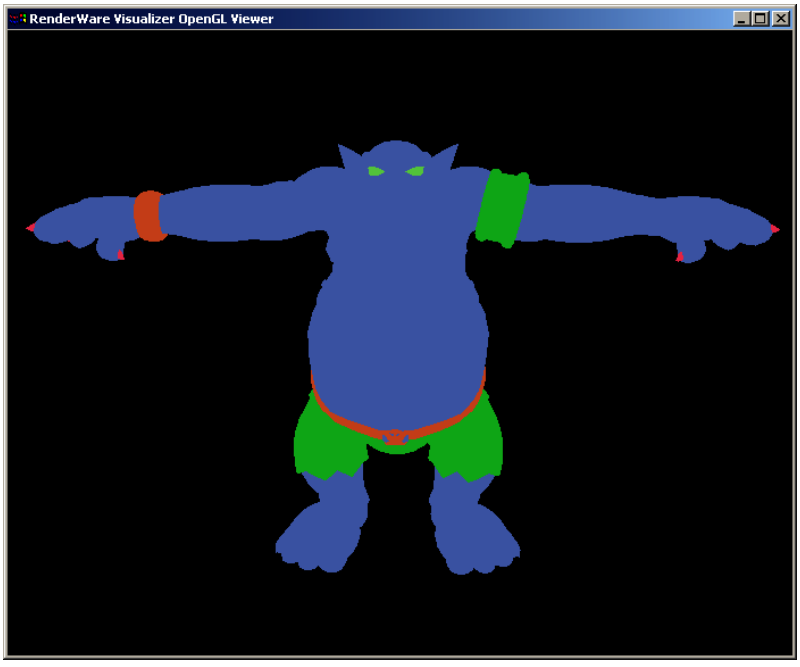
Tristrips No Degenerates

Setting the tri-strip lengths will render your scene in a mode that allows you to visualize the lengths of tri-strips. Tri-strips of length less than or equal to the tri-strip length setting will appear bright red and the longest dark red. Tri-strips longer than the tri-strip length setting will appear in black.



Meshes

Triangles assigned distinct materials are rendered with a distinct color for each material.



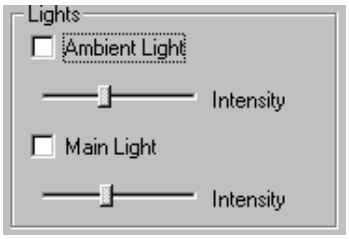
Use PVS

If a scene contains PVS data, this toggle enables or disables using it during rendering. This allows you to compare the performance with and without PVS rendering.

Statistics

Various performance related statistics are displayed by ticking the *Stats* box. Cross platform and platform statistics are available. Studying these should enable you to improve the performance of a problematic piece of artwork, or identify under utilized areas of the hardware that you could be using for additional detail, but aren't.

Control Panel Light Options

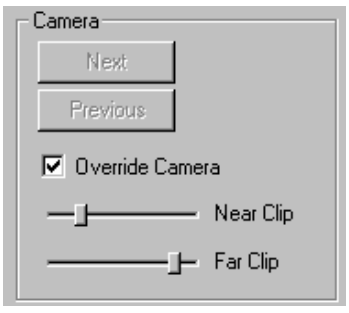


The ambient light and main light controls allow you to turn on and off the two lights in the scene. The related intensity controls give you fine control over the brightness of the lights.

The Visualizer attempts to locate appropriate ambient and main lights from within the scene itself. If it finds them, the intensity and on-off controls will use these lights.

If lights are not found in the scene, the control panel will add an ambient or main light if required. Both lights are white in color.

Control Panel Camera Options



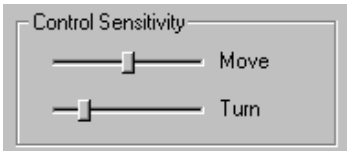
The camera controls give you control over the camera parameters used in the viewers. These values, along with the camera position, should default to values that give you a good view of the entire world or clump.

The next and previous camera buttons move you between cameras in your artwork.

The override camera option allows you to move around your artwork.

Depending on the scene you may find that adjusting the near clip plane can reduce Z bleeding between polygons.

Control Panel Control Sensitivity Options



The control sensitivity options are used to customize user control of the mouse and keyboard speed.

The move option enables control of the cursor key speed when moving around the viewer using the keyboard.

The turn option allows you to control the turning speed of the mouse when turning in the viewer.

Control Panel Animation Speed Options

The animation speed control option is used to adjust the speed of any animations that are playing in the viewer.

The middle of the slider is real-time; moving the slider bar to the right increases the speed of animation and moving the bar to the left slows it down.

Moving the bar fully to the left stops the animation.

The bar 'sticks' at the middle, which is exactly real-time speed. At this speed, there is no speedup or slowdown of the animation.

RenderWare Visualizer Viewer

The Viewer displays the assets and performance information about the assets.

Navigation Controls

Keyboard Shortcuts

The navigation controls are intuitive. The keyboard and mouse shortcuts to move around the viewer are:

Left click & drag	Mouse Look - position static, but changing view

	position
Right click	Activates and deactivates orbit mode. To activate orbit mode, depress the right mouse button and keep it depressed. Move the mouse or use the keyboard for navigation. Release the right mouse button to deactivate orbit mode.
→ or D	Strafe left
← or A	Strafe right
↑ or W	Forward
↓ or S	Backward
Esc	Exit application if in the Viewer window

The yellow cross in middle of viewer is the point to orbit around.

PlayStation2 Controls

Left analog stick	Change view direction
Right analog stick or	Move forward and backward
Triangle and Cross	
Left shoulder 1 or Square	Strafe left
Right shoulder 1 or Circle	Strafe right
Right shoulder 2	Activates and deactivates orbit mode. To activate orbit mode, press and hold R2. Move the left analog stick to rotate and the right analog stick to zoom. Release R2 to deactivate orbit mode.

Xbox Controls

Left analog stick	Change view direction
Right analog stick or	Move forward and backward
A or B	
Left Trigger	Strafe left
Right Trigger	Strafe right
X	Activates and deactivates orbit mode. To activate orbit mode, press and hold X. Move the left analog stick to rotate and the right analog stick to zoom. Release X to deactivate orbit mode.

GameCube Controls

Left analog stick	Change view direction
Right analog stick	Move forward and backward
A	Activates and deactivates orbit mode. To activate orbit mode, press and hold A. Move the left analog stick to rotate and the right analog stick to zoom. Release A to deactivate orbit mode.

Statistics

The display of statistics may be enabled from the Control Panel.

Cross Platform Statistics

In bottom left corner:

- Number of frames per second being rendered (updated every second).

In bottom right corner:

- Total number of sectors and triangles in world.
- Total number of sectors and triangles currently inside the camera's frustum.
- Total number of sectors and triangles potentially visible, according to the RenderWare Graphics PVS system if the world has PVS data and *use PVS* is enabled. If there is occlusion within the scene and world sectorization is done well, these numbers should be significantly smaller than the numbers inside the frustum.

Enabling world sector rendering and checking the world sector culling statistics above should give you a good idea of how good your world sectorization (if any!) is, and whether changing it can improve frustum culling or PVS to gain a performance boost.

PS2 Specific Statistics

- The names of the active rendering pipelines are cycled across the top of the screen. Visualizer will attempt to select the fastest pipeline available to render your scene. Changing scene properties such as reducing the number of lights can enable Visualizer to use a faster pipeline for better performance. For example, the G3x family of pipelines is faster than the G3 family of pipelines because they are specially coded to work with fewer lights and perform only triangle frustum culling rather than true clipping.

- Strip tris - The number of triangles submitted, including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.

- Real tris - The number of triangles submitted, not including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.

- tex Mb - Megabytes of texture uploaded, on average, per frame. Upload rate in megabytes per second is displayed to the right of this figure.

- tex uploads - the number of textures uploaded, on average, per frame.

- vertices - the number of vertices submitted, on average, per frame.

- resource allocs - the number of resource allocations on average, per frame.

- VU1 - what percentage of the time VU1 was busy, as determined by approximate sampling. This should indicate to you how much geometry processing capacity you are using. Enabling a faster/simpler pipeline, using better tri-stripping, or just using less vertices can help reduce this figure.

- DMA1 - percentage of the time DMA1 was busy, as determined by approximate sampling. RenderWare Graphics uses DMA1 for upload of renderstate and geometry. DMA1 can sometimes also be used for texture upload when necessary.

- DMA2 - percentage of the time DMA2 was busy, as determined by approximate sampling. RenderWare Graphics uses DMA2 for texture upload. Reducing the size or number of textures or texture changes in a scene can help reduce this figure.

- CPU - what percentage of the time the CPU was busy, as determined by approximate sampling. (Note that Visualizer itself takes about 6% of this on average, to collect and display metrics and do network communication, your game will have different overheads.) Large numbers of world sectors, particle systems, complex hierarchy and morph animation can use up available CPU. Adjusting the complexities of these quantities can reduce CPU usage.

If you load a scene running at less than 60 FPS, comparing VU1, DMA1, DMA2 and CPU usage should give you an idea of where the bottleneck lies and what you can change about the art or rendering options to fix it. Note that when running at less than 60 FPS, the busy percentage of the frame-rate-dropping culprit may actually drop to less than 100%. Try moving the camera or changing rendering options to cross the threshold of rendering at 60 FPS to get a true idea of where the bottleneck lies.

Xbox Specific Statistics

- Coverage - Percentage of the screen filled by pixels that passed the z-test, i.e. the pixels that were drawn requiring all the work to shade them. The rate in millions of pixels drawn per second is displayed to the right of this. The coverage gives you some idea of how much overdraw you have in the scene. If it's greater than 200%, you can probably benefit by using PVS to reduce your fillrate needs.

- Vertices - total number of vertices or vertex indices submitted to the hardware, on average, per frame.

- Submits - total number of drawing calls submitted to the hardware, on average, each frame. In general you get better performance by submitting larger batches of vertices in a few draw calls than submitting many small batches of vertices in many draw calls, so ideally you'd like this number to be small. However, material changes such as texture and color, and geometry changes such as to a different world sector or atomic each require a submit. So you can get better performance by using fewer (but perhaps bigger) textures on an object, or creating fewer, but bigger world sectors.

- Texture sets - total number of times a texture had to be bound to the hardware. Often a significant contributor to number of submits, above. If you have many small textures in a scene, replacing them with one large texture will reduce this cost.

- Stream sets - total number of times a vertex buffer had to be bound to the hardware. Often a significant contributor to number of submits, above. If you have many atomics or world sectors in a scene, grouping them together as bigger atomics or world sectors will reduce this cost.

- GPU geometry - percentage of the time the geometry stage of the GPU was busy, on average as determined by approximate sampling. The number of vertices submitted, how good your tri-stripping is and how expensive the vertex shaders used are all contribute to this figure. Any stalls waiting for the rasterizer are also included in this number.

- GPU rasterizer - percentage of the time the rasterizer stage of the GPU was busy, on average as determined by approximate sampling. This includes texture fetching, pixel shading, alpha blending and z buffer operations. Compressing your textures better, reducing their size and reducing overdraw in the scene can help reduce this figure.

- CPU - Amount of time the CPU was busy, judging by the time "left over" after drawing the scene waiting for the page flip. Large numbers of world sectors, particle systems, complex hierarchy and morph animation can use up available CPU. Adjusting the complexities of these quantities can reduce CPU usage.

These counters are based on hardware counters described in more detail the white paper "Xbox Graphics Performance Tuning Strategies" on `http://xds.xbox.com`.

GameCube Specific Statistics

- Coverage - Percentage of the screen filled by pixels that passed the z-test, i.e. the pixels that were drawn requiring all the work to shade them. The rate in millions of pixels drawn per second is displayed to the right of this. The coverage gives you some idea of how much overdraw you have in the scene. If it's greater than 200%, you can probably benefit by using PVS to reduce your fillrate needs.

- Vertices - total number of vertices or vertex indices submitted to the hardware, on average, per frame. The rate in millions of vertices processed per second is displayed to the right of this.

- Texels - total number of texels processed. The rate in millions of texels processed per second is displayed to the right of this.

- Strips tris - total number of triangles processed, including degenerates. The rate in millions of triangles processed per second is displayed to the right of this.

- Clipped vertices - total number of vertices that were clipped, on average per frame.

- GP tex - of the total time to draw the scene, the percentage of that time the texture unit was busy.

- GP clip - of the total time to draw the scene, the percentage of that time that was spent clipping.

- GP transforming - of the total time to draw the scene, the percentage of that time that was spent transforming geometry.

- GP light - of the total time to draw the scene, the percentage of that time was spent lighting geometry.

Clipping, especially against the near clip plane, is expensive on GameCube and should be avoided by altering artwork and camera settings wherever possible. Changing world sectorization can help cull away sectors rather than clip them.

The GameCube hardware counters are somewhat odd in that you can only measure a small subset of them at any one time. So the results you see are a mixture of results from different frames, sampled in a round robin scheme. Therefore, holding the camera still for a few seconds before reading them will give more consistent figures.

D3D8 Specific Statistics

- Strip tris - The number of triangles submitted, including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.
- Real tris - The number of triangles submitted, not including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.
- tex Mb - Megabytes of texture uploaded, on average, per frame. Upload rate in megabytes per second is displayed to the right of this figure.
- tex uploads - the number of textures uploaded, on average, per frame.
- vertices - the number of vertices submitted, on average, per frame.
- resource allocs - the number of resource allocations on average, per frame.
- material - the number of material changes on average, per frame.
- renderstate - the number of render state changes on average, per frame.
- texstage state - the number of texture stage state changes, on average, per frame.
- lighting - the number of lighting state changes on average, per frame.

State changes of each kind can be costly on D3D8. Advice for reducing these similar to that in the Xbox section applies to PC based hardware T&L video cards.

OpenGL Specific Statistics

- Strip tris - The number of triangles submitted, including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.
- Real tris - The number of triangles submitted, not including degenerates, on average, per frame. The rendering rate in millions of triangles per second is displayed to the right of this figure.
- vertices - the number of vertices submitted, on average, per frame.
- resource allocs - the number of resource allocations on average, per frame.

5. Summary

The RenderWare Visualizer allows artwork to be quickly and easily previewed.

The artwork may be previewed either locally, on the PC being used. It may also be previewed remotely on an appropriately configured PlayStation2, Xbox or GameCube.

The Visualizer may be run from the 3ds max or Maya exporters.

The Visualizer may also be run from a standalone launcher. The Launcher supports the present RenderWare file formats, `.rws` and `.rf3`, and the legacy `.dff`, `.bsp`, `.anm` and `.dma` formats.

The details of the connections to be used by the Visualizer may be configured from the exporters, or from the Launcher application.

The Visualizer supports many different rendering, lighting, camera, movement and animation options. These are manipulated from a control panel that displays on the user's PC.

6. Additional Information

LOD

The level of detail changes by zooming in and out of the viewer.

Textures

Assets look for textures in a sub-directory with the same name as the asset and embeds the textures within the `.rws` file.

Window Size

Window size and position saved in the registry.

File Names

RenderWare Visualizer is not case sensitive and it can handle files and directories with spaces.