

RenderWare Graphics

Artist Guide

3ds max™ Reference Guide

Copyright © 2004 – Criterion Software Ltd.

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.

The information in this document is subject to change without notice and does not represent a commitment on the part of Criterion Software Ltd. The software described in this document is furnished under a license agreement or a non-disclosure agreement. The software may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or non-disclosure agreement. No part of this manual may be reproduced or transmitted in any form or by any means for any purpose without the express written permission of Criterion Software Ltd.

Copyright © 1993 - 2004 Criterion Software Ltd. All rights reserved.

Canon and RenderWare are registered trademarks of Canon Inc. Nintendo is a registered trademark and NINTENDO GAMECUBE a trademark of Nintendo Co., Ltd. Microsoft is a registered trademark and Xbox is a trademark of Microsoft Corporation. PlayStation is a registered trademark of Sony Computer Entertainment Inc. 3d studio max, character studio are registered trademarks and Discreet is a trademark of Autodesk/Discreet in the USA and/or other countries. Copyright © 2002 ImageMagick Studio LLC, a non-profit organization dedicated to making software imaging solutions freely available. All other trademark mentioned herein are the property of their respective companies.

Table of Contents

1. Introduction	7
1.1 What do the exporters support?	7
1.2 What's New in 3.7	7
3ds max 6 support	7
Visualizer Connection Wizard	8
Lightmap improvements.....	8
Background color	8
1.3 How is this document organized?	8
1.4 Other Documentation.....	8
1.5 Multi-platform	9
1.6 Software Supported.....	9
2. Installation and Setup	11
2.1 Files	11
2.2 RenderWare Exporter.....	11
2.3 Utilities Panel	12
2.4 Installing the Exporter Manually	12
Setting up the environment variables.....	13
Updating the registry	13
Copying the exporter to your plugins directory	13
3. Modeling Reference	15
3.1 Triangle Strips (Tri-strips)	15
Improving tri-stripping performance	15
3.2 Triangle Lists (Tri-lists)	17
3.3 Triangle fans (Tri-fans)	17
3.4 Connect Vertices	17
3.5 Scale and Clipping Planes	18
3.6 Patches	18
3.7 Negative Scaling	19
3.8 Collapse Edit Stacks.....	19
3.9 Normals and Smoothing Groups	20
Smoothing Groups.....	21
3.10 Hidden Objects	21
3.11 Level of Detail.....	22
3.12 Instances	22
3.13 Potentially Visible Sets (PVS)	22
Visualizer	23
PVS Editor	23
3.14 Hints.....	24
What are hints	24
Hint types	25
Creating hints	25

	Hint strengths.....	27
	Hint Example.....	27
3.15	Partitions	28
4.	Material Reference	31
4.1	Standard Material.....	31
	General Settings	31
	Transparency.....	33
	Bump and Environment Mapping.....	34
	Normal Map.....	37
	Texture UV Animation	37
4.2	RwMaterial	39
	RpMaterial Parameters Group	39
	MatFX Options Group.....	40
	Environment and Bump Mapping.....	41
	Dual Pass	41
	Blend Modes	41
	Platform Specific Extensions	44
	Toon.....	46
	LightMaps	46
	Normal Maps	47
	Viewport Manager	48
	Hardware Preview	49
4.3	Common Material Topics	50
	Texture Maps.....	50
	Texture Size	50
	Texture Repeats.....	50
	Texture Filter Modes	51
	UV Mapping Coordinates	52
	Using Cropped Textures	53
	Material Color vs. Vertex Color.....	54
	Double-sided Materials.....	54
	Extra Vertices from Multiple Materials	54
	UV Vertices	55
4.4	Attaching Geometry.....	55
5.	General Reference	57
5.1	Lighting	57
	Omni Lights.....	57
	Ambient Lights.....	57
	Prelights	58
	Duplicate Vertices	59
	Light Instances	60
	Dynamic Lights	60
	LightMaps	60
	Normal Maps (Xbox and D3D9 specific).....	66
5.2	Camera.....	69

	Background color	69
5.3	Animation.....	69
	HAnim Animation	69
	Animation Curves	69
	Rotation Keyframes	70
	Tension Continuity Bias (TCB)	70
	Linear Interpolation	70
	Automatically Creating Animated Assets.....	70
	Sub-Hierarchy Animations	71
	Animating the Root Transformation	73
	Morph Target and DMorph Animation	73
	Bone Limit.....	74
	Bone Tagging.....	75
	Customizing Object Properties	75
6.	Exporting.....	77
6.1	RenderWare Menu	77
	Export.....	78
	View	80
	Output Window	80
6.2	Advanced Settings.....	81
	Assets.....	82
	Asset Templates.....	90
	Project Templates.....	91
	Connections.....	92
6.3	Batch Exporter	95
6.4	Plugin Information	95
6.5	Utilities Panel	96
	RenderWare Exporter.....	96
	RenderWare Object Settings	97
Index		98

1. Introduction

RenderWare Graphics is a 2D and 3D graphics library. It is used by programmers to create real-time 3D graphics applications, such as computer games and simulations.

RenderWare Graphics has exporter plugins for 3ds max, Character Studio and Maya, which enable artists to create compatible artwork for programmers to place into games.

1.1 What do the exporters support?

The high performance exporters for 3ds max include support for:

- 3ds max version 4.x and 5.x.
- Bones Pro 2 and 3 and versions of Character Studio from 2.0 onwards.
- lights, cameras, meshes, patches and splines.
- hierarchical, blend shape, morph, native, skin and UV animation.
- inverse kinematics (IK).
- vertex prelighting.
- multiple UV sets.
- instancing.
- custom attributes.
- texture filter and addressing modes.
- material effects - bump, environment, normal maps and dual pass.
- RenderWare Graphics specific material with Xbox and GameCube hardware shader setup.

1.2 What's New in 3.7

3ds max 6 support

Exporters are now provided for 3ds max 6.

Visualizer Connection Wizard

This new wizard greatly simplifies the process of setting up remote connections for use by RenderWare Visualizer.

Lightmap improvements

The algorithms used when the *Export Native Lightmaps* option is enabled have been improved leading to better lightmap UV packing. It is also now possible to mix lightmapped and non-lightmapped materials in the same exported asset. Because the Render To Texture functionality in 3ds max 6 no longer supports rendering lightmaps into the lightmap slot of material viewport extensions we have also added support for the lightmap DX9 shader. These can now be exported as RenderWare lightmaps when the *Export Native Lightmaps* option is enabled.

Background color

The environment background color is now preserved when performing views using RenderWare Visualizer. The background color can also be added to exported files (see the Technical Artist Guide for more information on enabling export of this data).

1.3 How is this document organized?

This document is split into sections:

- Installation and setup of RenderWare Graphics for 3ds max.
- Modeling Reference
- Material References
- General References
- Exporting



3ds max release 5 screenshots are used throughout this document.

1.4 Other Documentation

- [3dsmaxTutorials.pdf](#) - The tutorials document is organized around a series of tutorials that take you through the basics of RenderWare Graphics as well as some of the issues you need to think about as you create 3D worlds. If you're new to RenderWare Graphics this is where you should start.

- [TechnicalArtistGuide.pdf](#) - This document is aimed at advanced/technical users (artists/programmers) of the RenderWare Graphics Exporters. It describes in depth a process of customizing export templates and export process itself.
- [OptimizeStaticGeom.pdf](#) - This document is a case study of how to optimize static geometry using knowledge of the PlayStation 2 architecture.
- Three viewers can be used to view artwork exported using the RenderWare Graphics exporters. The viewers are: RenderWare Visualizer; Clump View and World View. There are two viewer documents describing the controls and setup of these viewers [RenderWareVisualizer.pdf](#) and [ClmpviewWrldview.pdf](#).
- RenderWare Graphics has a range of documentation material aimed mainly at developers but still useful for the artist. After installation, take a look at additional documents in the docs directory. The User Guide in particular should be useful to you as it covers a lot of material relevant to the artist.
- RenderWare's Fully Managed Support Service (FMSS) contains RenderWare Graphics Art Examples, which are available for download. In the FMSS <https://support.renderware.com/>, click *Downloads* on the left of the screen.

PDF format: Most RenderWare Graphics documents are in PDF format, which is a self-contained document format from Adobe. You'll need to install the (free) Acrobat Reader to view and print these. In some cases the quality is better in the printed form than on-screen.

The RenderWare Graphics PDF documents have been designed to be printed double-sided.

1.5 Multi-platform

Direct3D, OpenGL, Sony PlayStation 2, Microsoft Xbox and NINTENDO GAMECUBE platforms are all supported. These platforms are very different which means that in some situations you'll need to adapt your working methods to suit the platform.

1.6 Software Supported

The examples and details in this document are based on 3ds max 4.x and 5.x, Character Studio 2.2, 3.0 and 4.0, and RenderWare Graphics 3.7. The tutorials assume that you have good knowledge of 3D animation concepts and are familiar with basic 3ds max operation. Windows refers to Windows XP or Windows 2000.

2. Installation and Setup

The installer copies the exporter files and sets up the Start Menu options. The installation program also creates a SDK folder which is mainly of interest to the programmers but which contains the viewers you'll need later.

2.1 Files

For your reference, the installed RenderWare Graphics files are as follows:

- `rwexp.dle` to `3dsmax\plugins`.

An additional set of common exporter files are installed in your main RenderWare Graphics directory and added to your path environment variable. The exact location depends on where you choose to install RenderWare Graphics to but the default is `c:\Rw\Graphics\export\bin`. The files stored here are as follows:

- `ExGrid.dll`, `ImgMagick.dll`, `RwAxCtrls.dll`, `RwComm.dll`, `RwExp.dll`, `RwExpMgr.dll`, `Rwgnull.dll`, `RwInstance.dll`, `RwMI.dll`, `RwRf3.dll`, `RwXML.dll` and `RwMax4.dll` or `RwMax5.dll`



3dsmax in this instance refers to the full path name of your 3ds max folder.


2.2 RenderWare Exporter

When you start 3ds max, the RenderWare menu will be available with the following options:

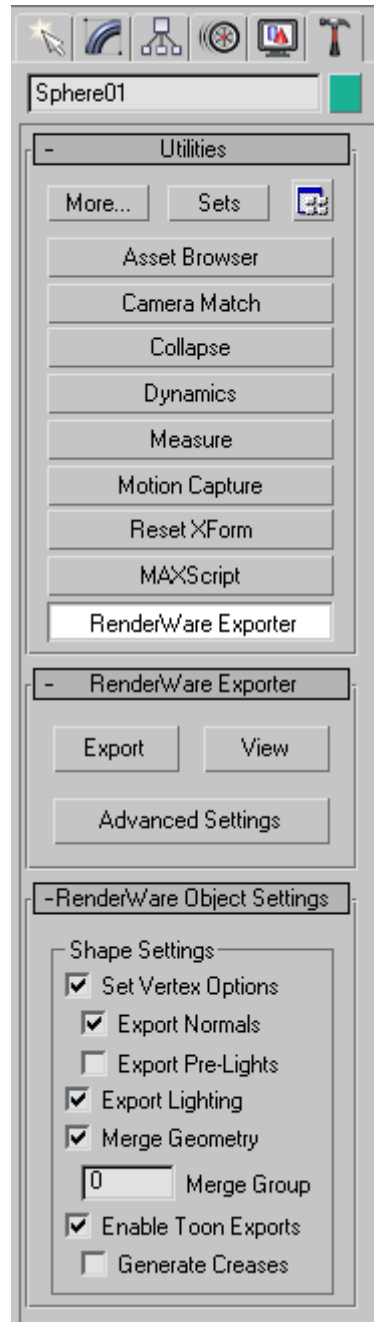
- Export RenderWare Assets
- View RenderWare Assets using Visualizer
- Advanced Settings
- Tools for Batch Exporter and Template Update
- Plugin Manager listing all the RenderWare plugin files used
- Help for all RenderWare Graphics exporter documentation

2.3 Utilities Panel

The RenderWare Exporter can be added to the Utilities Panel. To add a button:

1. In the Utilities Panel
2. Click on the Configure Button Sets button 
3. Increase the number of "Total Buttons" by one.
4. Click and drag "RenderWare Exporter" Utility onto the blank button

In 3ds max, RenderWare Object Settings can only be accessed from the Utilities Panel.



2.4 Installing the Exporter Manually

To install the exporter by hand, you need to setup your environment variables, update the Windows registry and copy the exporter into your 3ds max directory. This section tells you how to do this.

Setting up the environment variables

Setup your PATH environment variable with the directory path that contains the exporter DLLs: Rwg.dll, RwExp.dll, RwComm.dll, RwExpMgr.dll etc. (this is usually rw\graphics\export\bin):

1. Select Start -> Settings -> Control Panel.
2. Double-click on System, and then select the Advanced tab.
3. Click on the Environment Variables button, select PATH in the list of variables and then click Edit.
4. Add the path to the directory that contains that exporter DLLs (Rwg.dll, RwExp.dll, RwComm.dll, RwExpMgr.dll) to the list of paths, and then click Okay.

To test this configuration, open a new command prompt and type path. All your PATH settings will be displayed, including the one you added for the exporter. You may need to restart your computer, or at least log on and off, to ensure the registry is updated.

Updating the registry

The following COM controls should be registered using the Microsoft Register Server, regsvr32:

- ExGrid.dll, RWAxCtrls.dll and RwRf3Translator.dll, which are in the rw\graphics\export\bin directory.
- OutputPanel.dll which is in Program Files\Common Files\RenderWare\

Copying the exporter to your plugins directory

To add the exporter plugin to 3ds max, copy RwExp.dle into your C:\3dsmax\plugins\ directory.

3. Modeling Reference

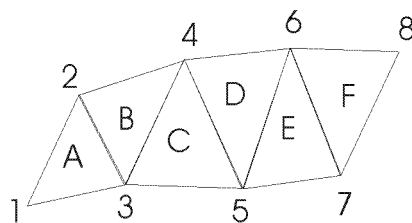
This section explains how you can model your artwork for RenderWare Graphics exporters so that you can obtain the best possible results. While there is an option of conditioning geometry for static worlds (see the [TechnicalArtistGuide.pdf](#)), it should not be seen as a replacement for good initial artwork.

Most of the general principles of good modeling for games apply to RenderWare Graphics. Examples are: minimizing vertices, keeping texture map sizes small, tri-stripping etc. There are a few specifics worth keeping in mind and these are listed below.

3.1 Triangle Strips (Tri-strips)

Tri-strips are an efficient way of storing geometry. Tri-strips consist of adjacent triangles which share vertices. Artwork created using tri-strips can reduce the number of vertex repetitions by a factor of three.

The series of adjacent triangles below use tri-stripping. Vertices 1, 2 and 3 draw triangle A. Triangle B requires only one vertex to define it, vertex 4, as it uses vertices 2 and 3 which have already been drawn.



Improving tri-stripping performance

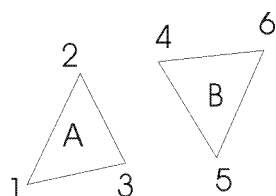
RenderWare Graphics calculates the number of triangles and the number of triangles actually processed by the hardware. The difference between these two statistics is the number of degenerate triangles inserted between the end of one tri-strip and the beginning of the next. To improve tri-stripping performance, and reduce the number of degenerate triangles, there are a few golden rules:

1. Avoid using tri-fans, or T junctions in meshes. These topologies are difficult to tri-strip. Where possible create surfaces using quads with each quad split into two in the same direction. Breaks in surface geometry will terminate tri-strips and this in turn will have an impact on rendering performance.

2. Use as few materials for each model as possible as tri-strips can not be created across material boundaries. Only one texture is allowed per material. Materials or normals applied to an individual triangle within a tri-strip, will break the tri-strip.
3. For single pass texturing, RenderWare Graphics vertices can only have one UV coordinate. UV coordinates should be shared on vertices wherever possible as vertices in the same position, with different UV coordinates, can not be considered for tri-stripping. This involves using the same texture on as many adjacent polygons as possible. Ideally, one texture is applied around a whole object, and the UV coordinates mapped onto the mesh. Texture coordinates should be laid out with tri-stripping in mind. A typical example is, a series of quads tiled with same texture, but with each polygon having its UV coordinates in corners of the texture. In RenderWare Graphics, the programmer should ensure that the texture is set to `rwTEXTUREADDRESSWRAP` and the texture coordinates of the quads continue across the repeating texture rather than starting at the beginning for each quad.
4. When applying lightmapping, the lightmapping plugin processes the geometry creating UVs for each triangle to use. The plugin tries to use UVs so that the lightmap textures are used as efficiently as possible. This means that these UVs may not be the same for two adjacent triangles (i.e. the shared vertices will have 2 UVs each). This will stop the tri-stripper from being able to create a single tri-strip from them. The lightmap creation code is designed to minimize this problem as much as possible, but there is a trade-off between good tri-stripping and small texture sets for the lightmaps.
5. RenderWare Graphics vertices can only have one normal. Where two vertices at the same position have different normals, these cannot be considered for tri-stripping, therefore it is important to smooth your models (especially if you are using pre-lighting, as normals are not used). If the models are built with vertices that can not be shared, tri-stripping will try to connect strips by building extra polygons between the disconnected ones. If the models are very disjointed, these extra polygons can add large amounts of data to a scene. In this instance, tri-lists may be more suitable
6. For untextured objects, or prelit objects where the vertex colors vary across the surface of the mesh, the color at shared vertices should be the same. Two vertices at the same position with different colors cannot be considered for tri-stripping. To obtain the best tri-stripping performance on prelit models created using 3ds max 5, set the option "mix vertex color" (*Utilities→More→Assign Vertex Colors* in 3ds max 5) in the vertex color 3ds max tool when the object uses pre-lighting

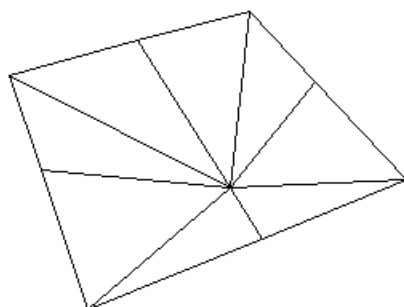
3.2 Triangle Lists (Tri-lists)

A tri-list is like a tri-strip but, instead of sharing vertices, a tri-list uses three vertices per polygon in every case. This is usually a lot less efficient than tri-strips if the models are well built, but if the models don't tri-strip very well then it can lead to improvements. This is because tri-stripping a bad model forces RenderWare Graphics to create lots of extra (degenerate) linking polygons.



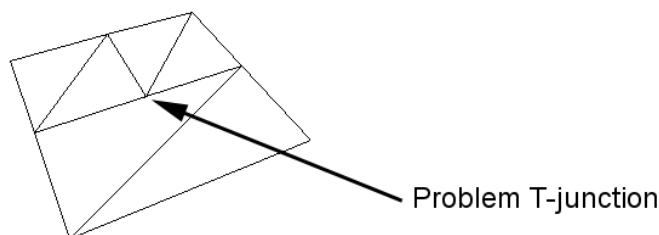
3.3 Triangle fans (Tri-fans)

A tri-fan is a way of creating a surface by placing a central vertex and having the individual polygons radiate from that. Consoles deal with these shapes slowly so try to avoid them in favor of a more grid like tri-strip approach.



3.4 Connect Vertices

In your models all the vertices should connect and there should be no gaps, overlaps or T-junctions. T-junctions, like the central vertex shown below, will cause problems along the horizontal seam.



To optimize geometry, all objects that are to be exported as a single *Static World* file should be welded together into one object.

3.5 Scale and Clipping Planes

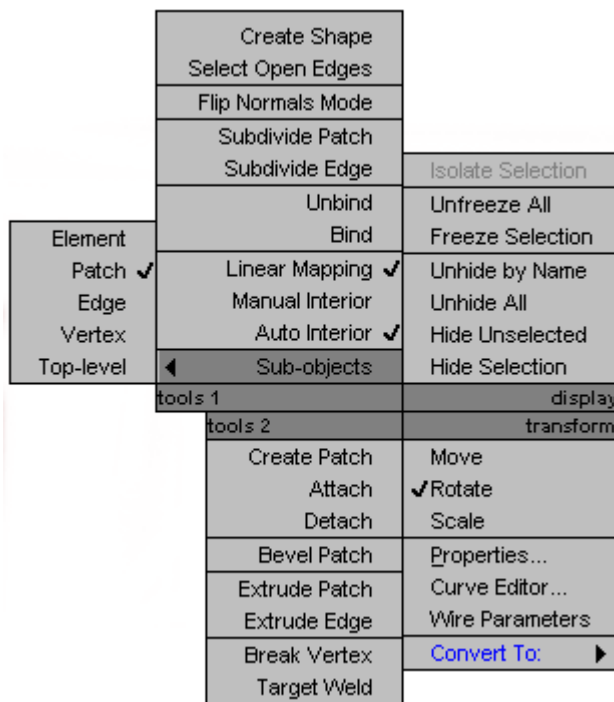
There are some hard-coded limits to where the clipping planes can be set in the viewers. This can cause problems with very large models. Generally, use a near/far clip ratio of not more than several thousand to one. Making the ratio any greater can lead to problem artifacts in the game (from the low z-buffer resolution). For example, using a near clip plane of 0.01 and a far clip plane of 10000 will probably cause problems, whereas using a ratio of 10:10000 should work well. Scenes should be scaled with this in mind. Note that the exporters can scale scenes during the export process.

3.6 Patches

Bézier Patches are exported as RenderWare Graphics RpPatchMesh objects in the *Animated Hierarchy* exporter. In the *Static World* exporter, they are tessellated and exported as part of the standard geometry. They are not converted to triangles first. Mapping in curved space for patches is *not* supported.

To obtain the same result in RenderWare Graphics, you should turn the following option on:

1. Select an editable patch.
2. Right click the selection and choose *Sub Objects* → *Patch*
3. Right click the selection and choose Linear Mapping from the Tools 1 quadrant.



The Exporter recognizes Bézier patches if they have been constructed from 3ds max's surface tools or converted to an editable patch from another 3ds max geometry object.

It is possible to model in patches and let the RenderWare Exporter automatically do the conversion by setting the *Export Patches* option, in the Advanced Settings Animated Hierarchy template, to FALSE. However, any skinning will not be exported in this case, but that should not be a problem since you can use the Turn to Mesh modifier before the Skin modifier in the stack. Also, if it's left unchecked, the geometry will be exported as polygonal geometry.

NURBS surfaces are not supported as separate type of objects and are tessellated in both exporters.



If a patch object is used, after changing the patch's vertex color the colors may be mixed. This is caused by the colors being modulated by the base object or texture color. Applying a pure white (255,255,255) material to the object first will resolve this problem.

3.7 Negative Scaling

If negative scaling is used to create a mirror image of geometry, the normals are reversed and the geometry will not work in RenderWare Graphics. A better approach is to use the Mirror or Xform modifier.

Occasionally, 3ds max gives dummy (helper) objects a negative scaling. As these are not geometry, you can ignore any warnings. However, since you cannot remove negative scaling from helper objects, don't parent other objects to them. You may be able to correct negative scaling using the Reset Transform tool under the Hierarchy panel.

RenderWare Graphics uses a right-handed coordinate system. This is different to the coordinate system used by 3ds max. Refer to [3dsmaxTutorials.pdf](#) for more information.

3.8 Collapse Edit Stacks

Whilst not critical, it is certainly desirable and good working practice to collapse the edit stacks once modeling and texturing is completed. This avoids large files that can be difficult to work with. Files can come down to a fraction of the size after collapsing the stacks.

Use the Collapse utility in 3ds max's Utilities panel to deal with large scenes quickly.



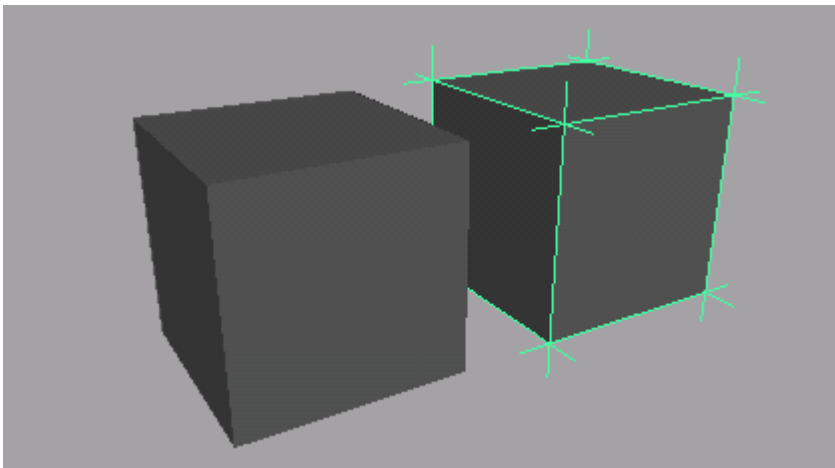
The skin modifier used must be at the top of the Edit Stack otherwise the RenderWare Graphics exporter will not recognize it as being a skinned object.

3.9 Normals and Smoothing Groups

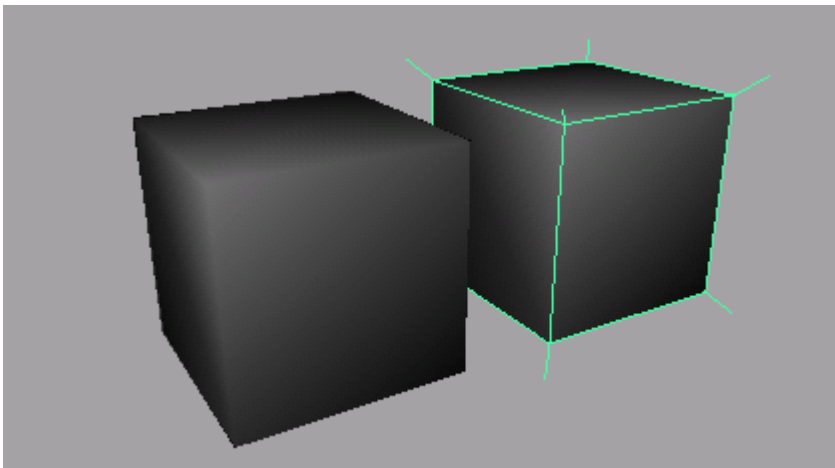
If you want to retain a low vertex count and good tri-stripping, you need to ensure that normals are shared between faces by applying smoothing groups to your objects. Note that the default for many 3ds max objects is *not* to smooth, which means you may be exporting many extra unnecessary vertices. The disadvantage of smoothing is that it changes the appearance of your models, as shown below.

RenderWare Graphics supports a single normal per vertex. In 3ds max, it is possible to have multiple normals per vertex. In these situations, the exporter has to duplicate vertices to maintain the different normals. So, to maintain the hard edged look, RenderWare Graphics will have to create three (or more) times as many vertices. In addition, it's not possible to tri-strip across the edges as the vertices are not common.

The screenshots below show a pair of default cubes before (above) and after (below) the normals have been adjusted. Note that the lower pair of cubes have only one normal for each vertex and that, as a result, the shading is smoothed around the corners.



Before: three normals per vertex – notice the sharp shaded edges



After: one normal per vertex – notice the smooth shaded edges

Smoothing Groups

Creating hard and soft edges in 3ds max is accomplished by assigning sets of faces to different Smoothing Groups. Vertex normals are averaged across smoothed sides but split where a hard edge occurs.

All faces must be assigned to smoothing groups, even if you want an object to appear faceted.



3ds max does not add extra normals to a vertex array when smoothing groups are cleared. For faces to be flat shaded, you need 4 normals per vertex and set the angle for *Auto-Smooth* to 0.0.



If lightmap shadows do not display as expected, errors may be caused by interpolating normals across the polygon. This is most common on flat surfaces where the vertex normals are not the same as the face normals.

3.10 Hidden Objects

Normally, when exporting geometry using the *Static World* template, any hidden geometry will be exported as usual. The reason behind this is to stop you constantly hiding and unhiding objects in large scenes. If you do want part of your world to be excluded from the export, select the object, open its Properties dialog and uncheck *Renderable*.

Hidden objects and dummies, such as Biped parts, will not be exported in the *Animated Hierarchy* format.



If an object in a 3ds max scene has no faces, the exporter may crash. To delete the objects with no face you can use the MAXScript:

```
for i in geometry do if (i.mesh.numfaces==0) then delete i
```

3.11 Level of Detail

RenderWare Graphics has the capability to switch models based on distance to the camera. All that is needed from the artist are separate models each exported individually with the required level of detail.

3.12 Instances

It's often useful to use instances in your scenes. When exporting *Animated Hierarchy* assets the RenderWare Exporter will attempt to preserve instancing, rather than exporting multiple copies of the geometry. Whilst 3ds max allows the changes to material assignments on instanced geometry, this is not supported by RenderWare Graphics. If you want instancing to be preserved when exporting to RenderWare Graphics you will need to ensure material assignments are not changed between instanced objects.

RenderWare Graphics does not support instancing in *Static World* files so instanced or referenced objects are always duplicated when exporting to this format.

3.13 Potentially Visible Sets (PVS)

RenderWare Graphics uses PVS to decide what can be seen by the user. It checks visibility from each part of the model. The parts that are *not* visible are culled reducing processing time. PVS culls all unseen or occluded geometry.

For example, a camera placed inside a modeled world may not be able to view all visible geometry. This geometry could be behind the camera or blocked by other geometry and is culled.

PVS data is stored in a database known as Potentially Visible Sets (PVS), calculated once per model.

Not all models benefit from PVS.

Example 1:

A landscape with few occluders will not benefit from PVS because every sector will be visible from every sector. The more occluders the better the PVS result.

Example 2:

The generation and efficiency of PVS is affected by cracks and T-junctions.

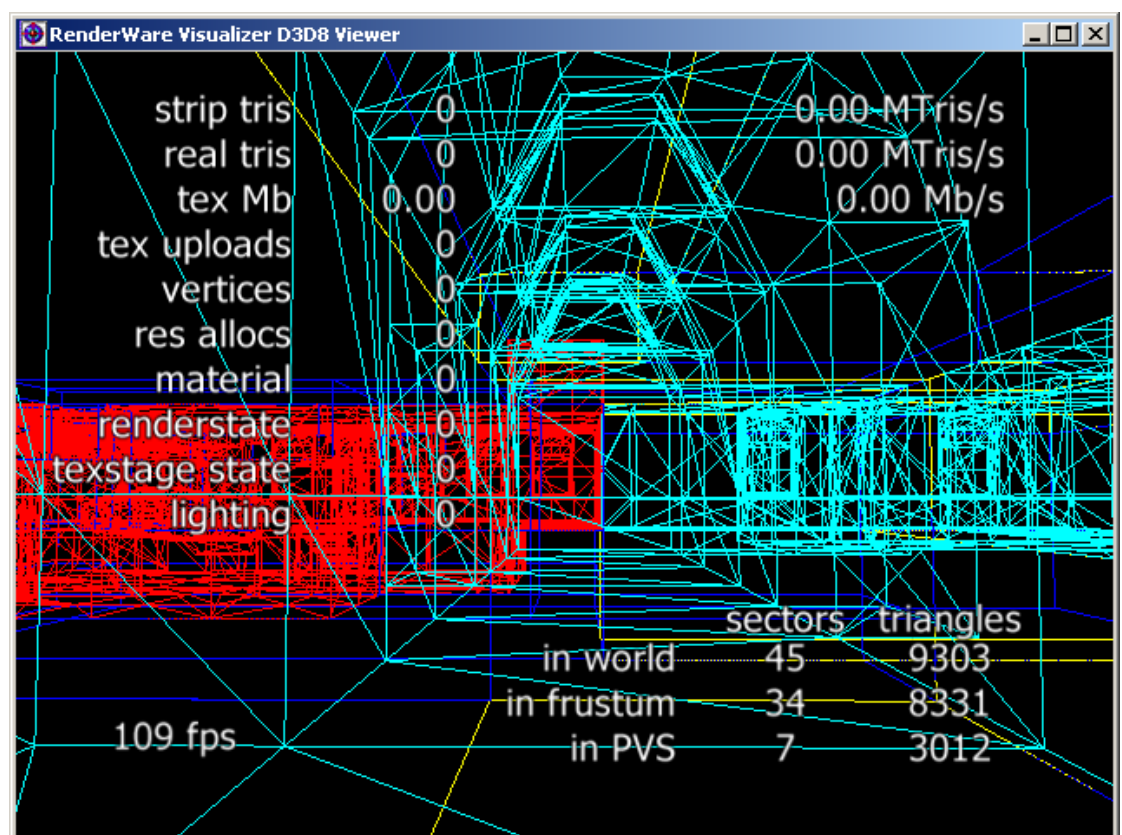


Visualizer

To view PVS in the Visualizer, open your model and:

- disable Triangles
- enable Wireframe
- enable World Sectors
- enable Use PVS

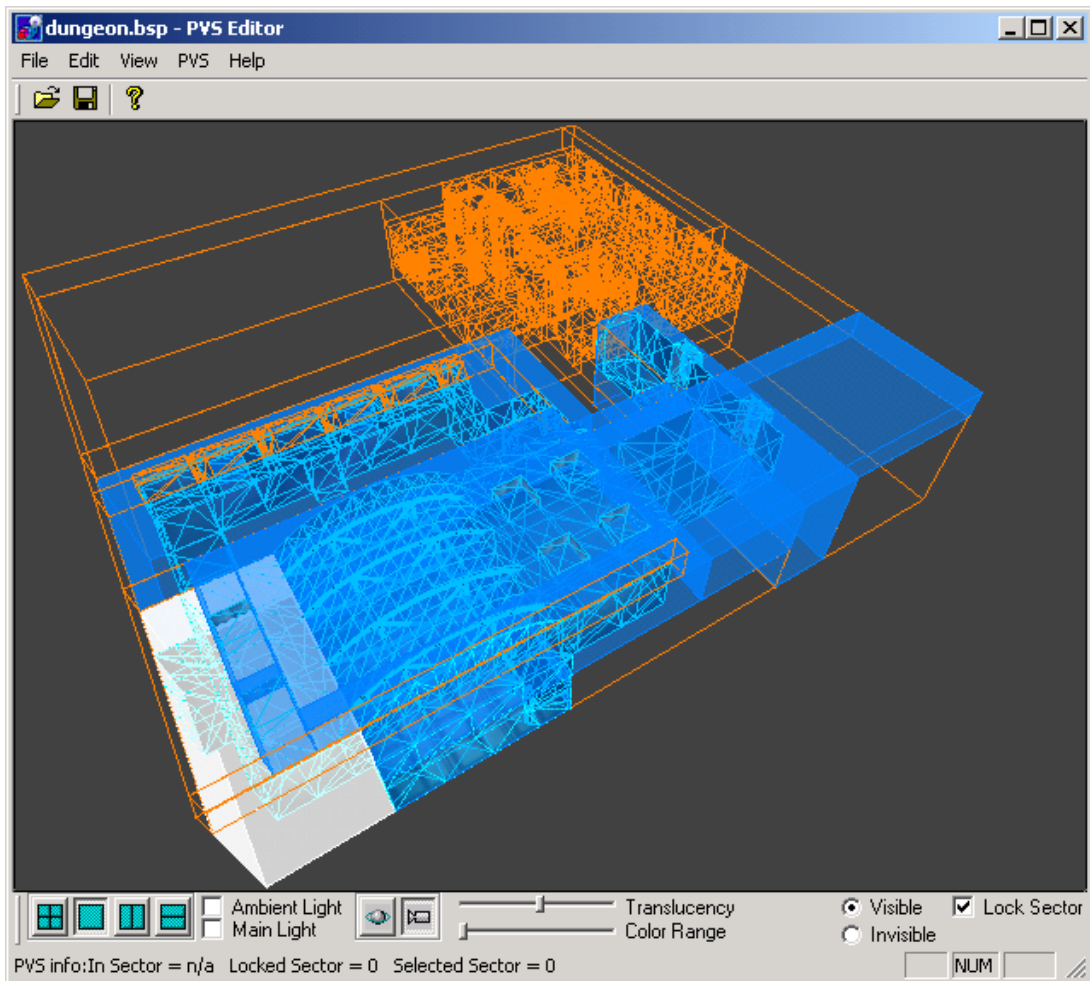
Your model should display a cyan wireframe for the sector the camera is in. The other colors represent non-visible sectors.



PVS Editor

RenderWare Graphics ships with a PVS Editor. The PVS editor is a tool that allows the PVS data in a world to be edited. The PVS Editor is located: `RW\Graphics\tools\pvsedit\PVSEdit_<platform>.exe`. The PVS Editor has a .chm which details keyboard shortcuts and can be accessed from within the PVS Editor. A separate pdf: `RW\Graphics\docs\tools\PVSEdit.pdf` explains the use of the PVS Editor in more detail.

The PVS Editor below, displays color coded sectors. If you are inside the selected white sector, then you can potentially see the blue sectors but you are unable to see the orange sectors. This speeds up performance as unseen sectors will be culled.



PVS Editor does not support .rws files.

3.14 Hints

While many of the considerations concerning the partitioning of a world are covered in the 3.15 Partitions section the 'hint' object that affects this stage is a Modeling concern.

What are hints

Hints are a special type of primitive that affect the sectorization of the scene on export but do not get exported as geometry to the final model.

Just a few hints placed strategically in the scene can significantly improve the sectorization, and they provide a way to affect the partitioning more controllably than before.

The hints are used in conjunction with the automatic partitioning of a scene.

Hint types

There are two types of hint: Shield hints and partition hints. In addition, they come in different shapes, all orthogonal: boxes and three plane types (XY, YZ, XZ).

Shield hints

Shield hints attempt to stop partitions cutting through them, and so they can be placed around small, complex geometry, such as a statue, to give the partitioning process a 'hint' as to where to avoid cutting.

Shield hints are color coded blue.



In many of the exporter partitioning schemes, a space culling algorithm is applied, and since this, by definition, does not cut through any geometry, it is permitted to cut through shield hints, if necessary!

Partition hints

Partition hints are an additional option that tells the partitioning process the locations where a partition would be appropriate. A bounding box partition hint might have its faces surrounded by partitions, while a plane partition hint would just have its single face coplanar with a partition.

Partition hints are color coded red.

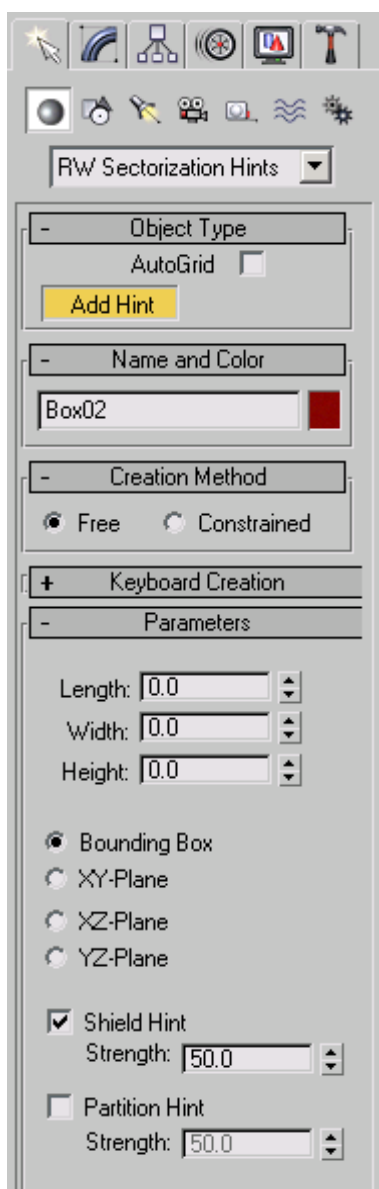
Combining hints

Since partitions hints and shield hints are not mutually exclusive, a hint can be assigned both as a shield hint and a partition hint! Combined hints are colored magenta.

Creating hints

A hint can be created by selecting
Command Panels → *Create* → *Geometry* → *RW Sectorization Hint* → *Add Hint*.

The hints attributes rollouts appear as below:



A hint object is drawn just like the box object. In addition, it has a number of options that make it act different, subtly affecting the partitioning process.

Initially, the hint is a bounding box. This can be switched to a plane type by selecting one of the radio buttons governing shape type. Note: A bounding box with a zero dimension can behave differently to an explicitly planar hint, so it is advisable to select the shape of the object before changing its dimensions.

Initially, the hint type is a shield. This can be deselected using the shield hint check box. The partition hint check box can be set likewise. Note, if both hint types are deselected, the hint will not affect the partitioning process at all.

Hint strengths

Each hint type has a strength associated with it. Even further control over the partition process will be gained using them strategically. As a general rule, it is best to leave the strengths at their default value. Only when a particular partition or shield hint should work differently should its value be altered.

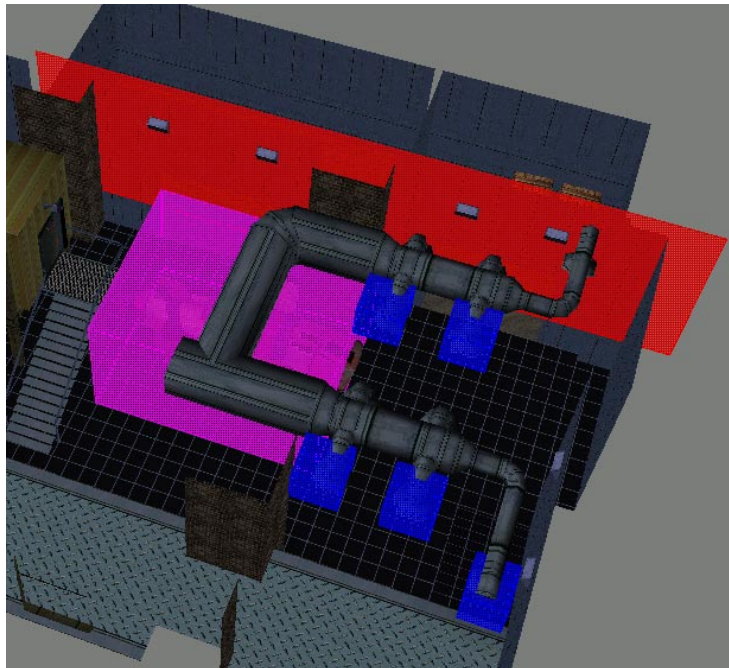
For additional feedback, the strength of the hint is associated with the brightness of it on the screen.

Here are some insights into the use of hints and their strengths:

- A partition hint with a high strength will be picked in favor of one with a lower strength. This means it will most likely be a parent of a lower strength one if that one is picked at all.
- Partition hints are affected by shield hints, but regardless of what they cut through they will be picked (to respect the artists intentions).
- If there are two partition hints and one of them cuts a shield hint, the other one will be picked in favor of the latter.
- A partition with a strength of x that cuts through two shield hints with powers of y and z will have a resulting strength of $x-(y+z)$

Hint Example

Here, we see a level that has had a set of hints applied to it.



Shield hints (blue) have been placed around each air duct in order to stop them being split; a partition hints (red) has been placed along the back wall so that the world is partitioned along it; and a combined hint (magenta) has been placed around a complex object so that it is partitioned around but not occupied by partitions. In the 3.15_Partitions section we'll see how this affects the partitioning of the world.

3.15 Partitions

The *Visualize Partitions* option in *Static World* asset properties shows the partitions/sectors that are generated as a result of the export process.

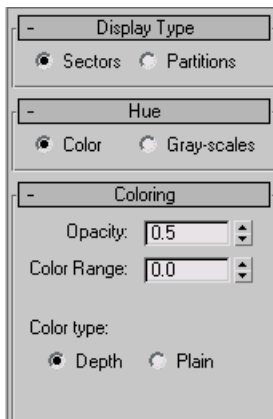
Whilst the partitioning of an exported world can be viewed in the viewer, it is sometimes useful, especially when using hints, to view them in the art package.

Note, the partitions are a read-only mechanism and are locked in place. No manipulation of them can be used as a feedback to the export process and they should only be used to study the final result of the export.

Initially, they are drawn as sectors, colored based on each sectors depth in the BSP tree (blue being root-most through the leaf-most red). The display options can be changed so that the partitions (which define the sectors) are displayed instead.

Once created, the partition attributes can be modified using the *Command Panels* → *Modify* option on the object.

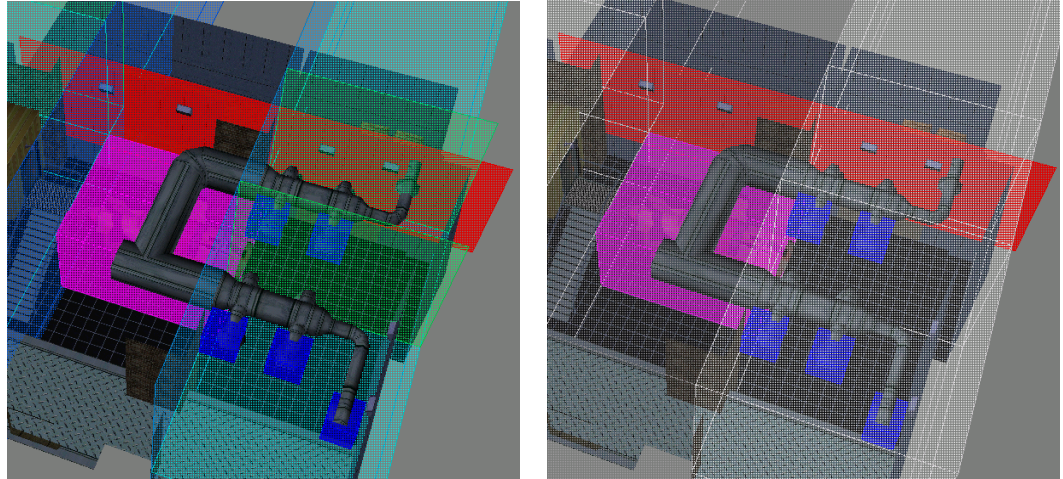
The partitions attributes roll outs appear as below:



For additional visual information there are two sliders that affect the opacity and the color-range of the visualization. Opacity can be set to any value between 0 and 100%. The color-range default is zero, this means that in a scene whose BSP tree has seven depths, each partition is assigned a color of the rainbow; sliding the color-range bar to the left shifts the range to the red-end of the spectrum, and to the right the blue-end. This is useful for identifying where in the BSP tree a partition lies, and provides further color enhancement.

If the visualization is overwhelming, they can be toned down: The coloring can be set to 'plain' which displays the partitions in a single color only. The hue type can be set to gray-scales which removes the color component.

These are two views (colored partitions and gray-scale sectors) of the example scene, from the 3.14_Hints section:



We can see how the partitions are visualized. Furthermore, we can see how the hints have been honored with the shield hints being uncut, and the partition hint being used to create some of the sectorization.

The exporters have more options about partitioning.

4. Material Reference

When producing artwork for RenderWare Graphics in 3ds max you have two options for creating materials. The first method is to use the built in 3ds max Standard Materials (Blinn, Phong etc). The exporter will convert these materials to RenderWare Graphics RpMaterials at export time. The second option is to use the exporter specific RwMaterial.

There are a number of advantages for using RwMaterial:

- `RwMaterial` restricts material options to those that RenderWare Graphics supports. This makes it clearer what settings the exported material will have.
- `RwMaterial` allows you to link to platform specific effect files. Effect files contain a low level description of the material state for a given hardware platform. Effect files are created using a standalone, platform specific editor tool (FXEditor for Xbox and GameCube). They are complex to design but present the full capabilities of the target platform without translation. To use the FX Editors you need to fully understand Xbox and GameCube hardware.

It's a good idea to promote large areas of a single material in a sector for good tri-stripping.

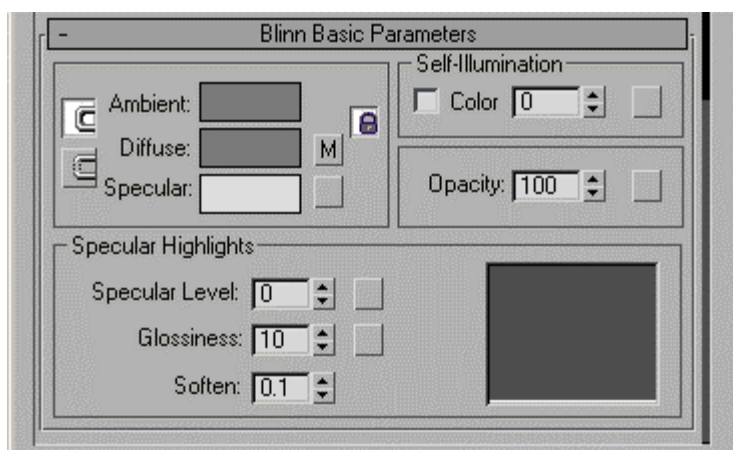


More information about the plugin `RpMatFX` and `RpWorld's` `RpMaterial` can be in the RenderWare Graphics API Reference and in the User Guide chapters *The Material Effects Plugin* and *World and Static Models*.

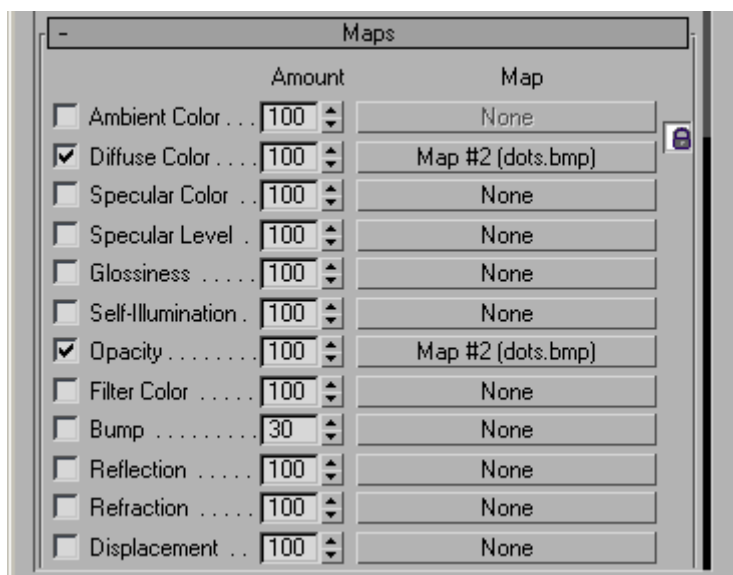
4.1 Standard Material

General Settings

The color of the exported RenderWare Graphics material is based on the node's wire-frame color if no material is assigned to the face. If the material is assigned it is based on the Diffuse color component, and if there is a Diffuse map it defaults to white in which case the name of the Bitmap filename (stripped of path and extension) will be used as the RenderWare Graphics texture map.



If a bitmap is applied to the Diffuse map, Opacity map will also be queried for a bitmap filename to use as a RenderWare Graphics opacity map. If no Diffuse map or no Opacity map is found the Opacity component value is used for the opacity value in RenderWare Graphics material.



Self-illuminating materials are supported. The Self-Illumination color/amount component will be added to the vertex color prelight (for more information see page 58.)

The material options set will be translated to RenderWare Graphics by the exporter.

MATERIAL OPTIONS	DESCRIPTION
Object Color	If no material is assigned the object's default color will be exported.
Material Diffuse	An object's diffuse RG color is exported when assigned to an object.
Diffuse Map	A bitmap assigned to the Diffuse Channel in the Material Editor will be exported and mapped onto the object matching the UV values set in 3ds max. Power of 2 sizes only should be used (for example: 32x32, 64x64, 128x128) in order to avoid RenderWare Graphics re-sampling the map sizes on loading.
Self-Illumination	If an object is required to be self-illuminated this value can be set in 3ds max and exported. For this value to be recognized <i>Vertex Prelights</i> must be set to TRUE in the Advanced Settings RenderWare Graphics Exporter.
Material Opacity	An object can be made transparent by setting the opacity value in the Material Editor. The transparency of the material is the alpha of the color of the material.
Map Opacity	Opacity of an object can be controlled by using a gray scale bitmap. This is applied in the normal way in the Opacity Channel of the Material Editor. White is opaque and black is transparent. Power of two sizes only should be used. Note: Materials with opacity maps must have a diffuse map as well to work.
Diffuse and Opacity Maps	Diffuse and opacity maps can be used together when the sizes of both must match.
Double sided Materials	Are supported by RenderWare Graphics. Export 2-sided automatically creates back-facing triangles where geometry has a 2-sided material defined.

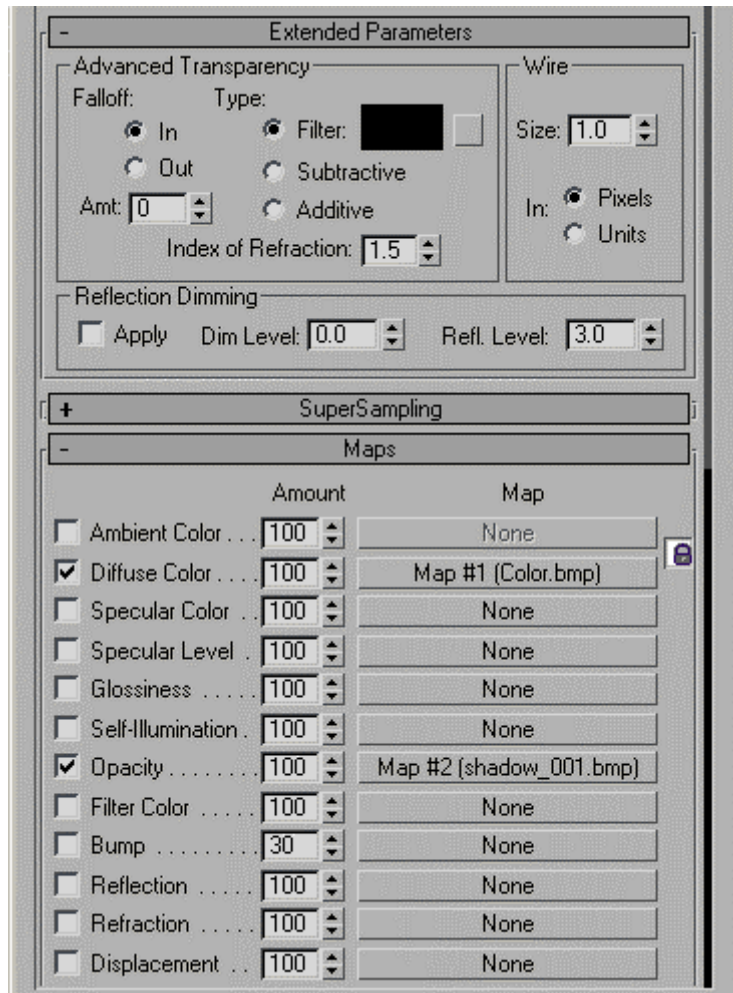
Transparency

Transparency is straightforward in RenderWare Graphics. Simply create a material as normal either by using a separate gray scale opacity image or by saving the texture maps with a built-in alpha channel. In the latter case you'll need to use the .png file format as .bmp doesn't easily support an alpha channel.

When a RwTexture is created from the diffuse and opacity maps, the maps are merged into a single RGBA texture (with alpha). The name of this new RwTexture is the name of the original diffuse texture. RenderWare uses texture names, so the merged texture is used for each instance of the diffuse texture. This can cause the textures in exported scenes to be different than those expected.

Seeing Built-in Alpha Channel

To use .png alpha in the Material Editor, load the same file into both the diffuse and opacity channels. Then in the *Advanced Transparency* section under *Extended Parameters* set *Type* to *Filter* and the color to black.



Exporting Transparent Objects

You should be careful when deciding how to export transparent objects from your scene. RenderWare Graphics does not guarantee that transparent objects are sorted before rendering and therefore transparent polygons may not always render as you expect. You should talk to the programmers on your project about how they want to handle transparency as it may affect how you need to model your scene. You may need to split transparent objects into separate *Animated Hierarchy* or *Static World* assets so they can be rendered after the opaque parts of the scene.

Bump and Environment Mapping

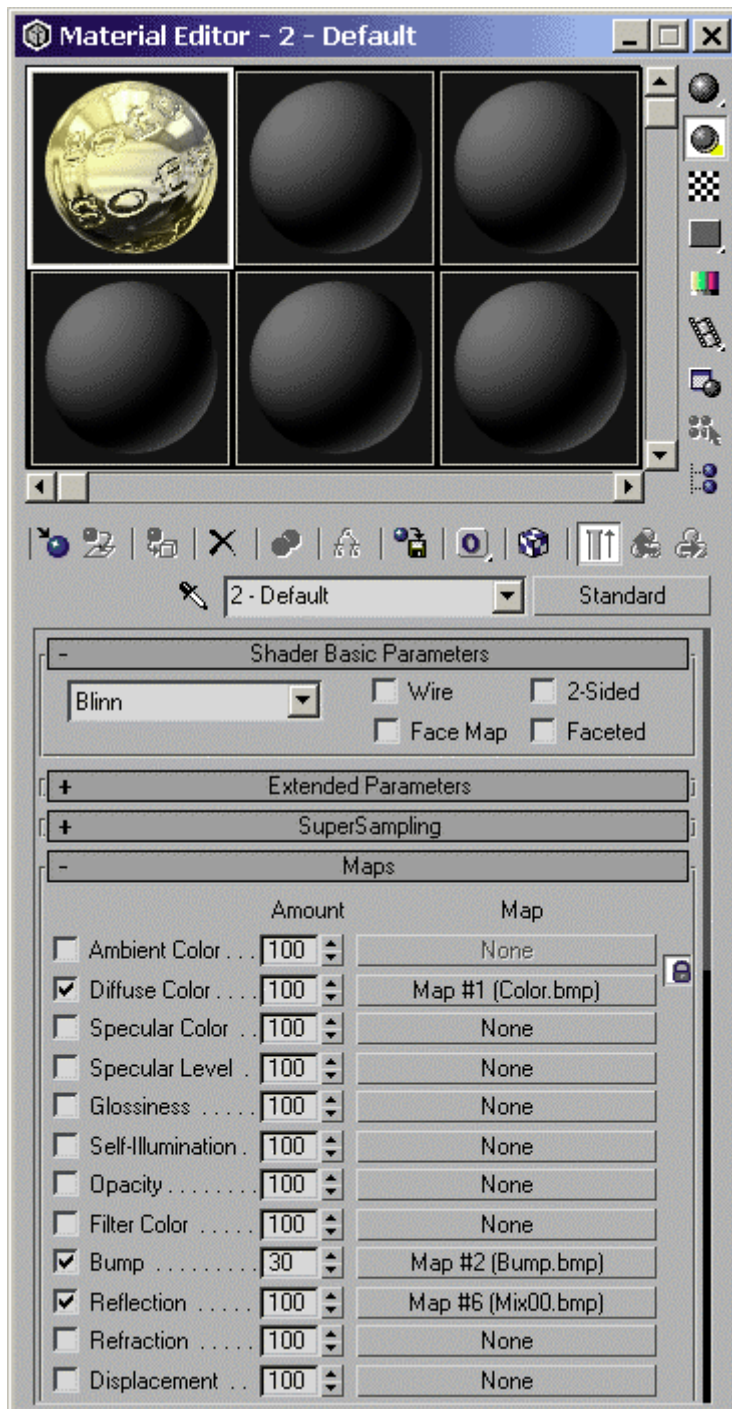
You can use environment maps, bump maps, or both together in RenderWare Graphics. Note that what you see in 3ds max will not be exactly what you get in your game, so export to your target platform often to check the results.

The only settings from the Material Editor that are used are the bitmap name of the bump or environment map and the Amount values.

Note that the Bump map must use the same mapping coordinates as the Diffuse map. In fact we only export the mapping coordinates for the Diffuse map, the Bump mapping coordinates are assumed to be the same.

Also note that you may not have a Diffuse map with an alpha channel at the same time as Bump mapping. This is not provided because the performance implications are severe.

Environment mapping in RenderWare Graphics corresponds to Reflection map in the material editor. The process is additive in RenderWare Graphics so the Amount value of 0 is 100% Diffuse map and 0% Reflection map, a value of 100 is 100% Diffuse and 100% Reflection map, i.e. the colors will be brighter.



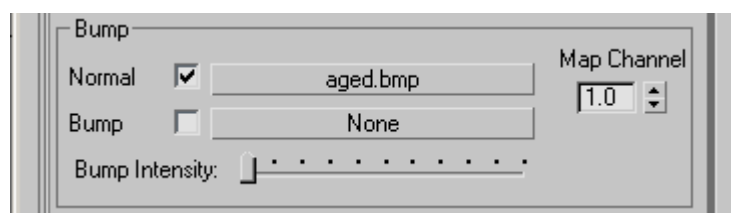
On PC, D3D8 and OpenGL targets support environment-bump mappings and are achieved in either multi-pass or multi-texture in some instances of D3D8 (if the video card driver supports multi-texturing).

On PlayStation 2 it's too expensive for bump mapping to effect the surface normals used for environment mapping. Bump mapping combined with environment mapping does not use environment mapping but uses emboss mapping instead. However, the displayed results look the same on all platforms using RpMatFX.

On Xbox and GameCube the RpMatFX multi-texturing functions provide support for environment-mapped bump mapping.

Normal Map

You can specify a normal map in the Metal Bump Viewport Shader. Note that this is only a 3ds max 5 feature.



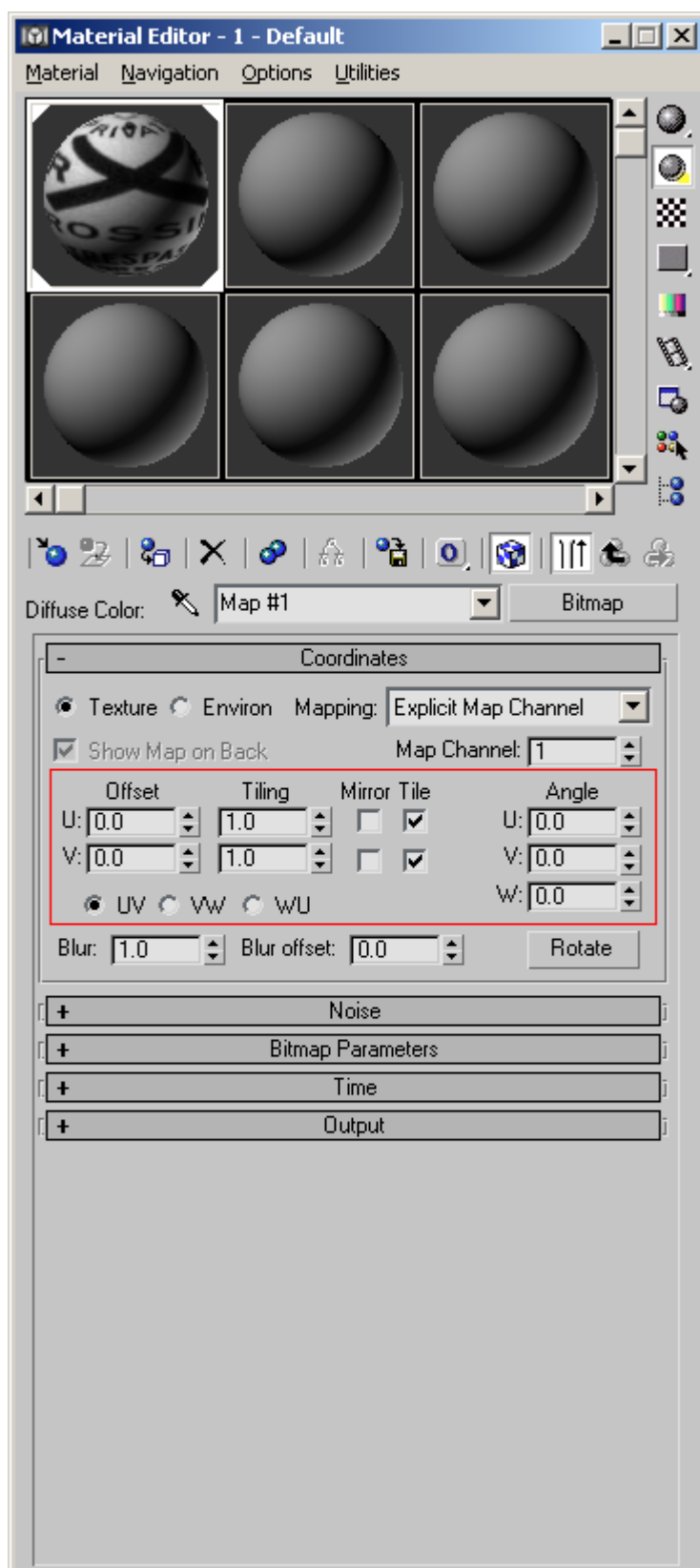
For more information on Normal Maps, refer to Normal Maps (Xbox and D3D9 specific).

Texture UV Animation

You can export texture UV animation to RenderWare Graphics. To do so, simply select the diffuse texture that you want to animate in the material editor, and then add your key frames to anyone of the Offset, Tiling or Angle fields in the “Coordinates” rollout.

When animating the angle field, only the W angle can be animated, since RenderWare Graphics does not support 3D textures.

With the standard material you can only animate one texture, which is the diffuse channel. You cannot use UV animation together with bump mapping or environment mapping, so make sure that these maps are disabled when applying UV animation. (You can still use the opacity map). If you do setup both bump or environment mapping and UV animation then the bump or env effect will take precedence and UV animation will not render.

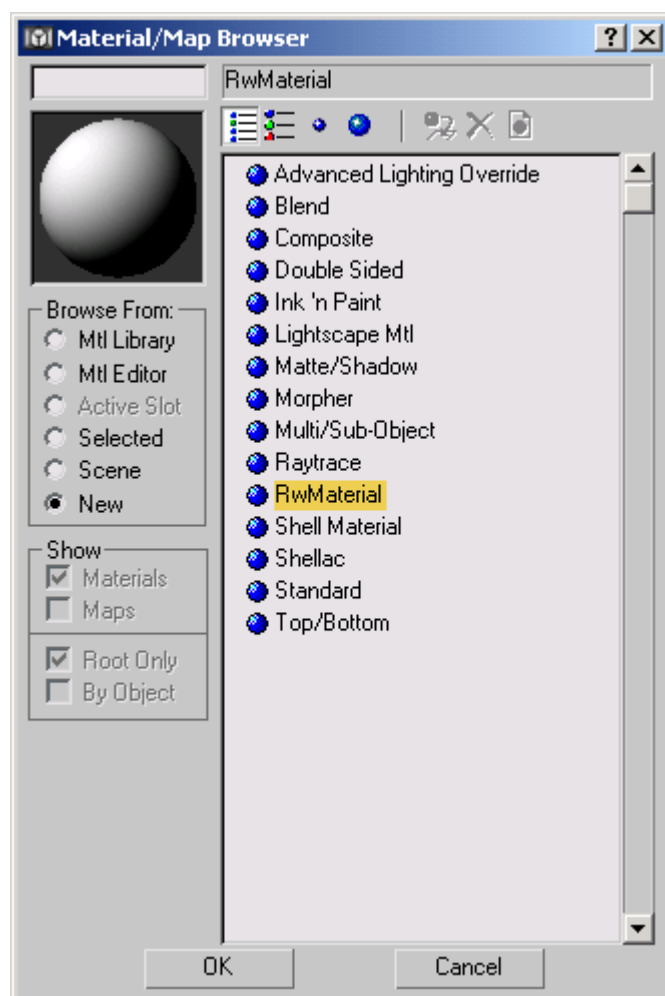


4.2 RwMaterial

It is recommended you use `RwMaterial`, the RenderWare Graphics material. It provides direct settings for the RenderWare Graphics `RpMaterial` object and no translation of these settings is performed when exporting.

To use `RwMaterial`. In the Material Editor, click the Standard button.

Select `RwMaterial` from the Material/Map Browser dialog. Click OK.



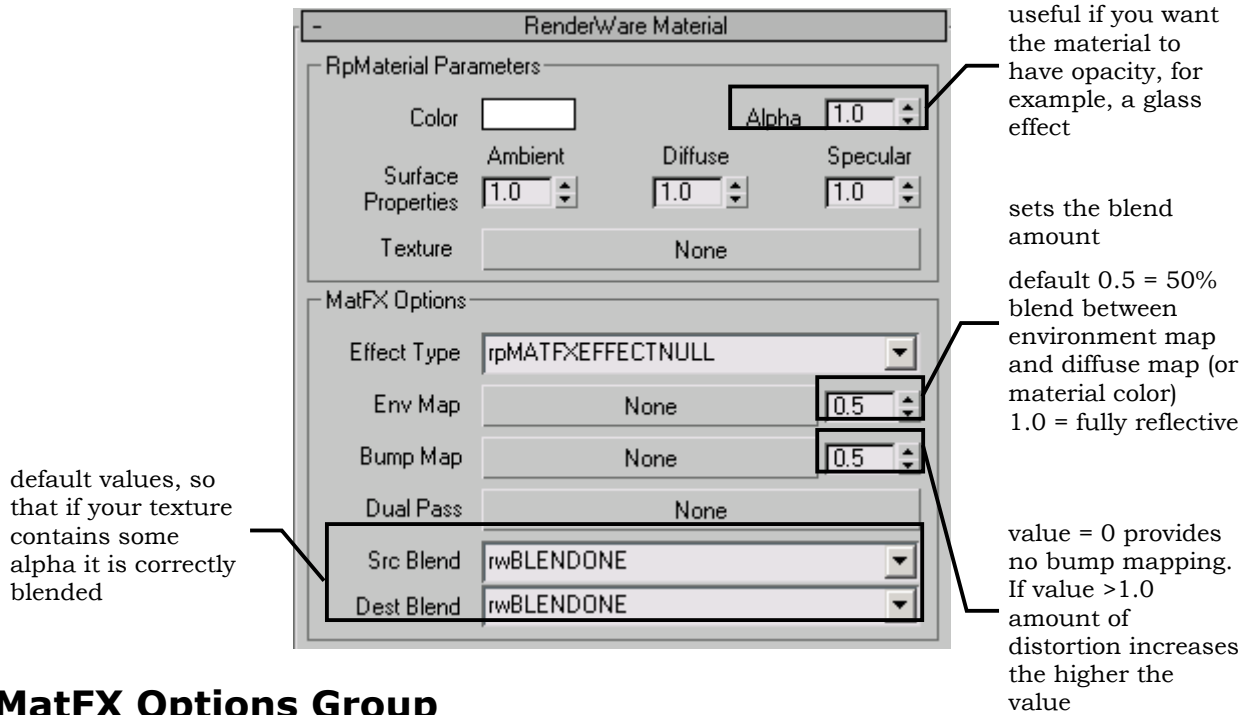
RpMaterial Parameters Group

The `RpMaterial` Parameters group provides settings for the `RpMaterial` options in RenderWare Graphics. Clicking on the Color canvas brings up a Color Selector dialog where you can set the material color. An alpha value is added to the alpha component of the RenderWare Graphics material color.

The Surface Properties values are used to control how the material reacts to light in the scene. At present the Specular value is not support on most platforms.

The Texture map control is used to specify a file texture that may be exported as the base texture in RenderWare Graphics. The base texture color is always modulated by the material color in RenderWare Graphics. The base texture should use the first UV set.

RenderWare Material options:



MatFX Options Group

These settings control the material effects (MatFX) settings for the RenderWare Graphics material. MatFX is used to setup special effects such as bump and environment mapping, and dual pass rendering. Depending on the MatFX effect selected, the various options beneath will be used. When using the dual pass effect, the blend mode used is set explicitly using the Src Blend and Dest Blend controls. The options in this group can also be seen in the RenderWare Graphics MatFX example in the SDK.

Effect Types

Effect Types relates to the function `RpMatFXMaterialSetEffects()` in the RenderWare Graphics API.

- `rpMATFXEFFECTNULL` has no material effect.
- `rpMATFXEFFECTBUMPMAP` enables bump mapping on the chosen material.
- `rpMATFXEFFECTENVMAP` enables environment mapping on the chosen material.
- `rpMATFXEFFECTBUMPENVMAP` enables both bump and environment mapping on the chosen material.

- `rpMATFXEFFECTDUAL` enables dual pass texturing. This option enables blending modes.
- `rpMATFXEFFECTUVANIM` single pass texture coordinate animation. You can use this effect without setting the UV animation on the texture.
- `rpMATFXEFFECTDUALUVANIM` dual pass texture coordinate animation. You can use this effect without setting the UV animation on the texture.

Environment and Bump Mapping

Environment mapping is used in RenderWare Graphics to add reflectivity to objects. It can be used to simulate effects such as chrome plating, glass or water. We recommended that you use the `RwMaterial` for environment mapping as it provides the direct settings provided by RenderWare Graphics.

Bump mapping can be used to simulate rough surfaces such as stone or tree bark.



PlayStation 2

In RenderWare Graphics for PlayStation 2, `RpMatFXMaterialSetEnvMapFrameBufferAlpha()` is used to modulate the environment mapping by frame buffer alpha (for those video modes that support it). For the bump-environment mapping effect, the frame buffer alpha contains the bump alpha value and the result gives the appearance of the reflection being affected by the bumps.

Dual Pass

This effect works by combining the material's own texture with a second texture according to specified combination flags.

The setup function is `RpMatFXMaterialSetupDualTexture()` and sets the following properties:

- The second texture (the first is defined by the `RpMaterial` object)
- The blend function for the source data
- The blend function for the target data

The dual pass texture should use the second UV set.

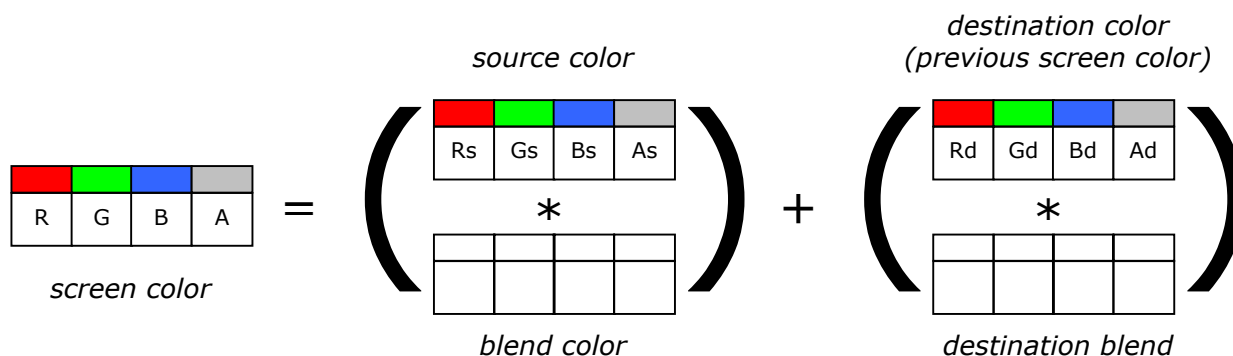
Blend Modes

Texture blending is the technique of combining the colors of a texture with the colors of the surface to which the texture is applied.

`Src Blend` and `Dest Blend` are used in the function `RpMatFXMaterialSetDualBlendModes()` in the RenderWare Graphics API. To use blend modes within `RwMaterial` the `MatFX` effect type should be set to `rpMATFXEFFECTDUAL`.

The diagram below shows how the screen color is created.

For the red part of the pixel, the red source color is multiplied by the given red source blend color. The result is added to the red destination color multiplied by the red destination blend color. The final result of this is the red of the pixel on screen. This occurs for each color.



R = red, G = green, B = blue, A = alpha, s = source, d = destination

Substitute the source blend and destination blend with values of the `RwBlendFunctions` used. A description of how each color component is blended in texture blended is listed below:

- `rwBLENDZERO` - RGBA channels set to zero
- `rwBLENDONE` - RGBA channels are set to 1
- `rwBLENDSRCOLOR` - source RGBA only
- `rwBLENDINVSRCOLOR` - inverse of source RGBA only
- `rwBLENDSRCALPHA` - source alpha only on all channels
- `rwBLENDINVSRCALPHA` - inverse source alpha only on all channels
- `rwBLENDDESTALPHA` - destination alpha only on all channels
- `rwBLENDINVDESTALPHA` - inverse destination alpha only on all channels
- `rwBLENDDESTCOLOR` - destination RGBA values only
- `rwBLENDINVDESTCOLOR` - inverse destination RGBA only
- `rwBLENDSRCALPHASAT` - source alpha (saturated)

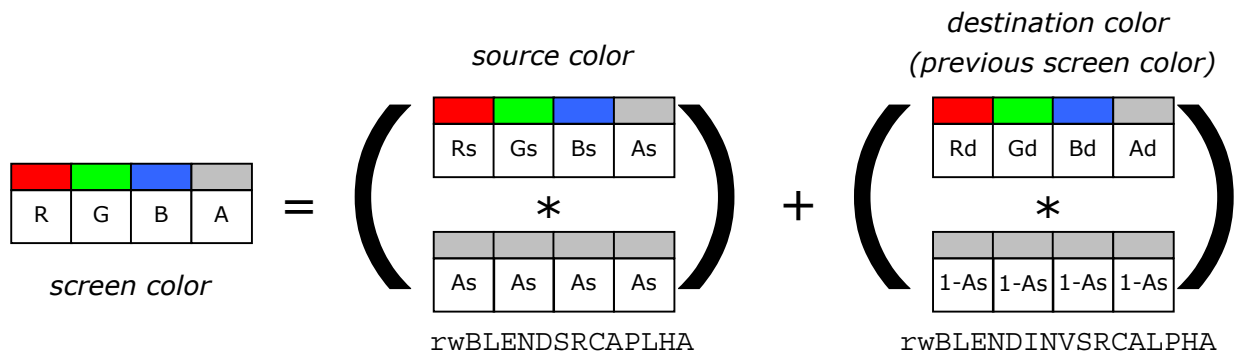
These values are only valid for dual pass using MatFX.

With dual-pass texturing, you can consider the first texture to be the destination and the second texture the source (you are blending the second onto the first).

The resulting pixel color is given by the formula:

```
( srcBlend * srcColor + destBlend * destColor )
```

The example below, shows the most commonly used blend modes.



R = red, G = green, B = blue, A = alpha, s = source, d = destination

The above example will be used when rendering alpha pixels, for example, the glass of a glass table. You would also use this blend mode to overlay a second pass texture, such as a logo onto a base texture.

Not all blend modes are supported on all platforms. The blend modes supported on each platform are listed in the platform specific section of the API Reference. Also note that programmers can also override the blend modes set using `RwMaterial`.

For further information about blend modes in RenderWare Graphics refer to the blending and matfx examples supplied with the SDK.

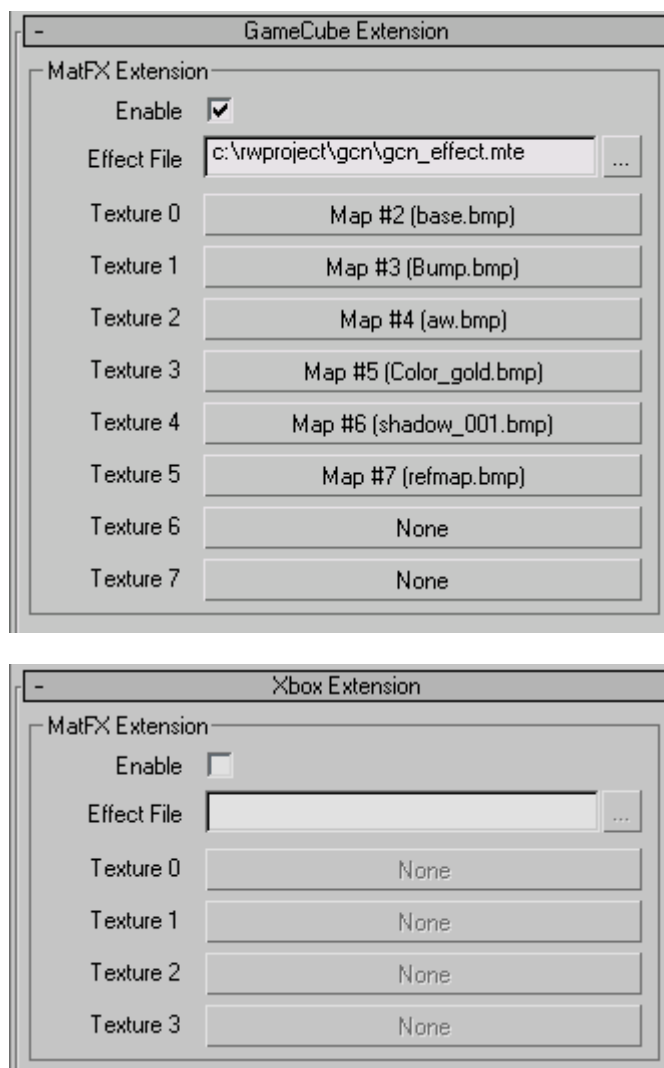
`RwBlendFunction` values:

```
rwBLENDZERO (0, 0, 0, 0)
rwBLENDONE (1, 1, 1, 1)
rwBLENDSRCCOLOR (Rs, Gs, Bs, As)
rwBLENDINVSRCCOLOR (1-Rs, 1-Gs, 1-Bs, 1-As)
rwBLENDSRCALPHA (As, As, As, As)
rwBLENDINVSRCALPHA (1-As, 1-As, 1-As, 1-As)
rwBLENDDESTALPHA (Ad, Ad, Ad, Ad)
rwBLENDINVESTALPHA (1-Ad, 1-Ad, 1-Ad, 1-Ad)
rwBLENDDESTCOLOR (Rd, Gd, Bd, Ad)
rwBLENDINVESTCOLOR (1-Rd, 1-Gd, 1-Bd, 1-Ad)
rwBLENDSRCALPHASAT (f, f, f, 1) f=min (As, 1-Ad)
```

Platform Specific Extensions

The `RwMaterial` provides platform specific settings for a number of targets. At present Xbox and GameCube are supported.

Each platform has an Enable control. If this is disabled (the default) there will be no specific settings exported for this platform. There are a number of maps that may be linked to a Bitmap file. That number varies depending on the capabilities of the target platform.



Effect files contain a low level definition of the material hardware state for the platform. Effect files are created and edited using a standalone, platform specific editor tool FXEditor (for Xbox and GameCube) which is installed as part of the main RenderWare Graphics SDK. They are complex to design but present the full capabilities of the target platform without translation.

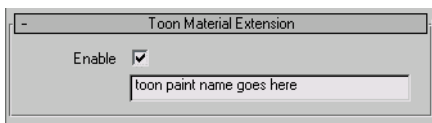
The exporter does not actually load the effect file you specify, it simply stores the effect file name (without a path but with an extension). You do not need to use the *Browse* button to select a file, you can simply enter an effect name in the text field. You will always need to do this if the effect you wish to use is part of an effect file dictionary. Effect file names are limited to 31 characters. If the name you use is longer it will be truncated on export and you will be given a warning message. When your exported assets are loaded into RenderWare Graphics the effect file will be found and loaded. You can view effect files rather like textures. They are edited externally to 3ds max, referenced by name from the 3ds max scene and loaded by RenderWare Graphics at runtime.

Toon

Toon is a plugin in RenderWare Graphics called RpToon. It's a chargeable optional extra in the FX Pack. Toon provides rendering of outlines and "cel shaded" texture based lighting for RenderWare Graphics objects to achieve cartoon-like 3D graphics. An art path is provided to allow the artist wherever possible control over the properties of the lines and shading, offering functionality similar to that of non-real time toon rendering plugins such as Illustrate! for 3ds max.

It's recommended that you read the Toon documentation to understand toon in more detail.

Toon Paint names can be specified per material to override an object's default paint if you wish.

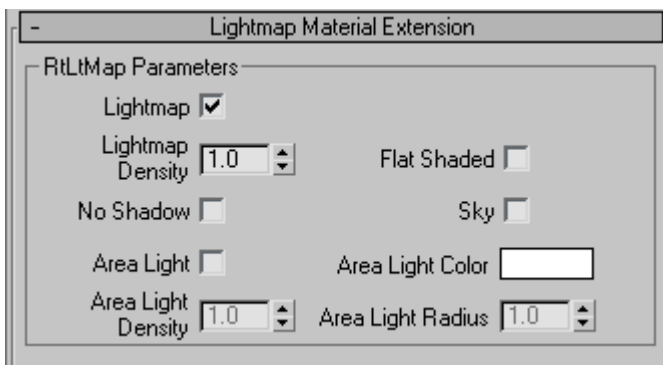


Checking the Enable box allows the paint name entered in the box below to override the default paint for the object for just this material.

LightMaps

The lightmap options in the exporters provide a means to render geometry in RenderWare Graphics using detailed, full color, static lighting information from lightmap textures.

The lightmaps plugin has limited support for alpha. It works with a one bit alpha, as long as there are totally transparent regions. Sometimes for fine pieces of geometry lightmapping is not the ideal solution and you may find it better to use vertex lighting on detailed objects.



Sets the lightmap options used in RenderWare Graphics. These settings are used by RenderWare Graphics to generate the lightmaps.

The lightmap options for a material are:

Lightmap - enables lightmapping for this material.

Lightmap Density - density of lightmaps for polygons of this material.

Flat shaded - enables flat shaded polygons of this material when lightmapping.

No Shadow - polygons of this material do not cast shadows.

Sky - polygons of this material have directional lights shining through them.

Area Light - polygons of this material will emit light.

Area Light Color - the color of the area light emitted by polygons of this material.

Area Light Density - polygons of this material are covered by the density of temporary RpLights.

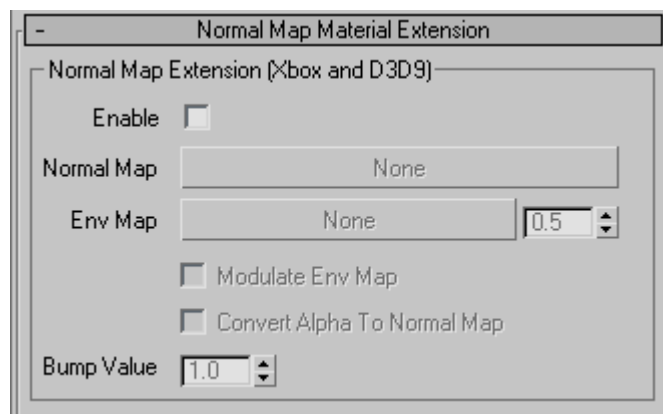
Area Light Radius - the radius of the area light emitted by polygons of this material.

Lightmaps are covered in more detail in the LightMaps section.

Normal Maps

This plugin provides facilities for using normal maps. Normal maps are textures that store a normal per pixel encoded as a color, converting the components of the normals from the range $[-1..1]$ to $[0..255]$. These packed normals are later unpacked in the pixel shader and are used to calculate the light contribution per pixel.

You can use one of the tools provided by 3ds max / gray scale limit convert alpha to normal map.



Enable: Enables normal mapping.

Normal Map: Loads a normal map file. Note that normal map can also be specified in the *Viewport Manager* → *Metal Bump Shader*.

Env Map: Environment mapping performed on a pixel. The value following, sets the strength of the environment map.

Modulate Env Map: This option controls whether the environment map should be modulated by the color of the base texture.

Convert Alpha to Normal Map: Convert standard image to normal map. To use add a standard bump map texture for Normal Texture and select this option. The value that follows is a scale factor for converting to normal maps and calculates the normal pixel.

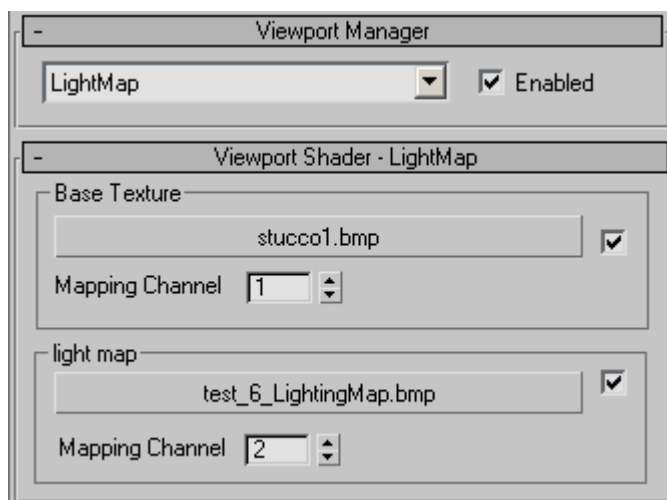
Normal maps are covered in more detail in the Normal Maps (Xbox and [D3D9 specific](#)) section.

Viewport Manager

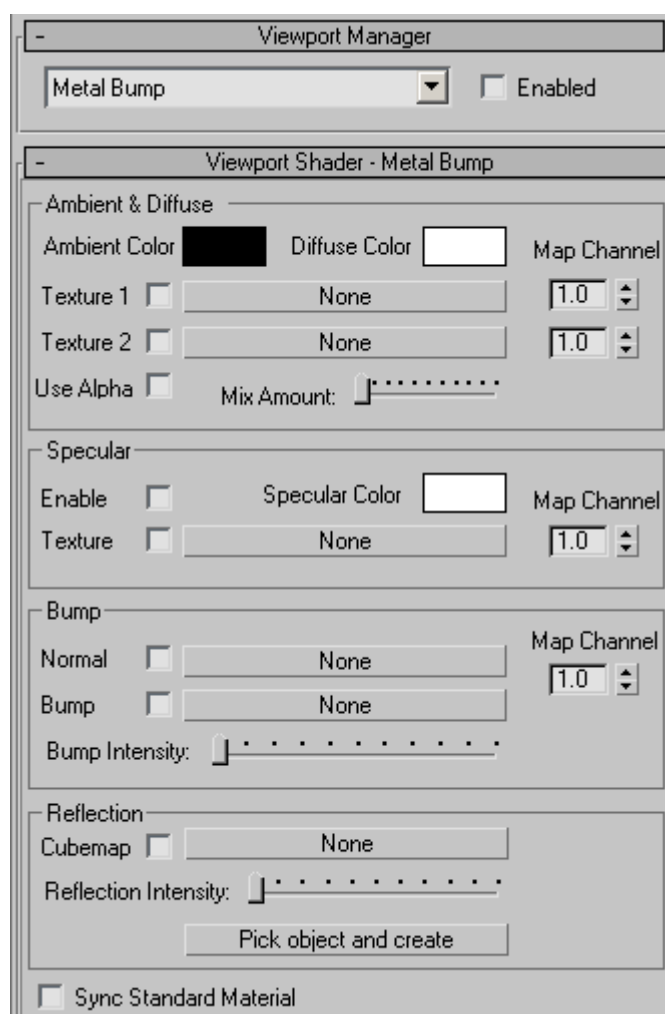
In 3ds max 5 within the Material Editor, using the DirectX driver, the Viewport Manager is included in the RwMaterial and Standard Materials.

Light Map

The Viewport Manager Lightmap settings can define both a lightmap and a diffuse channel. When set the lightmapped materials will be displayed correctly within the viewport. This lightmap and the diffuse textures will only be exported if the “Process Lightmaps” template option is set to “Export Native Lightmaps”. This option cause the conversion of the lightmap within the RenderWare Graphics exporters to a new lightmap which will be more optimal within RenderWare.



Metal Bump



RenderWare Graphics Xbox Exports

Normal maps can be included in RenderWare Graphics exports can be loaded using the *Normal* setting in the *Bump* section of the above dialog. Normal maps can be loaded in two places, in the above dialog and in the Xbox Specific section of *RwMaterial*.

If normal maps are loaded in both places the Xbox Specific normal map overrides the Metal Bump Viewport Shader normal map.



Hardware Preview

The hardware preview of *RwMaterial* in 3ds max varies depending on the capabilities of your graphic card. On the majority of cards you will be able to see multiplication (modulation) of up to first two or four texture maps. You can always see each map independently using the Show Map in Viewport tool button when editing a specific map.

Transparency is also supported and it is determined by an Alpha value in *RwMaterial*.

4.3 Common Material Topics

Texture Maps

Only bitmaps texture maps are exported, all procedural texture maps will be ignored.

When creating image maps it is generally recommended to:

- Keep file names to eight characters or less in the 8.3 format, because some consoles do not support long filenames;
- Use lower case;
- Do not use spaces;
- Save as .bmp or .png files. .png is recommended.
- Masks should be the same size as the base texture.

It's more efficient if you can use the same texture across large parts of your model. Putting different textures on adjacent polygons will force RenderWare Graphics to create lots of extra vertices that will slow down your game.

Texture Size

If textures are loaded into RenderWare Graphics from image files (such as bmps or pngs) rather than a texture dictionary, the image data needs to be converted to a format that RenderWare Graphics understands. This may involve resampling the image to a different width, height and/or depth which can slow load times. Some graphics hardware requires textures to have a width and a height that is a 'power of two' (i.e. 1, 2, 4, 8, etc). If the source images are not a power of two in either dimension, RenderWare Graphics may resample them. It is perhaps more preferable to resample the images in a paint package since RenderWare Graphics executes a very simple resampling algorithm.

If a textures is of a size that may cause problems on the platform being exported to the exporter will issue a warning in the RenderWare Output Window.

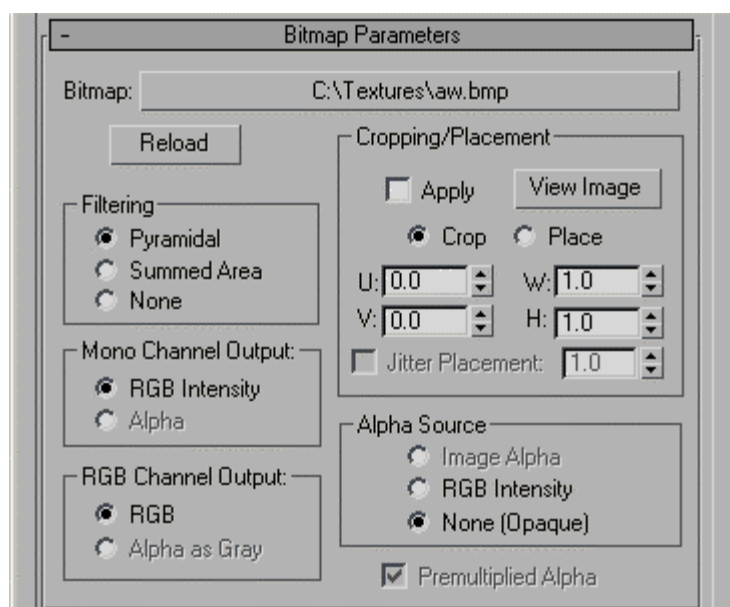
Texture Repeats

On PlayStation 2 there are strict limits to the texture UV range that may be used. In practice this means that for artifact-free rendering the size of your maps and the number of repeats must be restricted so that the product of the map size and the UV value must have a magnitude less than 2048. For example, a 256x256 map should not have UV values outside the range plus or minus eight. Rendering outside this range will introduce texturing errors.

The exporters have controls to enable the UV values output to be limited to specified ranges. When UVs are detected outside the specified range they will be shifted towards zero. The downside to this is that there will be discontinuities in previously smooth UV ramps. For example, rather than running 0-1-2-3-4-5-6-7 across polygons the UVs will now run 0-1-2-3-4 0-1-2. This will not affect the rendering but will cause vertices to be duplicated at the discontinuities and result in poorer tri-stripping. If a single polygon is detected with a UV range outside the limit the exporter will not be able to fix the problem (in other words the exporter does not attempt to tessellate geometry to restrict UV ranges).

Texture Filter Modes

Because the 3ds max filter modes do not map exactly to RenderWare Graphics ones not all RenderWare Graphics settings are available. There are three filter modes in 3ds max.



The table below illustrates how the 3ds max Filter Types map to the RenderWare Graphics filter mode.

FILTER TYPE	RENDERWARE GRAPHICS FILTER MODE
Pyramidal	rwFILTERLINEARMIPLINEAR - trilinear interpolation
Summed Area	rwFILTERLINEAR - bilinear interpolation
None	rwFILTERNEAREST- point sampled

The 3ds max exporter does not use the filtering types in the opacity slot, only the diffuse slot. Regardless of whether alpha is a separate image in 3ds max or combined in the diffuse, in RenderWare Graphics it's always combined into a single texture object. Since it is the texture object that has a filtering mode you are unable to have different filtering in the color part and alpha part of a texture in RenderWare Graphics. Therefore, we have to ignore the alpha filtering modes in 3ds max as RenderWare Graphics can not use separate alpha filtering modes.



The RenderWare Graphics Filter Modes are documented in the `RwTextureSetFilterMode()` function in the API Reference.

UV Mapping Coordinates

The UV mapping coordinates generated by the mapping tools in 3ds max should export with no problems, however there are some things you should know:

Mapping in curved space for patches are not supported (for more information see page 18.). At the Patch sub-object level there is a parameter in the right-click quad menu (Tools 1 quadrant) called Linear Mapping. You should turn it on to get the same results as in RenderWare Graphics.

Try to avoid using Box mapping. This generates large amounts of map vertices which all be exported to RenderWare Graphics.

RenderWare Graphics supports up to eight UV sets or mapping coordinates as they are referred to in 3ds max. It is very important to know how and which mapping coordinates are exported.

Standard Material

Using the Standard Material only one set of mapping coordinates can be exported to RenderWare Graphics. To obtain more sets of mapping coordinates you have to use `RwMaterial`.

Only one set of mapping coordinates used by the Diffuse map will be exported. Bump mapping coordinates are assumed to be the same as for the Diffuse map. The environment coordinates are generated at render time.

If you apply a UVW modifier to an object with built in mapping coordinates, the applied coordinates take precedence if map channel 1 in the UVW map modifier is used.

RwMaterial

This is where you can set and later export up to eight sets of mapping coordinates. There are some rules you'll have to follow when working with mapping coordinates:

- We support up to the first eight sets of mapping coordinates. This means all used UVW Mapping modifiers must have their Map Channel value between 1 and 8 where no gaps are allowed. If you use mapping coordinates number 1 and 3, and don't use mapping coordinates number 2, we will export three sets of mapping coordinates and the second one will be filled with zero values.

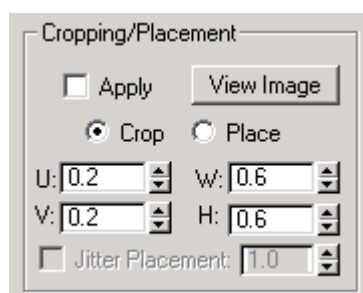
- The first Texture map which is exported as the base texture will always use the first set of mapping coordinates on the mesh. The Dual Pass map will default to the first set of mapping coordinates, if not set to use the second set or there are more than one set of mapping coordinates on the mesh, used by the maps in the platform specific extensions.
- If you use platform specific extensions, each map can use arbitrary set of mapping coordinates as long as the first rule is obeyed.
- The UV sets in RenderWare Graphics are 0-based, unlike in 3ds max where they are 1-based. Hence, the UV set with index 1 will correspond UV set 0 in RenderWare Graphics.

Using Cropped Textures

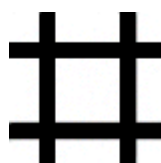
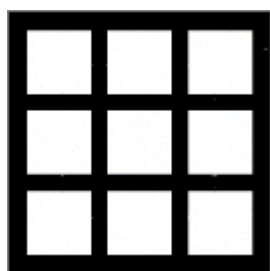
It is possible to use 3ds max's texture-cropping tool. But there are a few issues to be aware of, as follows.

Cropping Coordinates

RenderWare Graphics takes the coordinates directly from the UV and WH values regardless of whether the Apply box is checked or not. This is different from the 3ds max behavior. These values have to be set to zero if you don't want your texture cropped.



The settings above create this...

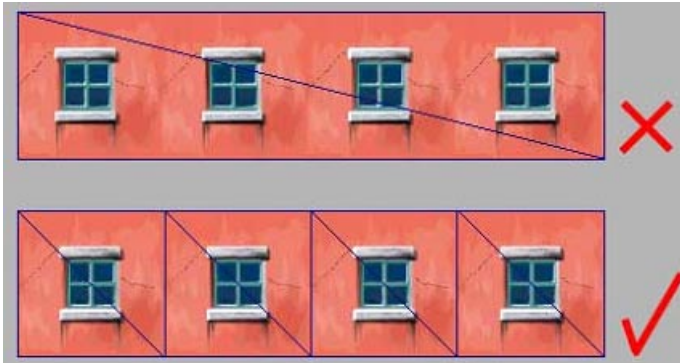


...in 3ds max, but this... ...in RenderWare Graphics

So if a material has had cropping applied but no longer needs it, set the UV and WH values back to zero; unchecking *Apply* will not work.

Cropping Across Polygons

In RenderWare Graphics you cannot tile a cropped texture across a single polygon as in the illustration below. It will not work, as RenderWare Graphics has no means of knowing where the crop should apply.



Don't tile a *cropped* texture across single polygons. It is illegal in RenderWare Graphics.

Material Color vs. Vertex Color

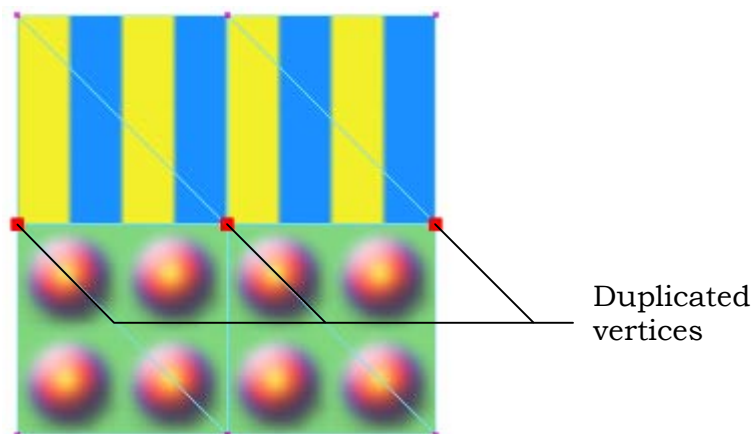
3ds max materials (colors or texture maps) are exported as RenderWare Graphics materials with self-illumination values baked in the prelight colors. Prelights are exported as vertex colors with support for the 3ds max features alpha and illumination values. These vertex colors are data applied at each vertex of the geometry and then interpolated over the face between the vertices. If you have very few vertices then you won't get very much color variation and poor quality lighting. If a geometry has both material and prelights then the final color is calculated by multiplying the vertex color by the material color at runtime.

Double-sided Materials

Double sided materials (also referred to as 2-sided materials) can be simulated by automatically creating backface geometry on export. This will duplicate the number of vertices and triangles in the geometry.

Extra Vertices from Multiple Materials

Every time the material changes across a model, RenderWare Graphics has to duplicate the vertices along the boundary since a second set of UV coordinates are needed. This can lead to a large increase in the number of vertices if adjacent materials change often. If you notice that there seem to be many more vertices in RenderWare Graphics compared with your 3ds max model, this could be one of the reasons.



UV Vertices

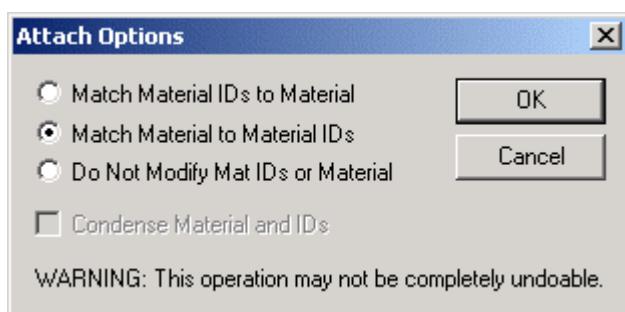
It is important to think about how UVs are setup in your artwork as 3ds max stores UVs per vertex per polygon but RenderWare Graphics stores a single UV per vertex. If there are multiple UVs per vertex in 3ds max artwork the RenderWare Exporter will need to duplicate vertices on export. Duplication of vertices prevents efficient tri-stripping of the artwork. See *Texture Repeats* on page 50.

The other consideration when assigning UV values in 3ds max is that your target hardware platform may have limitations on the UV range it can support. If your UVs value exceed this range then rendering artifacts may occur.

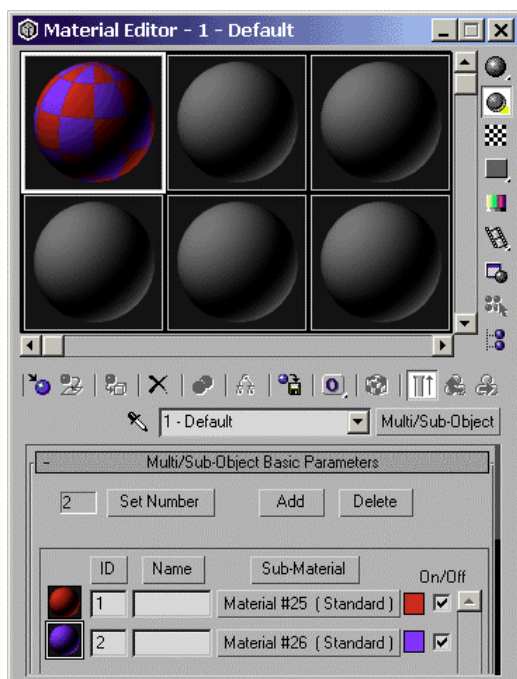
A balance needs to be struck between increasing UV values to ensure good tri-stripping and staying within hardware UV limits. The PlayStation 2 exporter has options, which help you to restrict UV values to set ranges and can be set using the *Limit UV Options* in the PS2 Project Template in the RenderWare Exporters.

4.4 Attaching Geometry

It is sometimes necessary to build a model in parts and attach them. 3ds max has an *Attach* tool which allows you to join all the adjacent parts of a world's surface geometry. If you have more than one material, however, the material assignments can get messed up when parts are joined. To avoid this be sure to select the middle option in the *Attach Options* dialog – *Match Material to Material IDs*. This will keep the texture material assignments of the various components intact.



Once all the component parts are attached, go into the material editor and select the eyedropper tool. Click on the object to create a new Multi/Sub-Object material. Name this material because if you create or merge more attached geometry you will get objects with unnamed Multi/Sub-Object materials. If you do, it really upsets the texture assignments.



5. General Reference

5.1 Lighting

RenderWare Graphics handles prelights and dynamic lights very differently. Prelights are baked on light effects which are cooked into the model's vertices whereas dynamic lights behave more like real lights. The big difference in RenderWare Graphics is that prelights are fast whereas dynamic lights can be animated. So it's a good strategy to use prelights for the stationary parts of your model, walls, floors, and save the expensive dynamic lights for characters, etc.

Omni Lights

If you use Omni lights only RGB and Multiplier values are recognized by RenderWare Graphics. To control a light's falloff over distance (decay rate) the exporters use the 3ds max's Omni light's Far Attenuation Value. Set the Start value to 0.0 and the End to whatever you need it to be.

Vertex lighting is good for static geometry although it is only as accurate as the tessellation of the geometry.

The Render To Texture in 3ds max 5.0 gives great lightmapping results.

Ambient Lights

If an ambient light has a value other than black (0,0,0 RGB) then it is available to be exported.

Ambient lights are a component of .rws files. To ensure that ambient lights are exported:

1. In Advanced Settings (*RenderWare* → *Advanced Settings*)
2. Select the Project Templates tab
3. Select the template you are working with, for example, if you are exporting to a PC choose Generic.
4. Create a new template based on an existing one. To do this, click on *Options* and choose *Add*.
5. Expand Output
6. Expand *Export RWS File*
7. Set *Embed Ambient Light* to TRUE

The light will be exported in the .rws file.

Prelights

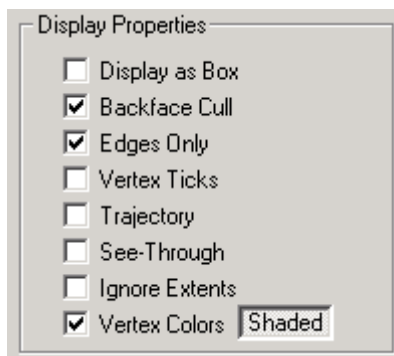


RenderWare Graphics geometry lit using Color Vertex prelighting

World (static) geometry can be prelit in 3ds max *before* it is exported to RenderWare Graphics. This gives the artist control over how the scene will appear in the final application. The density of the geometry affects the shading quality and light fall off – more triangles equals better shadows. This is an unavoidable overhead so really you need to decide how much you can afford to spend on shading geometry before major modeling work begins. Conversely, you would require many RenderWare Graphics lights to create the same effects, which requires a far greater processing overhead. So all in all, prelighting is a relatively inexpensive solution.

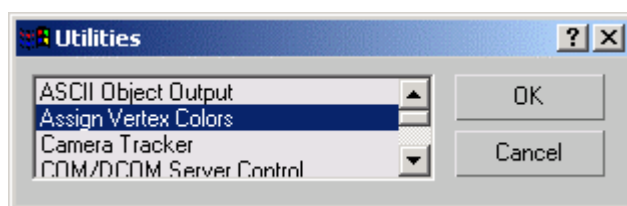
The general procedure for prelighting is:

1. Map and light your scene.
2. Turn on the Vertex Colors display property for any objects that you want prelit.

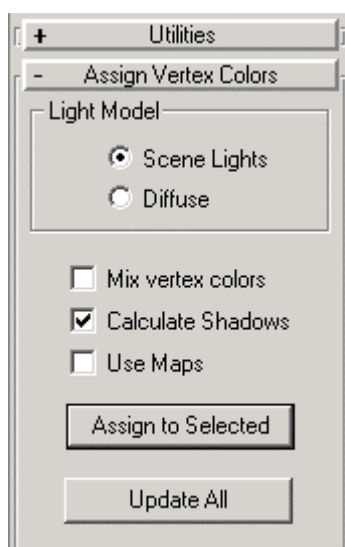


3. Select the objects you want to prelight.

4. Open the Utilities panel and click *More ...* then choose *Assign Vertex Colors* from the list. Leave the options at the default.



5. Click *Assign to Selected* to apply the Vertex Color modifier to the selection.



To view the results of exports use the RenderWare Visualizer.

Prelighting has several advantages: all 3ds max light types can be used; all their parameters are taken into account; shadow casting can be calculated; and the results should remain consistent between 3ds max and RenderWare Graphics.

No negative prelights: Prelights cannot be negative in RenderWare Graphics.



The default GameCube pipeline does not allow you to modulate material color with pre-light vertex values (this is to give TEC stage). If this is the case you can:

- combine material color you want to pre-light
- use MatFX multi-texturing extensions to add multi-texturing effects to your materials which use the material color.

Duplicate Vertices

When vertex pre-lighting is applied, *Static World* assets generate duplicate vertices at the same location. This is because two or more differently colored lights flooding adjacent triangles with their respective colors with no overlap cause duplicate vertices to be created.

For example, for a cube, the different faces would be lit with different colors causing vertices to be duplicated as there is lack of continuity. To resolve this, try making all lights the same color, i.e. all black or all white.

Light Instances

When duplicating lights that perform a particular function, clone them as *instances*. Now you only have to tweak the original to affect all the others. This way you can globally alter the lighting in your worlds quickly and easily.

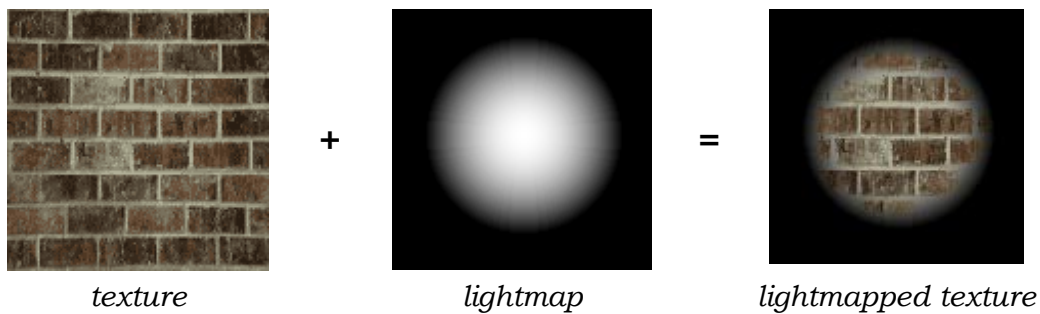
Dynamic Lights

Any lights in the export hierarchy will be added to the clump. Ambient, directional, point and spot lights are supported and RenderWare Graphics supports color, intensity, decay rate and spot angle for these light types.

Ambient is the cheapest light to use then in order of cost are: directional, positional, and spot, each of which is roughly twice as expensive as the one before with a soft spot possibly more expensive than a spot on some platforms. It is recommended that excessive numbers of lights are not used as performance will be affected.

LightMaps

Lightmaps store information about the illumination of a surface. Each surface in a scene can have its own lightmap. Lightmaps work by taking a texture and a lightmap and combining them at render time to create a lit texture.



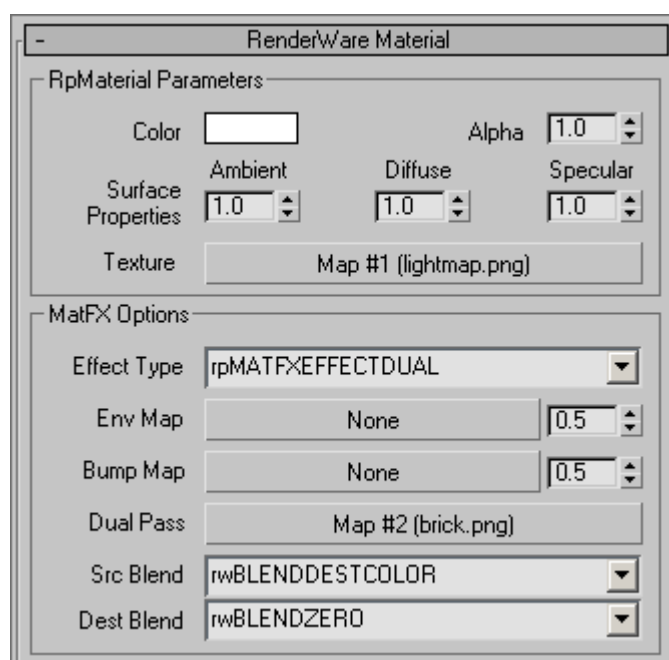
Lightmaps can be created and exported from 3ds max in three main ways. These are:

- Setting up a dual pass material within 3ds max using the RwMaterial dual pass settings and enabling the lightmap flag. You will need to set the *Process Lightmaps* option in your asset template to *Export Native Lightmaps*. Rather than being exported as a standard dual pass material a RenderWare Lightmap material will be exported using the dual pass texture as a lightmap. The lightmap Uvs will be re-sewed for better performance at runtime and the lightmap textures will be re-sampled for use by the RenderWare pipeline.

- Exporting the native light mapping created using the Render To Texture functionality of 3ds max. As above you will need to set the *Process Lightmaps* option in your asset template to *Export Native Lightmaps*. The lightmap Uvs will be re-sewed for better performance at runtime and the lightmap textures will be re-sampled for use by the RenderWare pipeline.
- Exporting light mapping creation data (axis info) for the generation of light maps externally, using the RtLtMap toolkit. You would use standard single texture materials within 3ds max and lightmaps would be added after export. In this case the *Process Lightmaps* option in your asset template should be set to *Generate RtLtMap Uvs*.

Manually setting up dual pass materials for lightmapping

A lightmap material could be created using RenderWare Material setting similar to those shown below. See section 4.2 for more information on setting up dual pass RenderWare materials. You will also need to set the lightmap material flag to tell RenderWare that this dual pass material should be exported as a lightmap.

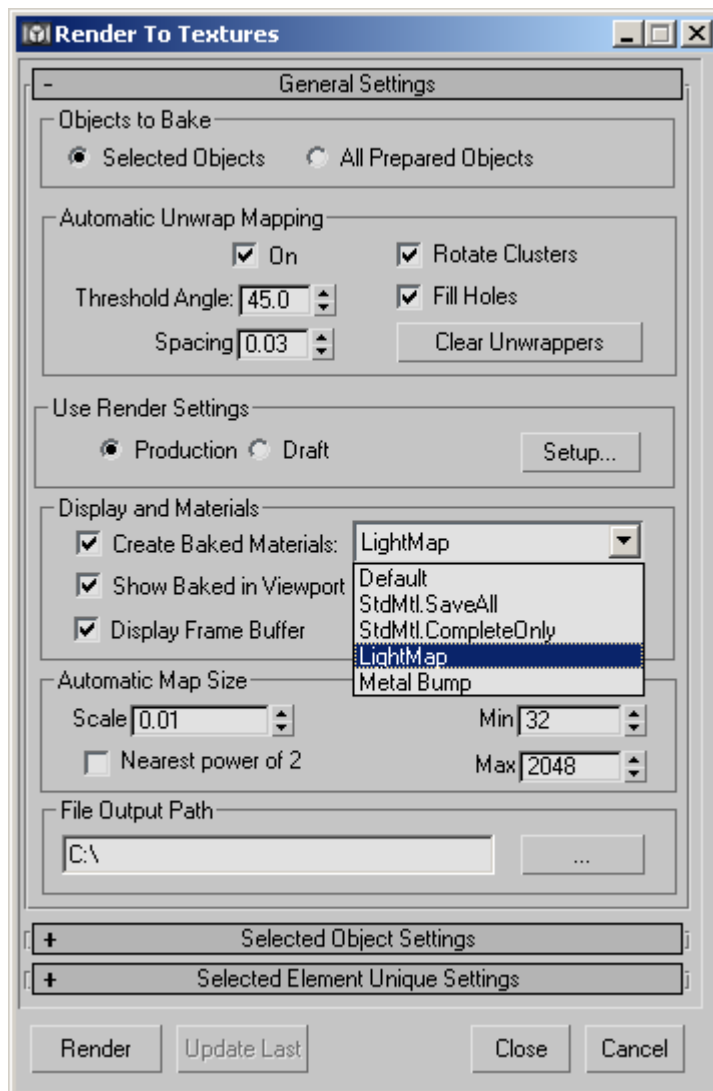


Exporting Native 3ds max lightmaps

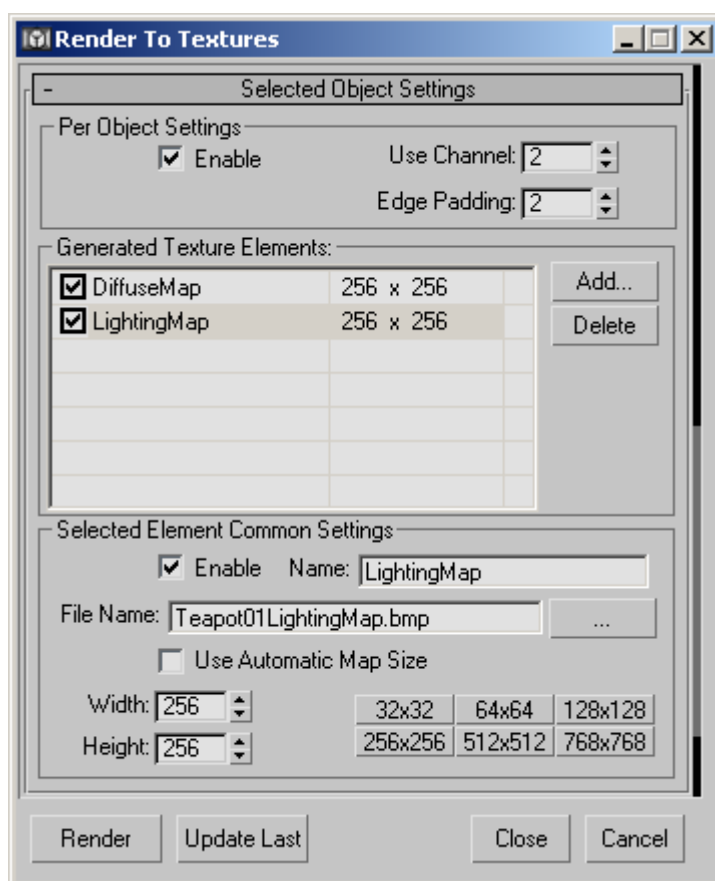
3ds max version 5 and onwards has support for creating and displaying lightmaps. There are various controls for how the created lightmaps should be applied and the recommended method for generating lightmaps varies slightly between 3ds max 5 and 6.

3ds max 5

The method we recommend under 3ds max 5 is to create baked shell materials and to save the lightmaps in the lightmap Viewport Shader slot.



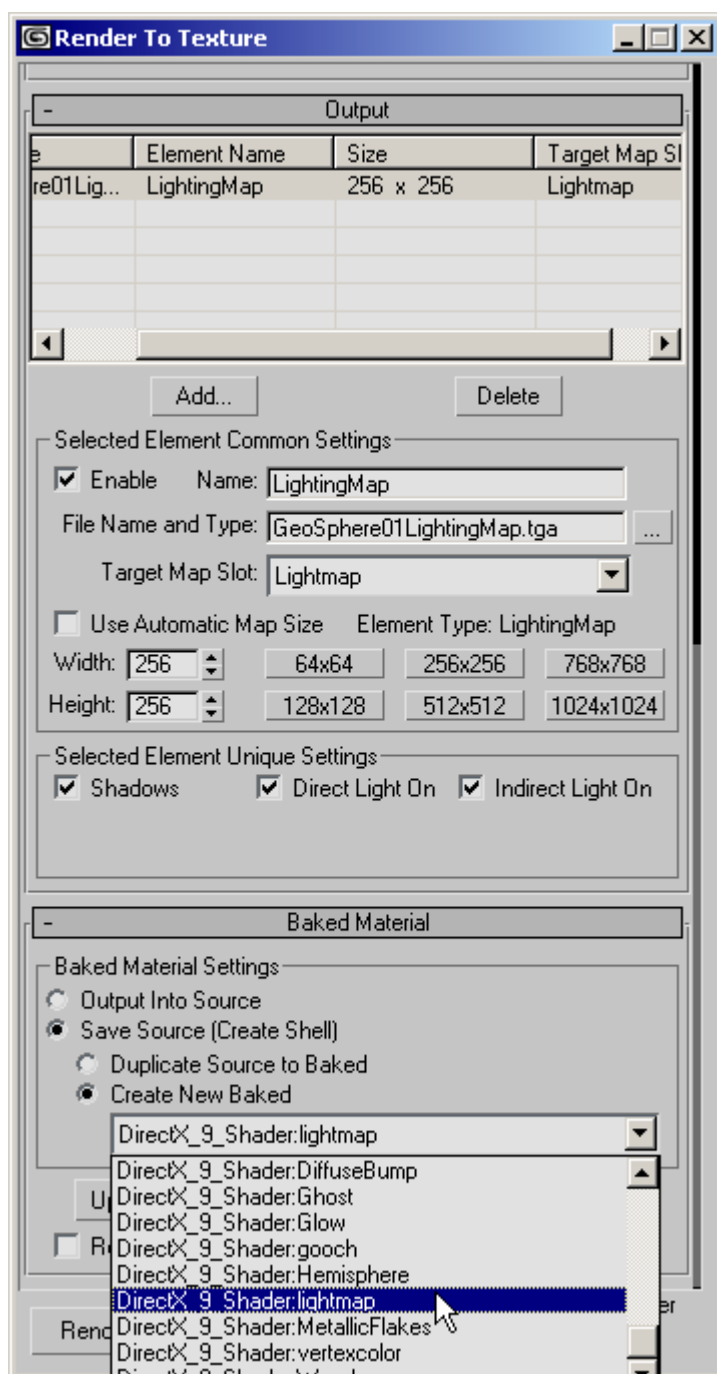
This will create a lightmap per object and set it on the viewport shader lightmap extension of the existing materials in the scene. When using the LightMap extension as the Render To Texture target, 3ds max 5 requires the creation of a diffuse texture as well as a lightmap, this diffuse texture shares the same UV set as the lightmap. You will probably not want to use the burnt diffuse texture and so you can turn it off manually in the viewport extension to re-gain use the original diffuse texture.



- Under 3ds max 5 it is recommended that lightmaps are rendered to UV mapping channel 2 and that the base diffuse texture uses UV mapping channel 1 while exporting lightmaps.

3ds max 6

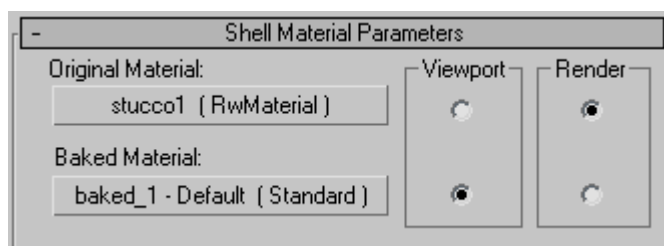
Under 3ds max 6 it is no longer possible to generate lightmaps into the Viewport Shader so we recommend you bake a new DirectX9 lightmap shader and store the lightmaps in the “Lightmap” target map slot. It is no longer necessary to create both a diffuse map and a lightmap.



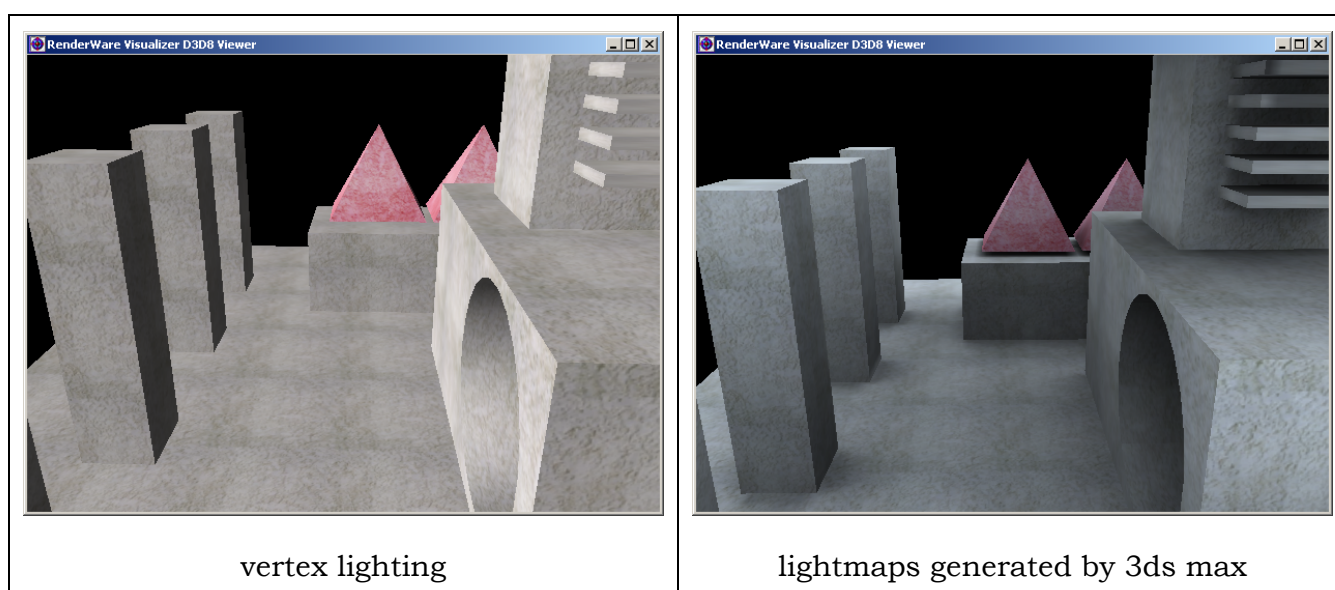
- Under 3ds max 6 it is recommended that lightmaps are rendered to UV mapping channel 4 as the DirectX9 lightmap shader is setup to use this slot. The exporters will attempt to use whichever slot is defined in the shader so feel free to modify it.

Baked Materials

Because some materials can be shared between objects the lightmap info is stored in the baked material section of a shell material with the original material being stored in the original section. During an export to RenderWare Graphics the lightmap and diffuse texture will be taken from this baked material to create a new material. If either are not available the original material will be used for the missing info. If the original material is a multi/sub material and a diffuse texture is not available within the baked material then a material will be created for each sub material; each containing a reference to the lightmap.



- The viewport shader extension and DirectX9 shaders for lightmapping are only available while using the DirectX viewport device.



As shown above lightmaps can make a vast difference to a low polygon scene.

Generating lightmaps externally

Lightmap settings for generating lightmapping info for creating lightmaps externally for use with RenderWare Graphics are explained in the LightMaps section. The *Process Lightmaps* option in the Asset Template should be set to *Generate RtLtMap UVs* for this type of lightmap support.

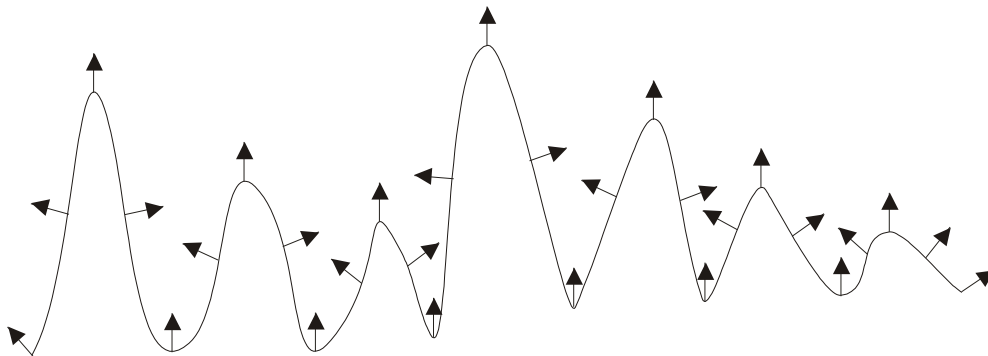
Platform specific settings

When exporting lightmaps for use on Playstation 2 it is vital that you use a project template that specifically targets the platform (such as the PS2.rwt that ships with RenderWare). This is because RenderWare lightmapping on this platform works quite differently from the other platforms, it actually uses darkmaps rather than lightmaps. The Playstation 2 template therefore sets options that generate darkmaps rather than lightmaps. If you view lightmap assets that have been exported for Playstation 2 on other platforms they will look incorrect.

Normal Maps (Xbox and D3D9 specific)

Bump mapping describes how the normal vector is perturbed across the surface so that the differing direction of reflection will result in lighting changes. Bump mapping works by creating embossed effects. Using bump maps you are unable to derive true diffuse/specular/reflections from the bumps. Bump mapping can be used to calculate normal maps.

A normal map is similar to a bump map but instead of containing color values (as RGB) it contains normal values (as xyz). This results in better lighting calculations and better results than bump map models. Normal maps can render real bump maps using DOTPRODUCT3.



normals - gray scaled bump map

You are able to provide a bump grayscale alpha texture and it will be converted to a normal map by treating each pixel in the picture as a vertex of a triangle. The values are calculated using a scale factor based on relative values between the four surrounding pixels.

The normal map is then normalized using the range [0..1].



Gray scale bump mapped texture

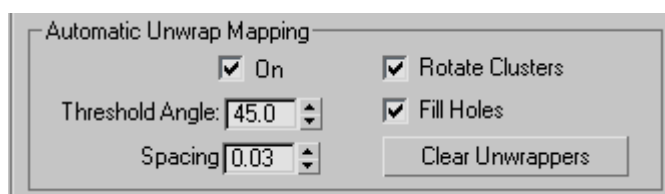


Normal mapped texture

Using normal maps can reduce the number of polygons required to make a model look more realistic.

We'll go through a short example of creating and exporting a normal map from 3ds max. Note that you could create your normal map in other application or use the ones you already have. You can also specify an grayscale alpha texture in `RwMaterial` and leave it to the exporter to generate a normal map for you.

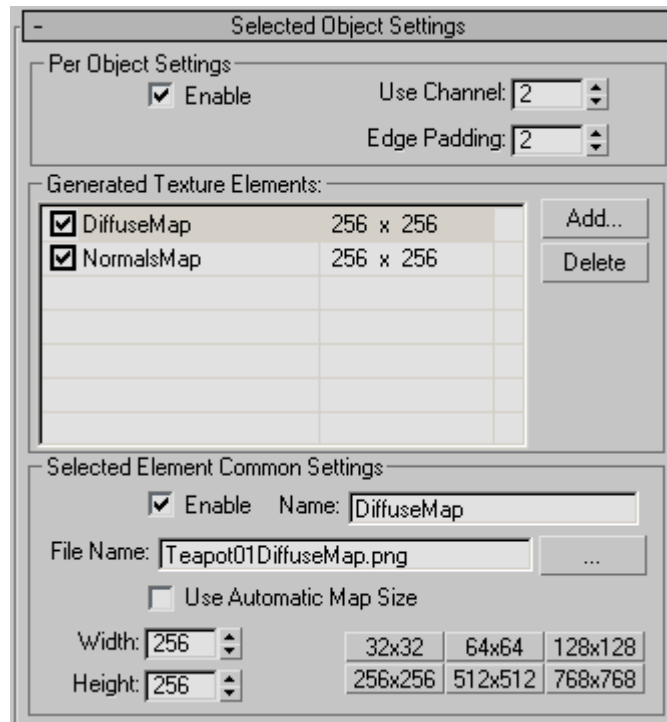
1. Create a simple object and assign a Standard Material to it.
2. Add a diffuse map to the material and a Noise map as a bump map.
3. Now go to Render To Texture (RTT) dialog and select your object. Click Add button in Selected Object Settings and add the DiffuseMap and NormalsMap.
4. If you don't want RTT to generate UVs for you, but to use the ones you've specified, you can turn off Automatic Unwrap Mapping in General Settings. The default is on.



Change the Use Channel option to 2 or 1. This will specify to which channel RTT should generate the UVs to. Normal Map plugin in RenderWare Graphics supports only these two combinations:

5. First UV set for both diffuse and normal map.
6. First UV set for diffuse-map and second for normal map

We'll set it to two. Make sure you don't leave it on default value which is three in which case we will export three UV sets and the plugin would fail to choose the right ones.



7. Press Render button.
8. If you've used RTT to create Baked Materials with Metal Bump option, go to the Viewport Shell material RTT has generated and make sure you have your original diffuse map, that you want to export, set up (RTT doesn't do that for you if you've used Metal Bump option) and your new rendered normal map in the Metal Bump Viewport Shader. Exporter is going to process only Viewport material in the Shell material and is going to export the standard diffuse map and the normal map in the Metal Bump Viewport Shader.

If you haven't used bake materials option, then you need to add the Normal Map as described in the Metal Bump Viewport Shader to the assigned material.

If you are using RwMaterial you have Normal Map Extension settings where you should set the map. RwMaterial also has additional set of options regarding normal maps.

There is an ini file BakeTextureMappings.ini specifying additional ways for RTT to set up maps after they are rendered. You can customize it in your own way so it fits the way you're working. For additional information refer to 3ds max help.

Normal maps options in RwMaterial are explained in the Normal Map section.

More information about DOTPRODUCT3 and information about the Adobe® PhotoShop® Normal Map Plugin which creates normal maps from height maps for per pixel rendering can be found:

http://developer.nvidia.com/view.asp?IO=Simple_dot3_bm

5.2 Camera

RenderWare Graphics supports the exporting of cameras. There are no camera specific export options in the user interface.

Any cameras in the export hierarchy will be added to the clump. Both perspective and orthographic cameras are supported. The camera parameters exported by RenderWare Graphics are matrix, projection type and clipping planes.

Background color

The background color of a scene can be exported depending on the Background Color setting in the rws options. The default setting is off, but it is enabled when a scene is viewed rather than exported. The color is extracted from the global rendering environment (Rendering -> Environment -> Background Color).

5.3 Animation

HAnim Animation

Linear transform and linear rotation keyframes can be exported from 3ds max to RenderWare Graphics. But scaling is not supported by *any* of the keyframe options.

To animate keyframes in the normal way, the animation data will be exported as a corresponding child Animation asset along with the Animated Hierarchy containing the geometry. The assets will be exported to one .rws file.

Legacy file formats - .anm and .dff

To animate keyframes in the normal way, the animation data will be exported as a corresponding .anm file along with a .dff file that contains the geometry. You can export multiple .anms to use with a single .dff, but the hierarchy in the .dff must be exactly the same as the hierarchy in each .anm.

These legacy files formats can be set by the option *Export Legacy Files* in *RenderWare* → *Advanced Settings*, Project Templates tab.

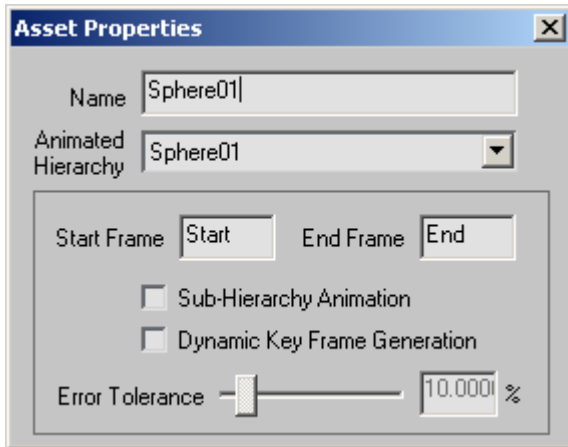
In the Asset Template, the Animated Hierarchy template refers to .dff file format and the Animated template refers to .anm file format.



Animation Curves

RenderWare Graphics only supports linear interpolation between animated keys. Tension, continuity and bias (TCB) values are not exported. So it's best to use linear controllers in 3ds max, that way your animation will export exactly to RenderWare Graphics.

The only way to recreate the TCB 3ds max controls is by adding keyframes. The exporter will do this if the *Dynamic Key Frame Generation*, in Asset Properties, is unchecked. This automatically tries to reproduce the spline motion, although this can add a lot of extra keys.



Rotation Keyframes

Animating 360° rotations with a single keyframe can cause problems due to the math involved. A better solution is to use multiple keys each with a smaller rotation. For example, to rotate an object 360° over 30 frames place one 180° key at frame 15 and a second 180° key at frame 30, rather than a single key at frame 30.

Tension Continuity Bias (TCB)

RenderWare Graphics supports linear interpolation only on translations. It does not support TCB. However to simulate TCB keys you may use the dynamic keyframe generation option to approximate the curve between the keyframes, but you will end up with many more keyframes.

In 3ds max we add extra four keyframes between each two keyframes at 5%, 15%, 85% and 95% positions which will result in:

```
newNumKeyFrames = numKeyFrames*5 - 4;
```

Linear Interpolation

RenderWare Graphics uses Linear interpolation keyframes for translations and rotations. When it detects Bézier interpolation it attempts to add keyframes but this can create inconsistencies. Converting all the keyframes to Linear interpolation should fix any problems.

Automatically Creating Animated Assets

The way the exporter works is that if geometry has no animation and is seen as a static object then it will create a node that defaults to a *Static World*.


However, if there is anything that positively identifies the object as a *Animated Hierarchy* (keyframes, morph targets, hierarchies etc) then it will default to *Animated Hierarchy*. So, by creating a translation keyframe for your object at frame one the exporter will default to *Animated Hierarchy* for that object.

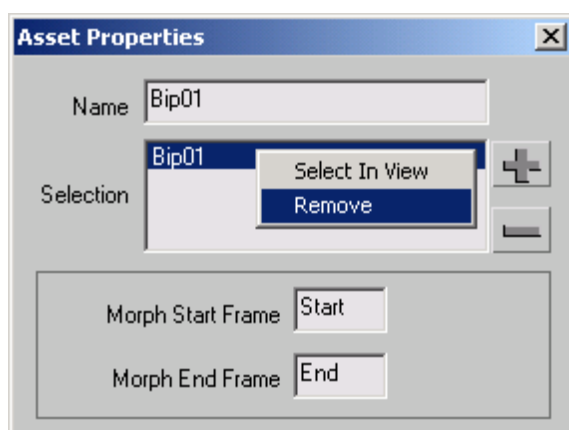
Sub-Hierarchy Animations


Sub-hierarchies can be exported using the Sub-Hierarchy Animation option in the Animation properties dialog.

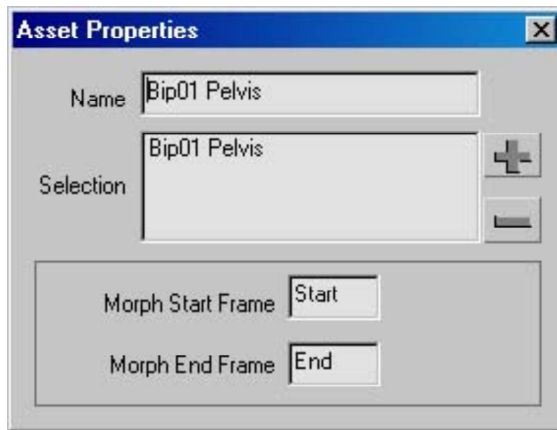
By default, an animated hierarchy will be created from the root. To export a sub-hierarchy, the selection needs to be changed.

For example, an animation contains a character running with his head moving from side to side. You may want to export just the legs separately as a sub-hierarchy. This can be done as follows:

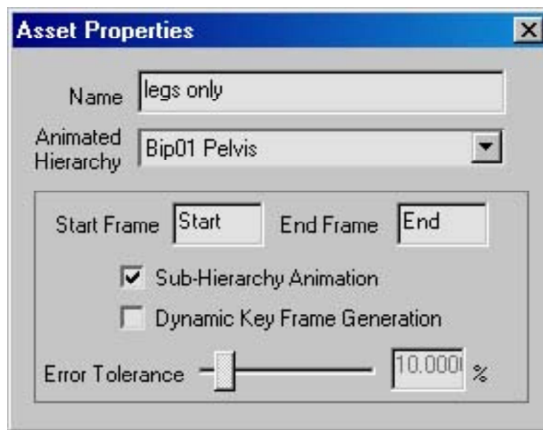
1. Create an asset for the sub-hierarchy animation. By default, the animated hierarchy will be created from the root, regardless of the selection.
2. Double-click the newly created asset, and amend the animation selection by right-clicking the selection and choosing Remove (or clicking on the selection and then clicking ):



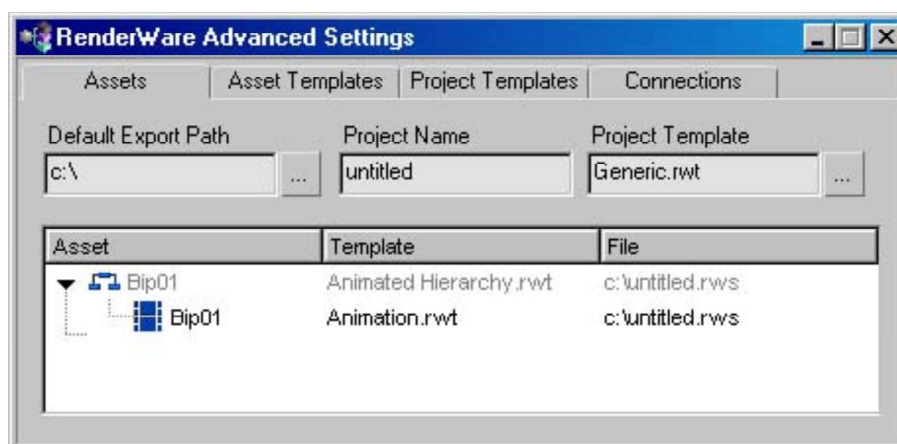
3. Select the node at the root of the sub-hierarchy to be exported, and click on  in the Animated Hierarchy Asset Properties:



4. For the above selection to have any affect on the exported hierarchy, check the Sub-Hierarchy Animation option in the animation's child Asset Properties dialog:



5. Keeping the node selected in the scene view, assign a user-defined tag to the sub-hierarchy. This is most easily done by adding a User Defined Property in the Object Properties dialog. Remember that this tag must be numerical, for example tag=001. More is explained in the Bone Tagging section below.
6. Make a note of the tag used. The programmer will need to know about this tag for the tag to be accessed through code.
7. Disable the Animated Hierarchy asset containing the sub-hierarchy node by right-clicking the asset and choosing *Disable*. This means that only the sub-hierarchy animation file (.anm) will be exported:



8. Export the sub-hierarchy animation in the same way that you would export any animation (*RenderWare -> Export...*).



The Visualizer can not view sub-hierarchy animations. If you need to view the effects of your sub-hierarchy animation, see your programmers for a suitable tool.

Animating the Root Transformation

When exporting a hierarchy, with an animated root node, the animation for the root is generated relative to the root's initial frame. If you create multiple animations, each one with a different initial frame, mixing and matching them on the same hierarchy might lead to inconsistent world space jumps. To avoid this problem, you can create an additional dummy transformation at the top of your hierarchy, causing it to be your new root, and allowing the generation of absolute animation data on your old root.

Morph Target and DMorph Animation

When *Export RpMorph Targets* option is enabled RenderWare Graphics exporter samples the geometry's vertex and normal positions at user specified intervals and later RenderWare Graphics interpolates between them. They can be used with other modifiers except Morpher. The greater the number of samples the better the fidelity but the bigger the file sizes. One issue, however, is that interpolation between targets is linear. Because of this you may need to create more targets to keep object distortion to a minimum.

Delta morphing is much closer to Morpher modifier in 3ds max. A base geometry has a number of deltas applied which are also called delta morph targets. They can overlap and morph any combination which gives great flexibility specially when more then one morph target is used in the Morpher modifier.

Difference between RpDMorph and RpMorph Targets

Export RpDMorph refers to the delta morphing (RpDMorph) plugin. We can use RpDMorphInterpolator objects to morph between RpDMorphTarget objects, which are the blend shape Targets that are exported. This works in the same way as 3ds max blend shapes. Multiple morph targets can be blended between making it very powerful.

Export RpMorph Targets refer to the standard morphing RpMorph plugin. This is faster than delta morphing (on PlayStation 2 it is much faster) but interpolation is between only two morph targets at once.

The two options are mutually exclusive.



Note that morphing is relatively more expensive (i.e. slower) to process than skinned bones. Because of this it makes sense to divide your model into morphing and non-morphing parts if possible and then animate only the parts that change.

Timing

If the deformation appears 'different' in RenderWare Graphics it is because the default output from the Exporter creates a morph target on every 5th frame.

Two ways round this are:

- Output a morph target more frequently (will be more accurate but increase file sizes) or,
- Animate your stuff with this in mind up front. In other words place your animation events at 'key' points (1,5,10,15,20,25,30,35 etc). You should get an accurate match then.

The final point to bear in mind is that all animation is exported as "Linear" keyframe type. Function Curves, TCB values etc are not taken into account. Ensure that all translations are linear.

Bone Limit

The PlayStation 2 RenderWare Graphics implementation has a limit of 64 bones in the skeleton hierarchy. On Xbox the limit varies depending on the exact rendering settings used. Other platforms have a limit of 256 bones.



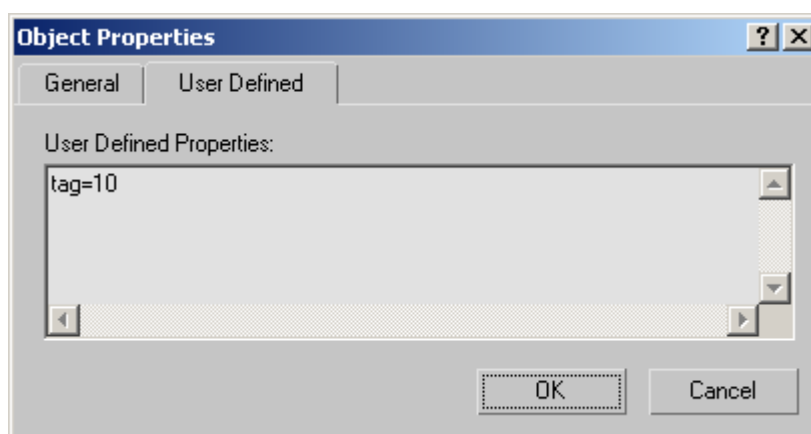
The bone limit can also be found in the Project Templates tab of RenderWare Graphics' Exporter's Advanced Settings.

Alternatively, please refer to the *Skinning* Chapter in the RenderWare Graphics User Guide or the Xbox RenderWare Graphics API Reference.

Bone Tagging

Tagging involves marking objects so that programmers can easily find those objects. A typical use would be to tag the hand joint of a character. The tag number assigned to the bone would then be passed onto your programmers so that they can find the tagged node during the game and dynamically attach objects to it, for example. To tag a bone manually in 3ds max, select the User Properties tab in the properties dialog for the object. Typing the statement 'tag=10' (omitting the quotes), assigns a bone ID of 10 to that bone. Untagged bones will get auto-generated tags starting from the: `max(user specified tags) + 1`.

There are some special rules about Character Studio Biped bones. If not tagged manually they'll be tagged in 1000 and 1104 range, and other untagged non Biped bones in the same hierarchy will get tags from 2000 up, or maximum of user specified tags if that is greater than 2000.



There is a problem with auto-generated tags for Character Studio Biped bones, in non-English versions of Character Studio for example French and Japanese. For these version CS Biped bones will behave as standard 3ds max bones, meaning they will be tagged from `max(user specified tags) + 1`.

Customizing Object Properties

To tag areas of scenes with user defined attributes, object properties can be defined in the User Defined tab of the Object Properties dialog (shown above). All text typed into that User Defined tab will create user data for the associated RwFrame. Every line of text will appear as a userdata string on the RwFrame except for text starting with "tag=", because that is reserved for tagging HAnim bones.

This is explained in more detail in the Blind Data section of the [TechnicalArtistGuide.pdf](#)

6. Exporting

The options available in the export menu are fully explained in [TechnicalArtistGuide.pdf](#). This document explains the basics to take your assets and export them to RenderWare Graphics formats.

The exporters, by default, export to `.rws` files. This is the RenderWare Graphics file format for exporting any type of RenderWare data. RenderWare Visualizer can be used to view `.rws` files.

The file formats `.anm`, `.dff`, `.bsp`, `.spl` are now considered legacy. The binary format of these files will continue to be supported, and RenderWare Graphics will continue to read them. There is no need to re-export existing artwork to `.rws` file format. However, it is recommended that you export to `.rws` files as these file formats *may* be removed in future releases.



File Formats

`.rws` files contain RenderWare Graphics stream information and may include animated hierarchy, static world, animation and spline information as well as embedded textures. `.rws` files can contain the contents of multiple `.dff`, `.bsp` files etc.

The file formats `.anm`, `.dff`, `.bsp`, `.spl` are now considered legacy and may be removed in future releases. It is therefore recommended that you export to `.rws` files.

`dff` files contain Animated Hierarchy (RpClump) information.

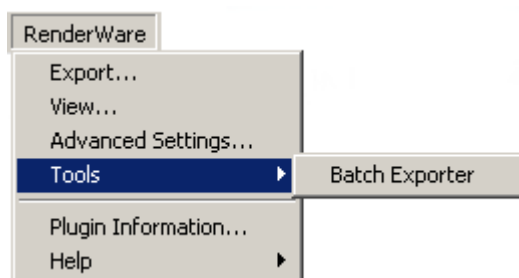
`bsp` files contain Static World (RpWorld) information.

`anm` files contain Animation (RtAnimAnimation) information.

`spl` files contain Spline (RpSpline) information.

6.1 RenderWare Menu

The RenderWare Menu has the following options. These are explained below.



Export

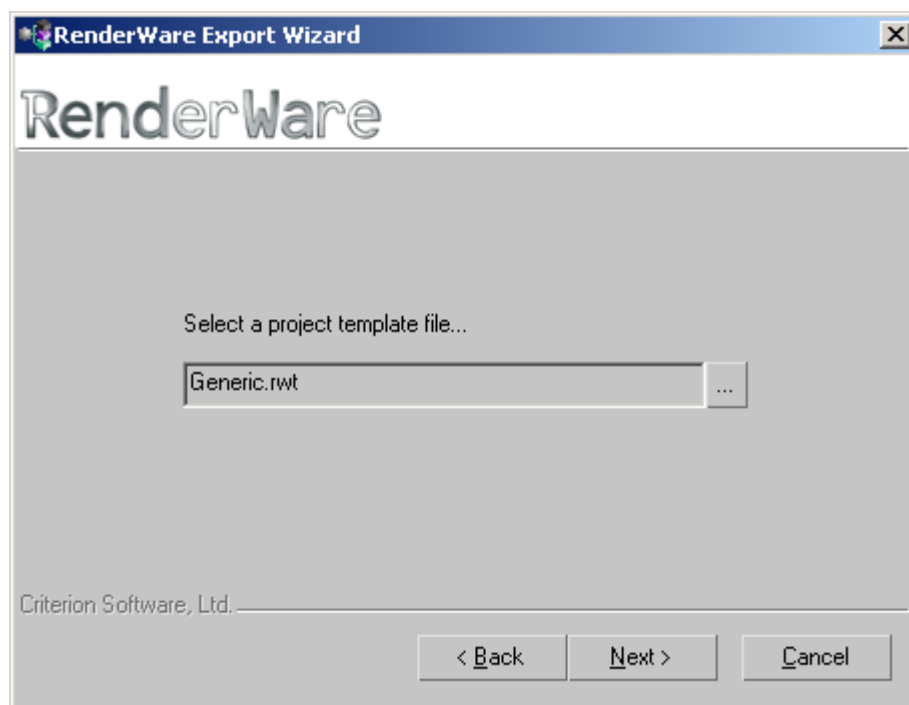
The Export option in the RenderWare menu uses the Export Wizard to automatically create assets and export them to a .rws file.

The first time you perform an export of a scene the wizard will appear and ask for the following information:

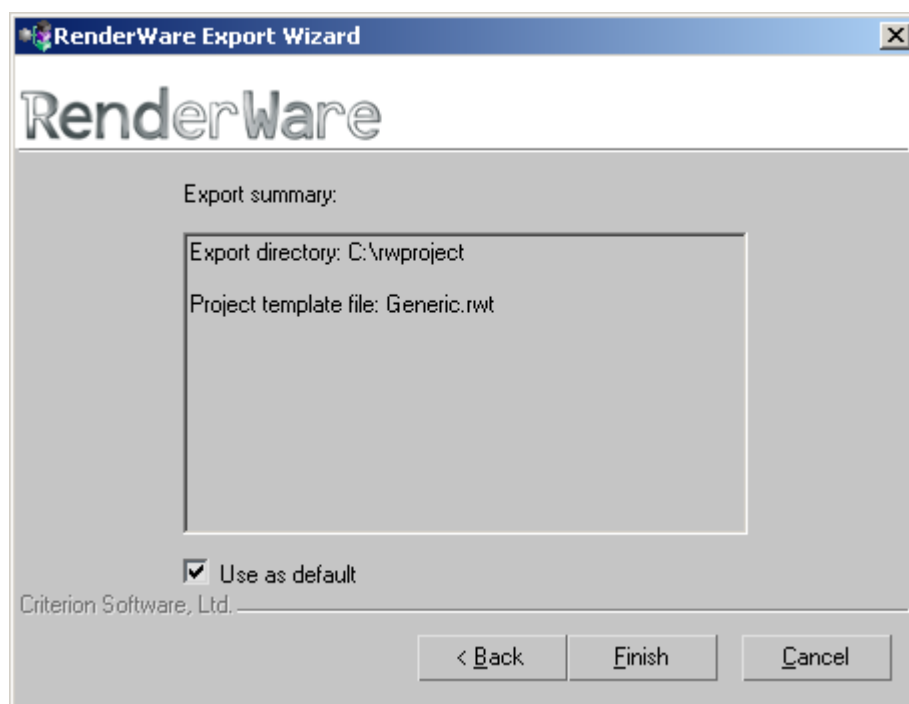
- State your target directory for your exported files



- Select the project template required. There are four default project templates which relate to the four supported platforms:
 - Generic.rwt used for exporting files to the PC (e.g. D3D8 or OpenGL)
 - GameCube.rwt used for exporting files to GameCube
 - PS2.rwt used for exporting files to PlayStation 2
 - Xbox.rwt used for exporting files to Xbox



- The export summary displays the export directory and project template used. Selecting "Use as default" will ensure that these settings are stored as the default.



Your scene is exported using the settings in the project template file you selected. For the default templates the output file type will be .rws. The contents of the .rws (your scene) are displayed in the Output Window.

View

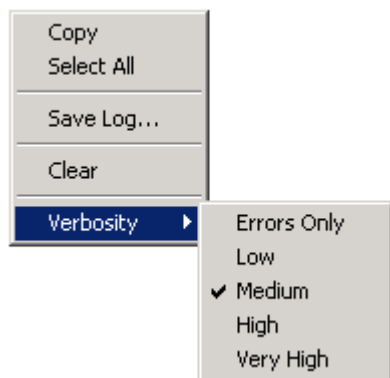
The View option in the RenderWare menu will export the assets in your scene and display them in RenderWare Visualizer. The view operation will override the output settings in your project template and always export a temporary .rws file with textures. The contents of the .rws (your scene) are displayed in the Output Window.

Output Window

When a scene is exported or viewed the output is recorded in the Output Window. The output window contains useful information about the type of asset exported including animation settings, geometry stored, tri-stripping, frames, lighting.

The output window is useful for debugging and comparisons.

There are several options available from right clicking the output window:



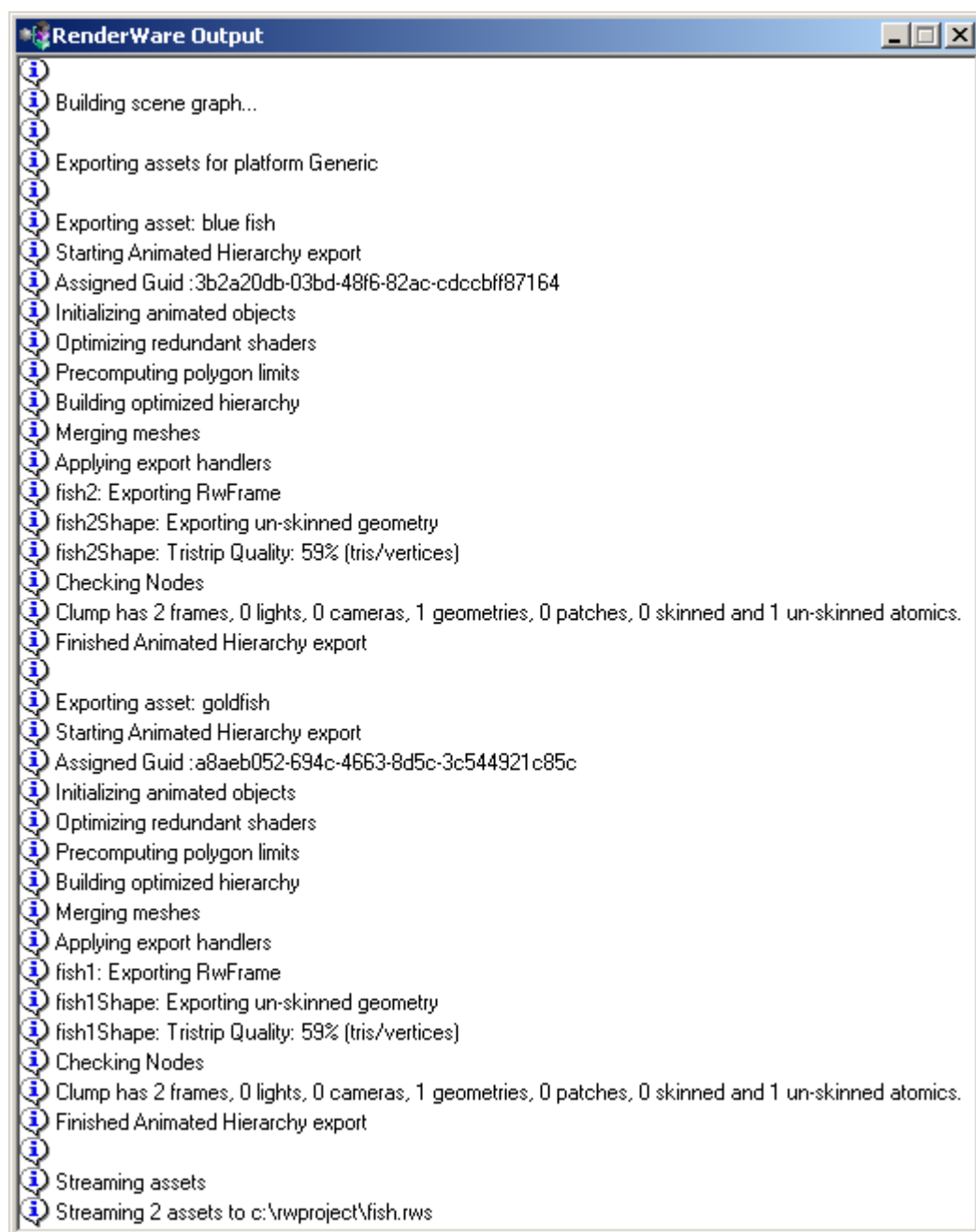
Copy: copies the selected text to the clipboard

Select All: selects all the text in the output window

Save Log: saves the content of the output window to a log file

Clear: clears the output window

Verbosity: changing the verbosity dynamically changes the contents of the output window. Higher levels contain more detailed information useful to advanced users.



6.2 Advanced Settings

Advanced Settings are settings which enable the user to customize assets, asset templates, Visualizer settings and platform specific settings.

Advanced Settings are accessed through *RenderWare* → *Advanced Settings*.

The Advanced Settings dialog has four tabs:

- Assets_ (page 82) - the assets of a scene, for example, a skinned object
- Asset Templates (page 90) - the settings for an asset

- Project Templates (page 91) - the project settings used for exporting assets to different platforms. Supported platforms are: PC; GameCube; PlayStation 2 and Xbox
- Connections (page 92) - the connection settings for RenderWare Visualizer

Assets

Assets are RenderWare nodes stored in your scene that represent a single item of export data. They link a part of your scene with the settings to use on export. For example, a game may contain a skinned animated character. Two assets are created, an asset for the character and a child asset containing the animation data.

There are two modes of operation for asset generation. When the Customize option is disabled the exporter analyses your scene and generates assets automatically for you. As objects are added or deleted in your scene the assets are automatically updated. When the Customize option is enabled you have manual control over how assets are created. In a new scene the default mode is non-customized assets.

Using the Export Wizard (*RenderWare*→*Export* or *RenderWare*→*View*) the assets can be exported and viewed quickly. All assets present in the scene are exported. Using the Asset dialog of the advanced settings window you can export and view assets individually.

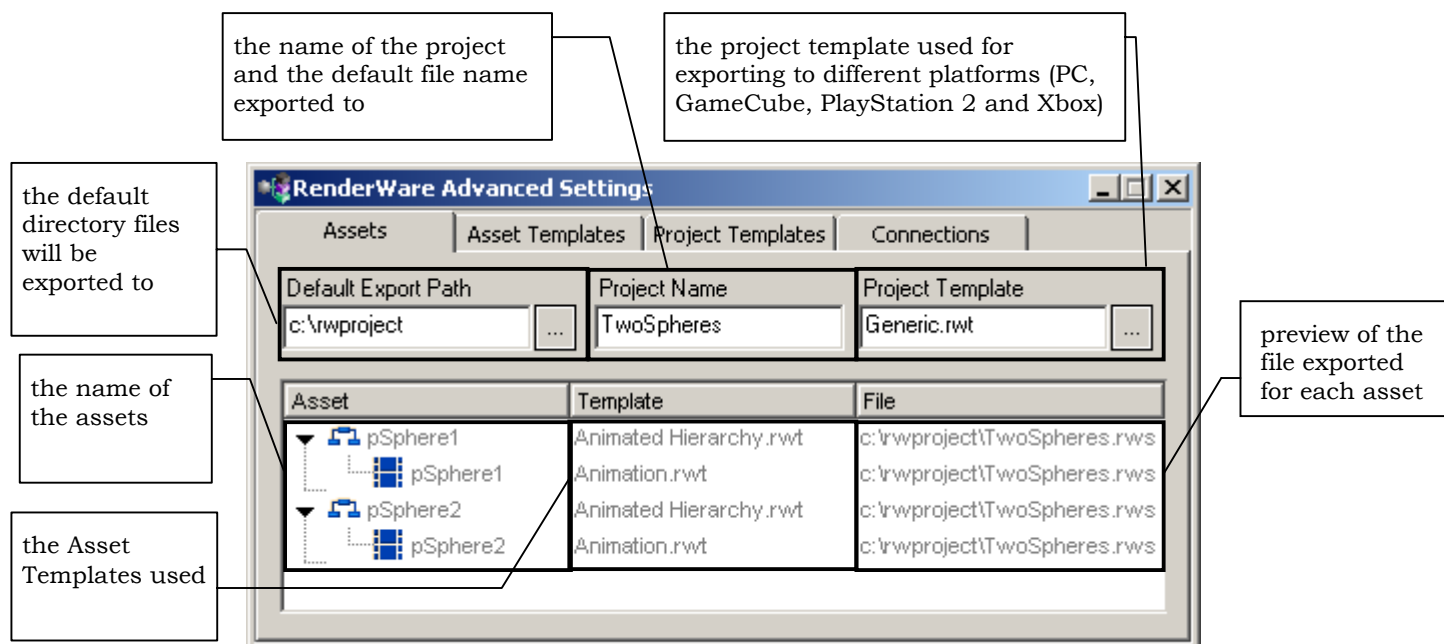
All assets use an asset template. There are four default asset templates and these read-only templates contain a specific set of options and settings.



Assets were referred to as batch nodes in RenderWare Graphics release 3.3 and 3.4.

Asset Dialog

The asset dialog contains settings which are important for customizing existing assets and creating your own assets.



The above example shows automatically generated assets. Right-clicking over an asset enables you to export or view the selected asset. Right clicking below the assets enables you to be able to export and view all assets.

The Assets dialog contains assets which relate to objects in your scene. There are four types of default asset:

- Animation
- Animated Hierarchy
- Static World
- Spline

Customizing Assets

The default behaviour for the asset manager on new scenes is a mode in which assets are automatically created for you as you model your scene. Whilst this is very convenient as you start to model a scene or get to know the RenderWare tools it is likely you will eventually need finer control over your assets. You can take control of asset generation by entering customized asset mode:

RenderWare → Advanced Settings → Assets right click and choose Customize

This activates the customize mode and the Customize option in the right mouse menu is checked.

You are now able to manually edit your assets. The exporter will no longer automatically generate assets as you modify your scene. You can leave customized asset mode by re-selecting the Customize option from the right click menu. The exporter will warn you that any asset modifications you have made will be lost.

Adding Assets

To create your own assets based on a selection:

RenderWare → *Advanced Settings* → *Assets* right click and choose *Add*

This option enables you to create your own assets. Modifying the objects selected

Auto Create Assets

Whilst in customized asset mode you can still have the exporter re-analyse your scene and create assets for you:

RenderWare → *Advanced Settings* → *Assets* right click and choose *Auto Create*

This is similar to automatically generating assets. However, this option does enable you to keep your existing customized assets, if desired.

Adding a Child

An animation is a child of an animated hierarchy. When an animated hierarchy asset is created, an animation child asset is also created. This option enables you to add your own child to an animated hierarchy.

RenderWare → *Advanced Settings* → *Assets* right click and choose *Add Child*

This option can be used on a selected asset which can be customized.

Deleting an Asset

Right click the asset and choose *Delete* or select an asset and press *DELETE* on the keyboard.

Only customized assets can be deleted.

Exporting and Viewing Assets

To export and view assets using Advanced Settings:

- Right click below the assets in the Advanced Settings Assets tab and choose *Export all* or *View all* to export or view all assets

- Right click an asset and choose *Export* or *View*. This exports or views the selected asset only.



Multiple assets can be selected and exported or viewed. Use the Ctrl key or rubberband drag with the mouse to select multiple assets.

Enabling and Disabling Assets

During development stages of a scene you may want to create differing variations of export options. Assets can be enabled and disabled giving greater flexibility to the artist.

- Right click an asset and choose *Disable* an asset. This asset will not be exported when all assets are exported or viewed.
- Right click an asset and choose *Enable* an asset. This asset will be exported individually or as part of a full export or view. By default this option is on.

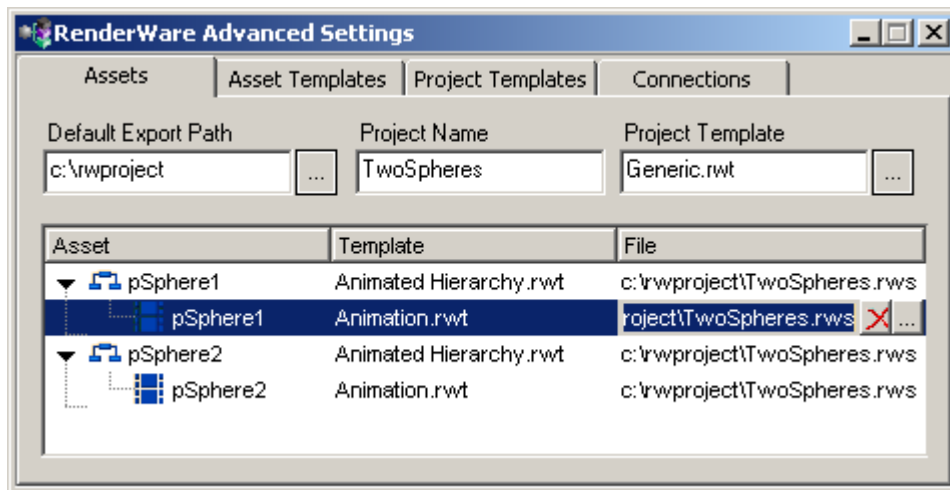
Export Location

The files that the exporter generates during an export are dependent on various settings. The type of files exported is always dependent on the project template in use. By default the file names and locations are calculated as indicated by the following table:

EXPORT FILE TYPE	EXPORTED FILES
RWS	<i>Default Export path + Project name + .rws</i>
RF3	<i>Default Export path + Project name + .rf3</i>
Legacy Files (anm, bsp, dff, spl)	<i>Default Export path + Asset name + .anm or .bsp or .dff or .dma or .spl</i> (depending on the asset type)

The file column of the asset manager provides a preview of the file that will be generated when the asset is exported. When the column displays multiple files with the same name this indicates that the assets will all be added to the same file. In some situations, such as when the project template is setup to export an RWS file and legacy files, the exporter may not be able to provide a suitable preview of the files that will be exported.

The export file location (but not file type) can be overridden per-asset when in customized asset mode. This is done by clicking on the asset filename. You can either type an export filename directly or use the browse button to navigate to a location. If you enter an absolute path it will be used as is. If you enter a relative path it will be displayed and exported relative to the Default Export Path location. Any extension you specify will be ignored as the file type is still controlled by the project template in use. To export multiple assets to one file, define the same export file for each asset.







After the output file has been customized the filename will appear in bold in the filename preview column. This lets you quickly identify which assets have customized output files.

If you later wish to return the asset to its default output file you can either enter an empty string or use the cancel button to clear the custom file.

Asset Properties

The four types of assets have differing asset properties. These assets are based on the four asset templates.

Different symbols are used for the different types of assets. The table below refers to the different asset templates available and the RenderWare Graphics term they relate to.

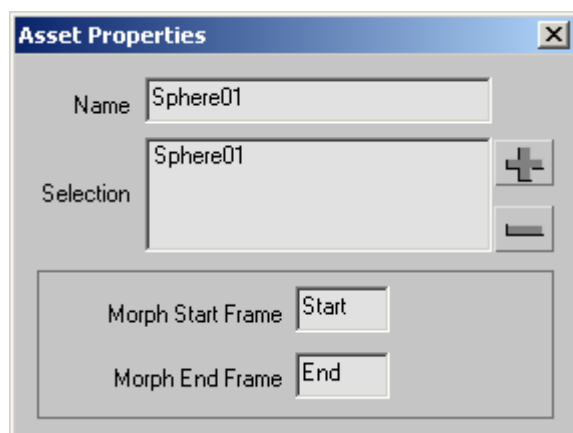
	ASSET TEMPLATE	RENDERWARE GRAPHICS TERM
	Animated Hierarchy	RpClump
	Animation	RtAnimAnimation
	Static World	RpWorld
	Spline	RpSpline

Each asset has properties. To display the property dialogs, double click the asset or right click the asset and choose *Properties*.

Animated Hierarchy Asset Properties

An Animated Hierarchy asset 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 selected, in the Animated Hierarchy Asset Template, the exported file saved can contain the object hierarchy, hierarchical animation, skinned animation, and morph target animation.



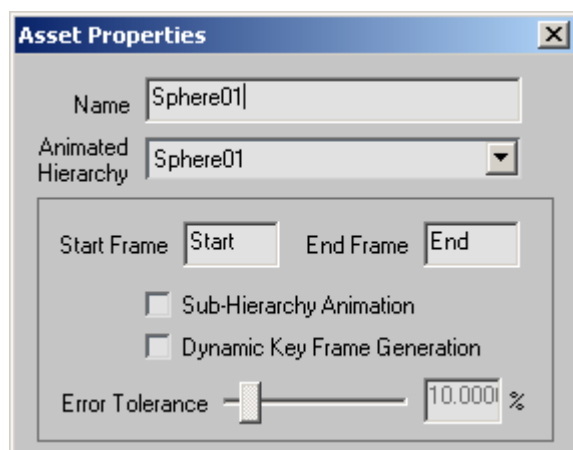
Name: Name of the asset.

Selection: Objects contained within the asset.

Morph Start and End Frames: Sets the start and end range of the animation to use in exports for morphing. "Start" and "End" will query the values from the time slider. Alternatively, you can enter numbers in the fields but ensure that only positive numbers only are used.

Animation Asset Properties

An Animation asset stores only animation data and is always a child of an Animated Hierarchy asset which contains the geometry. It is recommended that the asset names of the Animation and the Animated Hierarchy are identical.



Name: Name of the Asset.

Animated Hierarchy: The name of the parent node.

Start and End Frame: These options set the start and end range of the animation to use in exports. "Start" and "End" will query the values from the time slider. Alternatively, you can enter numbers in the fields but ensure that only positive numbers only are used.

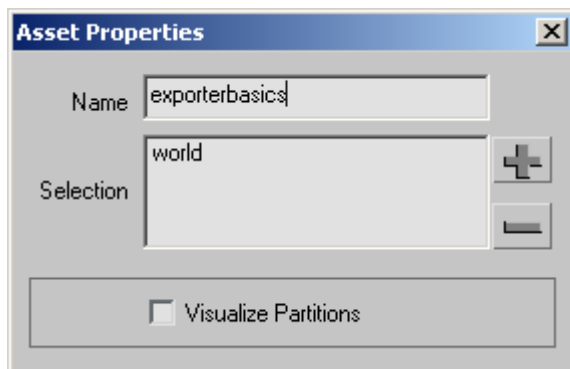
Sub-Hierarchy Animation: This option is used to export part of an animation by selecting the node at the top of a node hierarchy. Exporting with this option set will export the animation from the selected node down to the bottom of the selected tree. The selection used comes from the parent Animated Hierarchy asset.

Dynamic Key Frame Generation: RenderWare Graphics currently supports only linear interpolation, so when non-linear controllers are used, the exporter will insert extra keyframes between the existing ones at the 5% and 15%, 85% and 95% positions.

Additionally you can turn this option on for error checking. This option then adds extra keyframes to your animation although it maybe preferable for the artist to manually add extra keyframes on the most important objects.

Error Tolerance: This option is grayed out unless the *Dynamic Key Frame Generation* option is selected. It controls the percentage error allowed at integer keyframe times before the exporter will add a new keyframe in the exported animation.

Static World Properties



Name: Name of the Asset.

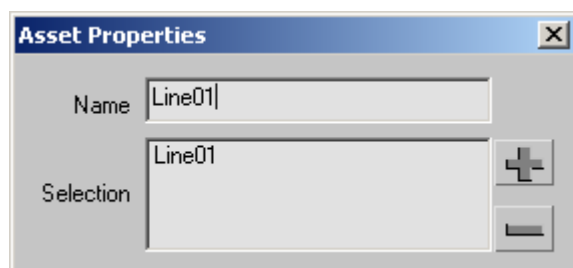
Selection: Objects contained within the asset.

Visualize Partitions: shows the partitions/sectors that are generated as a result of the export process.

Whilst the partitioning of an exported world can be viewed in the viewer, it is sometimes useful, especially when using hints, to view them in the art package.

For more information about partitions and hints refer to [3.15_Partitions](#) (page 28) and [3.14_Hints](#) (page 24).

Spline Asset Properties



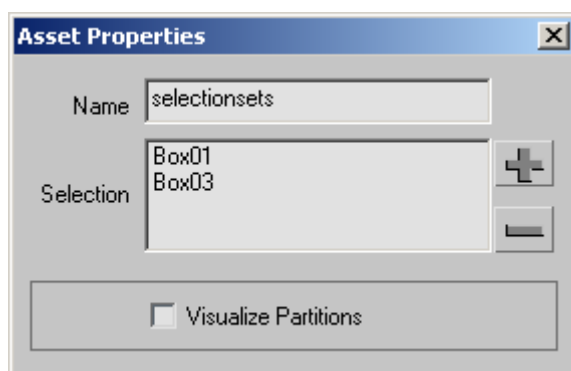
Name: Name of the Asset.



Selection: Objects contained within the asset.

Asset Selection

The example below explains the steps required to manage asset selection.

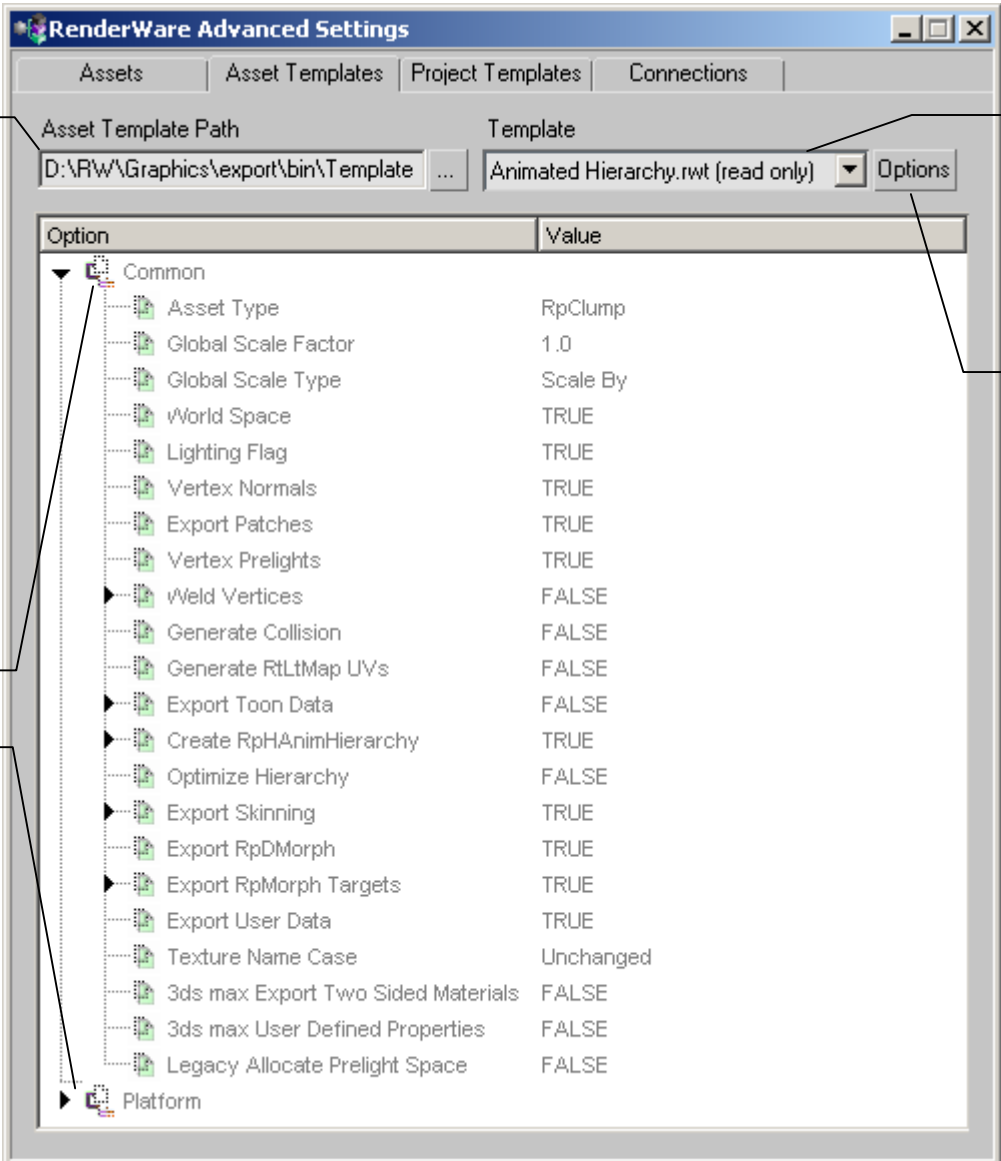
1. Create 3 boxes in a scene
2. *RenderWare* → *Advanced Settings* → *Assets* tab
An asset has been automatically generated.
3. Right click in the Assets tab and choose *Customize*. This enables you to customize your assets.
4. Delete the automatically created "untitled" asset.
5. Select 2 boxes in your scene.
6. In Advanced Settings Assets tab, right click and select *Add*. This creates an asset based on your selection.
7. Double click the "untitled" asset to access the Asset Properties.
8. Rename the asset "selectionsets".
9. The dialog lists the objects included in the asset, for example Box01 and Box03.



-  enables you to add an object to the selection set. To add an object, select an object in your scene and click this button to add the object to the selection. For example, you might want to add Box02.
-  enables you to remove an object from an asset. To remove an asset select the objects in the dialog and click this button to remove the object from the selection set. For example, you might want to remove Box03.

Asset Templates

Asset Templates contain the options settings for exporting an asset in a particular format.



The directory containing the asset templates. Click on the browse button to change the asset template path.

The name of the template currently displayed. The default templates are read only.

Use to add and rename templates.

Two options:
Common - contains asset information which can be customized.
Platform - contains platform specific settings.

Option	Value
Common	
Asset Type	RpClump
Global Scale Factor	1.0
Global Scale Type	Scale By
World Space	TRUE
Lighting Flag	TRUE
Vertex Normals	TRUE
Export Patches	TRUE
Vertex Prelights	TRUE
Weld Vertices	FALSE
Generate Collision	FALSE
Generate RtlMap UVs	FALSE
Export Toon Data	FALSE
Create RpHAnimHierarchy	TRUE
Optimize Hierarchy	FALSE
Export Skinning	TRUE
Export RpDMorph	TRUE
Export RpMorph Targets	TRUE
Export User Data	TRUE
Texture Name Case	Unchanged
3ds max Export Two Sided Materials	FALSE
3ds max User Defined Properties	FALSE
Legacy Allocate Prelight Space	FALSE
Platform	

There are four default Asset Templates which you can use or you can create your own templates. The default templates are read-only and we recommend that you set up your own templates based on the default templates. The default templates are:

- **Animated Hierarchy** - used to export a single hierarchy within a scene as a RenderWare Graphics container object, called `RpClump`.
- **Animation** - used to export animation data. Requires an animated hierarchy with the same name.
- **Spline** - used to export spline data. Requires a static world with the same name to contain the world.
- **Static World** - used to export objects as RenderWare Graphics World, called `RpWorld`. Worlds do not contain any hierarchy or animation information and are typically used for static level geometry in a game.

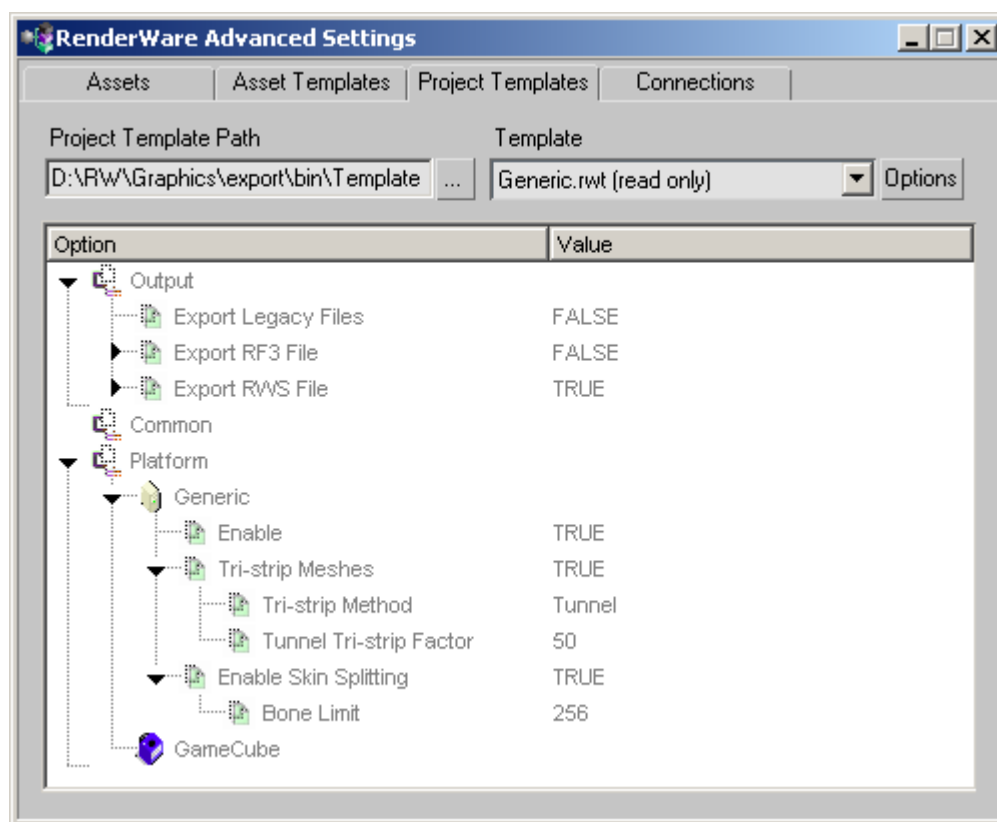


Asset Templates were referred to as batch node properties in RenderWare Graphics releases 3.3 and 3.4.

The options for the default settings are listed below.

Project Templates

The Project Templates contains the scene project settings. Output settings were previously found in Scene Settings in the RenderWare Graphics 3.3 and 3.4 exporters. These options enable the user to export files to `.rws` (the recommended file format), `.rf3` (the new XML file format) and to legacy file formats (`.bsp`, `.dff`, `.anm`, `.spl`).



Project Template Path: The directory containing the project templates.

- `Generic.rwt` used for exporting files to the PC (e.g. D3D8 or OpenGL), exports files with the extension `.rws`.
- `GameCube.rwt` used for exporting files to GameCube exports files with the extension `.rg1`.
- `PS2.rwt` used for exporting files to PlayStation 2 exports files with the extension `.rp2`.
- `Xbox.rwt` used for exporting files to Xbox exports files with the extension `.rx1`.

The Project Templates contains the setup information. The actual project template used is specified in the Assets.

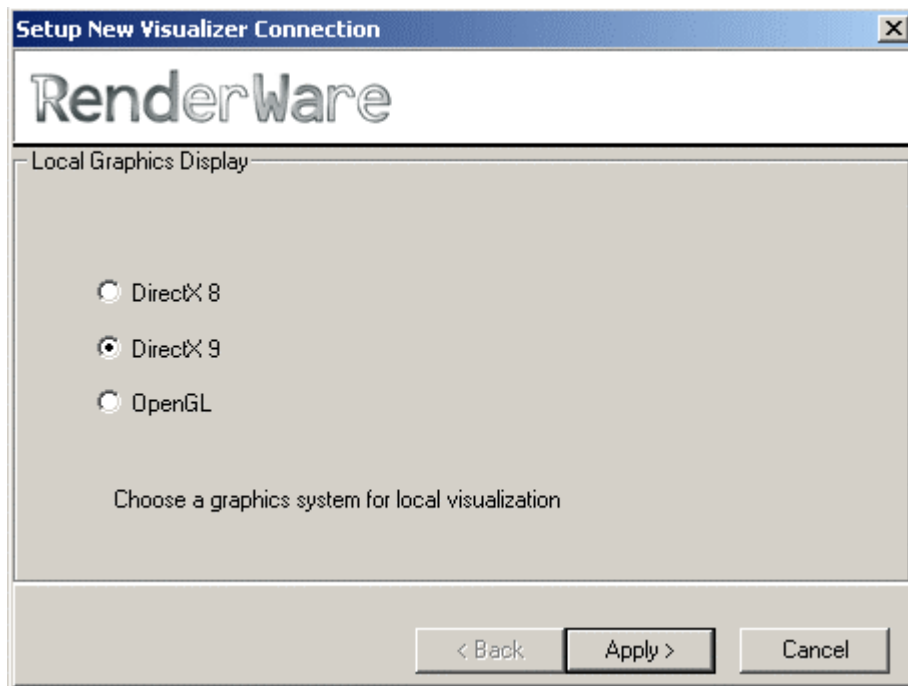
The Output and Platform options are explained in full in [TechnicalArtistGuide.pdf](#).

Connections

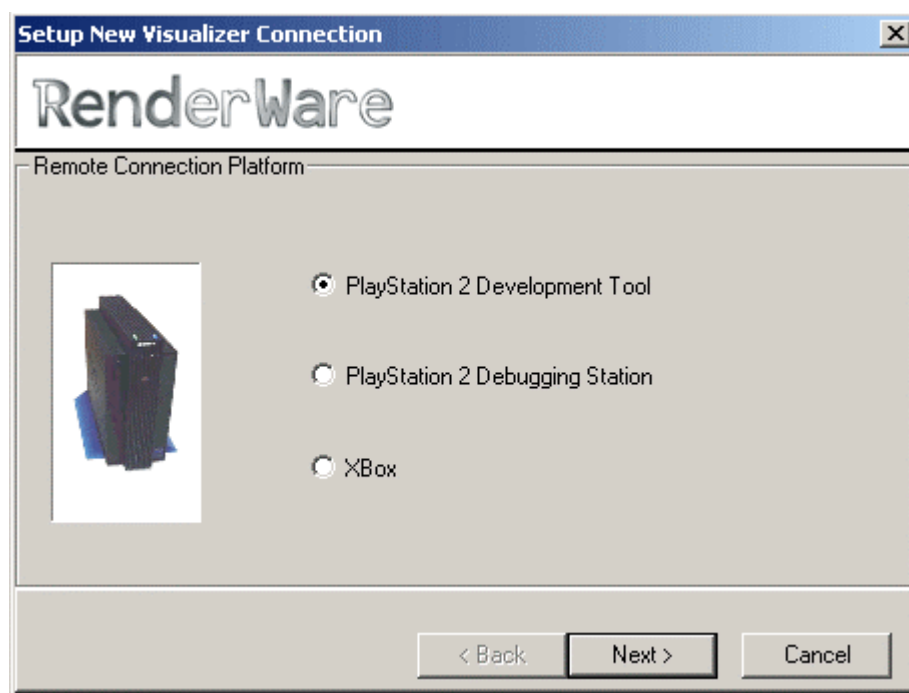
Connections stores the connection settings require to view assets in RenderWare Visualizer.

Visualizer

When you run Visualizer from 3ds max for the first time, a connection wizard will run. Initially, you are given the option of choosing which graphics system to use for local visualization:



Subsequently, you may setup a connection to a console:

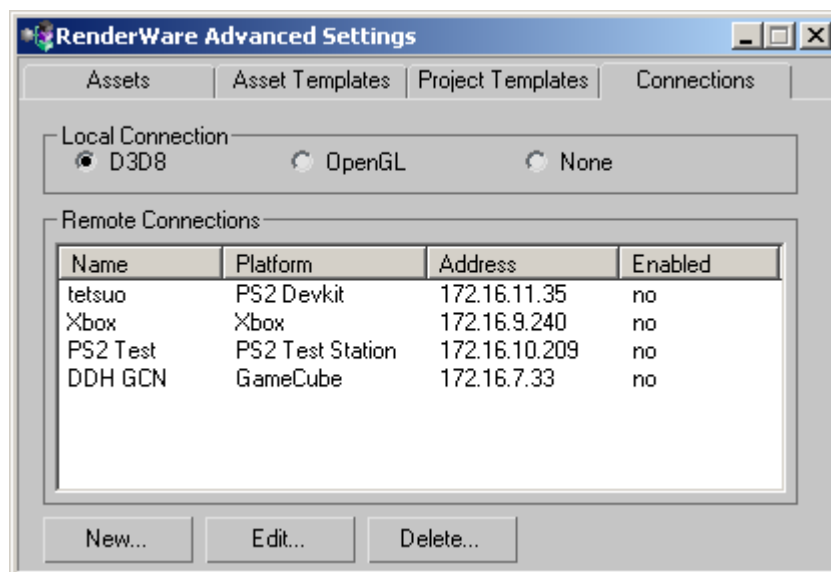


The wizard will guide you through the steps necessary to initialize a connection.

Talk to your system administrator during this phase of configuration. The Visualizer host that runs on the console is able to automatically obtain IP addresses if a DHCP server is present. Failing that, your administrator can provide you with a fixed IP address that can be used by the Visualizer host.

Configuring connections

To change a connection for the PC, select a local connection for D3D8 or OpenGL. Local D3D8 and OpenGL connections will automatically spawn viewers and close them on disconnection.



Connections is divided into *Local Connection* and *Remote Connections* so that you can easily view the rws file on different platforms. Local Connection can be used to view D3D8 and OpenGL on your PC whereas Remote Connections can be used to view rws files on target platforms. More information on setting up the Connection Editor can be found in the *RenderWare Visualizer* viewer document.

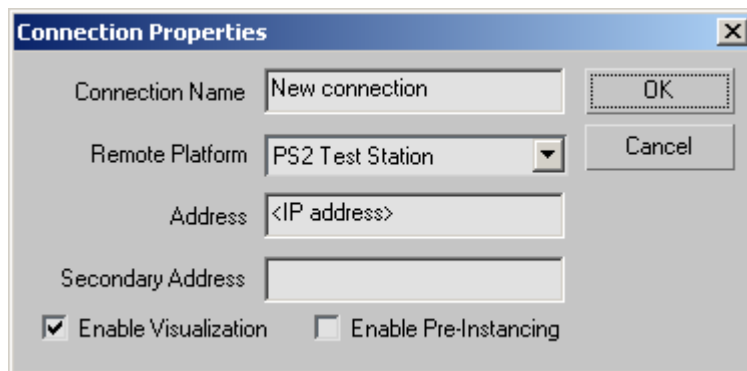
New creates a new connection.

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

To create a new connection click on *New* in Connections.



Connection Name the name of the connection to be made. This name is purely for your reference and does not need to be related to the actual name of the target hardware.

Remote Platform the remote target platform.

Address the IP address or machine name of the target hardware.

Secondary Address for the target platform. This option is only enabled for PS2 Devkit and Xbox targets.

Enable Visualization is used to state which platform viewers will display the RenderWare Export Nodes when the *View* buttons are pressed.

When all assets or selected assets are viewed the .rws file is viewed on all enabled connections using RenderWare Visualizer.

When you view assets using the RenderWare Exporter, viewer.exe is run, the registry is accessed and connections are made to the enabled target platforms. The .rws file is then displayed in RenderWare Visualizer.

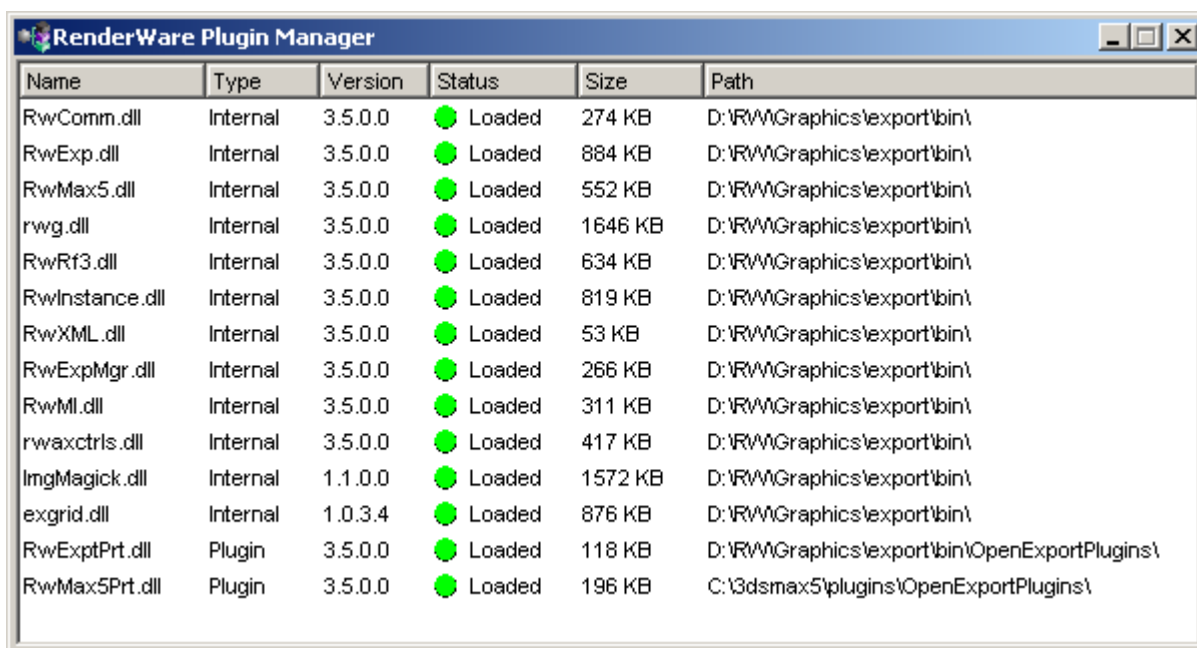
Enable Pre-Instancing controls whether this connection will be used to generate platform specific data when an .rws file is exported with the 'Pre-Instance Data' option enabled. If assets are not being exported to .rws files or the 'Pre-Instance Data' option is not enabled in the Project Templates tab then this setting has no effect.

6.3 Batch Exporter

RenderWare → *Tools* → *Batch Exporter* exports all 3ds max files in a specified directory.

6.4 Plugin Information

The Plugin Manager is available for information only. It contains the location of the .dll files used. The plugins section includes RenderWare Graphics plugins, for example, the Particles plugin which can be bought as an optional extra in the FXPack.




Name	Type	Version	Status	Size	Path
RwComm.dll	Internal	3.5.0.0	Loaded	274 KB	D:\RW\Graphics\export\bin\
RwExp.dll	Internal	3.5.0.0	Loaded	884 KB	D:\RW\Graphics\export\bin\
RwMax5.dll	Internal	3.5.0.0	Loaded	552 KB	D:\RW\Graphics\export\bin\
rwg.dll	Internal	3.5.0.0	Loaded	1646 KB	D:\RW\Graphics\export\bin\
RwRf3.dll	Internal	3.5.0.0	Loaded	634 KB	D:\RW\Graphics\export\bin\
RwInstance.dll	Internal	3.5.0.0	Loaded	819 KB	D:\RW\Graphics\export\bin\
RwXML.dll	Internal	3.5.0.0	Loaded	53 KB	D:\RW\Graphics\export\bin\
RwExpMgr.dll	Internal	3.5.0.0	Loaded	266 KB	D:\RW\Graphics\export\bin\
RwML.dll	Internal	3.5.0.0	Loaded	311 KB	D:\RW\Graphics\export\bin\
rwaxctrls.dll	Internal	3.5.0.0	Loaded	417 KB	D:\RW\Graphics\export\bin\
ImgMagick.dll	Internal	1.1.0.0	Loaded	1572 KB	D:\RW\Graphics\export\bin\
exgrid.dll	Internal	1.0.3.4	Loaded	876 KB	D:\RW\Graphics\export\bin\
RwExptPrt.dll	Plugin	3.5.0.0	Loaded	118 KB	D:\RW\Graphics\export\bin\OpenExportPlugins\
RwMax5Prt.dll	Plugin	3.5.0.0	Loaded	196 KB	C:\3dsmax5\plugins\OpenExportPlugins\

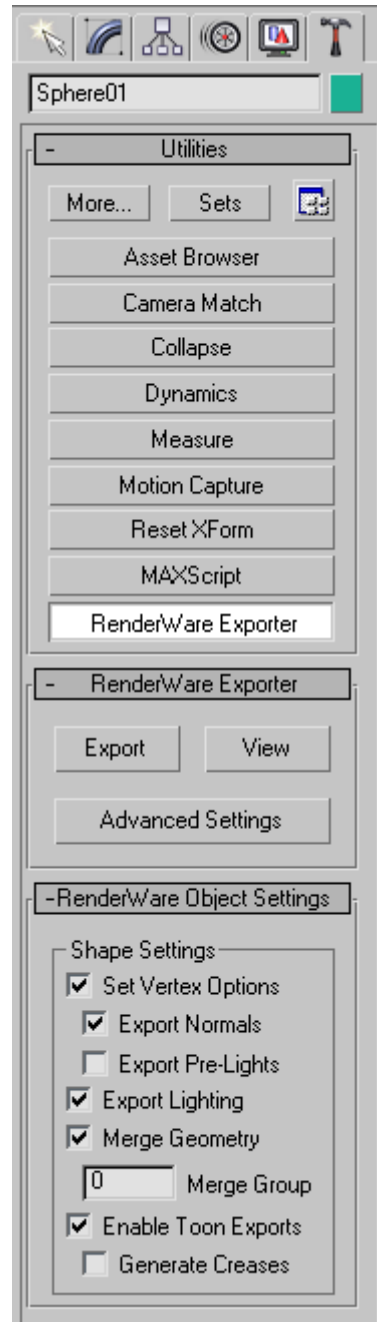
6.5 Utilities Panel

The Utilities panel can be used to access RenderWare Object Settings.

To add a button to the Utilities panel.

1. In the Utilities Panel
2. Click on the Configure Button Sets button 
3. Increase the number of "Total Buttons" by one.
4. Click and drag "RenderWare Exporter" Utility onto the blank button

In 3ds max, RenderWare Object Settings can only be accessed from the Utilities Panel.



RenderWare Exporter

The RenderWare Exporter can be used to:

- Export - equivalent to *RenderWare* → *Export...*
- View - equivalent to *RenderWare* → *View...*
- Advanced Settings - equivalent to *RenderWare* → *Advanced Settings*

RenderWare Object Settings

Set Vertex Options: Enables and disables the *Export Normals* and *Export Pre-Lights* options.

Enable Normals: This option controls whether normals will be extracted from this object when exporting. The Vertex Normals template option must be enabled for this option to have any effect on the final export data. If this option is disabled and the template option is enabled default normals will be stored in the exported data.

Enable Pre-Lights: This option controls whether pre-light colors will be extracted from this object when exporting. The Vertex Prelights template option must be enabled for this option to have any effect on the final export data. If this option is disabled and the template option is enabled default prelight colors will be stored in the exported data.

Export Lighting: This option controls whether the lighting flag on RenderWare Graphics RpGeometry object is set for this node.

Merge Geometry: This option controls whether any geometry created from this node should be merged with other geometry in the scene.

Merge Group: This option is grayed out unless the *Merge Geometry* option is selected. The value is an integer describing the *Merge Group* this node is a member of. The geometry from all nodes with the same *Merge Group* will be combined. For two nodes to be merged they must be in the same Merge Group, have the same lighting flag, both be skinned or not, and both be textured or not.

Enable Toon Data: This option controls whether this object will be toon shaded. The Export Toon Data template option must be enabled for this option to have any effect on the final export data. The option only affects assets exported with the Animated Hierarchy template.

Generate Creases: If this option is not checked and you export a cube, only the silhouette edges are rendered. If this option is checked, each edge that lies between polygons with different smoothing groups is drawn as a crease edge (you must also export vertex normals for the object for this work). The Generate Crease Edges template option must be enabled for this option to have any effect on the final export data and the option only affects assets exported with the Animated Hierarchy template.

Index

3		
3ds max		
Material Editor.....	35	
Version	9	
A		
Animation.....	69	
Controllers.....	69	
Rotation Keyframes.....	70	
TCB	69	
B		
Bézier Patches	18	
BMP	50	
Bump Mapping.....	34, 37	
C		
Character		
Morph Targets	73	
Clipping planes.....	18	
Cropped Textures	53	
E		
Environment Mapping.....	34, 37	
F		
File Naming.....	50	
Filter Modes	51	
H		
Hidden Objects.....	21	
I		
Interpolation	69	
L		
Level of Detail.....	22	
Lights.....	57	
Dynamic	60	
Prelights.....	58	
M		
Masks	50	
Materials.....	40	
Cropping.....	53	
Extra Vertices.....	54	
Filter Modes.....	51	
Hardware Preview	49	
RpMaterial	39	
RwMaterial	39, 52	
Standard Material	31, 52	
Texture Repeats	50	
MatFX.....	40	
Modeling		
Bézier Patches.....	18	
Edit Stacks.....	19	
Normals.....	20	
NURBS	19	
Problems.....	17	
Smoothing.....	20	
Tri-fan	17	
Tri-list	17	
Morphing		
DMorph.....	73	
Morph Targets.....	73	
N		
Normals.....	20	
P		
Patches	18	
PDF format.....	9	
PlayStation 2		
Bone Counts.....	74	
Memory Limits.....	50	
PNG	50	
Potentially Visible Sets	22, 23	
Prelights	58	
PVS.....	22, 23	
R		
RenderWare		
Version.....	9	
RpMaterial	39	
RwMaterial	39	
S		
Smoothing.....	20	
Standard Material.....	31	
T		
Tagging	75	
Texture		
Masks	50	
Repeats.....	50	
Transparency.....	33	
Tri-fan	17	
Tri-list	17	
U		
User Properties.....	75	
UV		
Box Mapping.....	52	
Curved Space Mapping	52	
UV Mapping	52	