Working with WF-3D

A WireFusion plug-in

Version 1.1

1 Requirements

1.1 Requirements

In order to use WF-3D (version 1.1), you need a working installation of 'WireFusion 3' (version 3.1 or higher) on your system (tryout version or registered version).

WF-3D imports 3D models specified in the *Virtual Reality Modeling Language* (VRML97 or VRML2.0) standard. For this you need a third party 3D authoring tool, such as *3ds max*, *Plasma* or *Maya*, which can export to the VRML standard, or any other tool capable of exporting to VRML or creating VRML.

1.2 System requirements

Recommended development system requirements are:

Processor: Intel Pentium III (or compatible) 800 MHz, or higher
Operating system: Microsoft Windows 2000/XP
Internal memory: 256 Mb of available RAM, or more
Hard disk: Minimum of 60 Mb available hard-disk space
Screen resolution: 1280×1024 pixel
Sound: Sound card for multimedia presentations

2 Installation

The installation process of WF-3D is very easy.

1. First make sure to have WireFusion installed and working on your system.
2. Unzip '*wf-3d.zip*' (which includes the plug-in file *wf-3d.wpl*) to an empty location on your hard disk
3. In WireFusion, choose *File* > *Install Plug-in...* (Figure 1)
4. Browse for the WF-3D plug-in, called '*wf-3d.wpl*'. Click Open (Figure 2)
5. A dialog opens; choose either *Tryout Mode* or *Register...*
6. Restart WireFusion

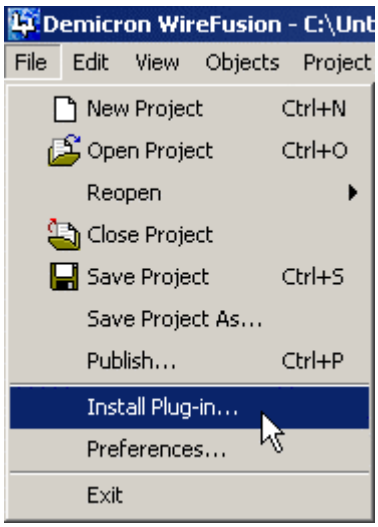

Figure 1. Installing the WF-3D plug-in

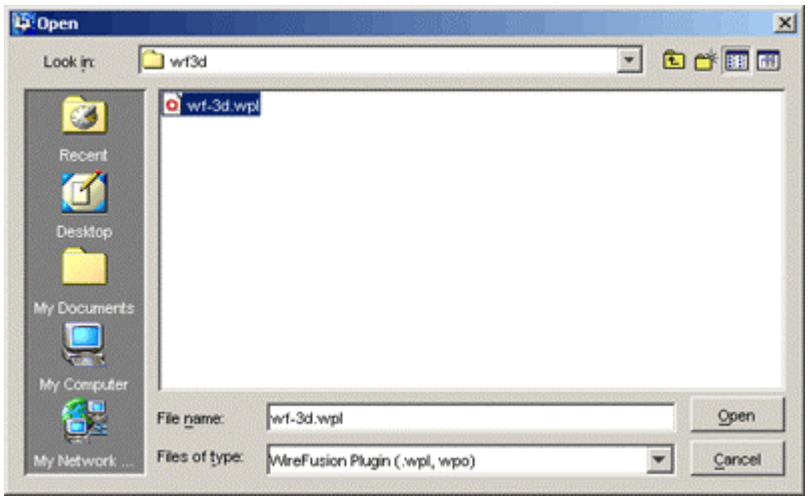

Figure 2. Opening wf-3d.wpl

3 Specifications and features

3.1 Features

The 3D player (3D render engine) included with WF-3D is only about 30kb in size and includes the following rendering features:

- **Shading**

  - **32-bit color depth rendering** - ensures high image quality
  - **Multiple colored dynamic light sources** - Two types of light sources are handled; Directional and Omni.
  - **Constant shading** - No lighting is performed.
  - **Flat shading** - Calculates the lighting for each facet and gives it a constant color.
  - **Gouraud shading** - Calculates the lighting for each polygon vertex. The resulting colors are interpolated over the polygon surface to give it a smooth appearance.
  - **Phong shading** - Performs shading calculations for each screen pixel. This shading correctly shows highlights on glossy objects and generally gives the object a smoother look compared to the Gouraud shading.
  - **Lightmap** - By mapping a texture with a view of the virtual surrounding onto 3D objects, so that the texture doesn't move with the object but with the changes in the view, you simulate the reflections of the surrounding from the objects. This can be used to increase the metallic feel of glossy objects, and also provides a way of lighting the objects without specifying light sources. The following features are supported for lightmaps:
    - **Bilinear filtering** - Interpolates between the bitmap pixels to create a smoother non-pixelated view of close-ups of the seen reflections. This way you can use smaller textures and still retain high quality of the reflections.
    - **8-bit or 24 bit high-resolution bitmaps** - Use GIF, JPEG or PNG images with any width and height.
    - **Dynamic lightmaps** - Replace a static map with an animated lightmap, using the `Texture` or `Texture Array` objects.
    - **Per-object specification of lightmap** - Every object can have its own lightmap specified, which allows you to make sure every object gets the right material feeling.
    - **Combine standard texture mapping with lightmap** -The lightmaps are added on top of the standard textures.
  - **Default shading and lightmap** - A default shading and a default lightmap is applied to the scene. But individual shading methods and individual lightmaps can be specified for each individual object.

- **True 32-bit z-buffer** - The z-buffer method is a way to ensures that only the closest objects are displayed by preventing background objects from overwriting foreground objects on the screen. 32 bits precision ensures that virtually all scenes are correctly displayed.

- **Full scene anti-aliasing** - Aliasing is caused by the sampling of smooth data onto a screen consisting of discrete pixels. The result is the visible stair stepping or jaggies at the edges of the object polygons. Anti-aliasing is the method to remove this and results in smoother edges and sharper images. Two modes of anti-aliasing are supported; *On/Off* and *Auto* (automatic switching between On and Off).

- **Edge anti-aliasing** - Results in smoother edges, even when moving or rotating objects.

- **Transparency/opacity**

  - **256 levels of transparency**
  - **Correct rendering of overlapping transparent surfaces** - When two transparent surfaces are overlapping they will be correctly combined on the screen. This means transparent objects like glass cups are correctly displayed.

- **Backface culling** - A way of making sure that parts of the object that face away from the observer is not processed, which speeds up the rendering.

- **3D clipping** - Clips objects against the 3D view *frustum*, which is the pyramid-shaped view created by looking at the 3D world 'through' a screen. This means any detail not in view, including detail virtually seated in front of the screen and behind a certain far distance, is removed. Most importantly, it also makes sure that detail halfway inside and halfway outside the view will be correctly displayed.

- **Texture features**

  - **UV texture mapping** - Polygon vertices are assigned coordinates in the texture, which in turn will be interpolated over the surface to assign a texture value to the screen pixel. This is the most flexible way of mapping.
  - **Bilinear filtering** - Assigns to the screen pixel the interpolated value between the closest associated texture pixels. This ensures a smoother non-pixelated view of close-ups of the textures on the 3D objects.
  - **Texture opacity** - Defines how much the texture will contribute to the color of the object.
  - **Advanced textures** - Replace a texture with a WireFusion Texture, which allows animated textures or textures with built-in logic and interactivity.
  - **Perspective correction** - Perspective correction increases realism by taking into account the depth of an object when applying the texture map, which removes the distortion that appears when a texture is applied to a 3D object. This provides the appearance that texture details near the viewer are larger than detail found further away, and ensures that parallel lines such as railroad tracks converge in the far distance.
  - **Sub-pixel precision** - Sub-pixel precision of the texture mapping ensures the texture is more correctly placed on a 3D object, which is noticeable when changing the view by e.g. rotating the object.
  - **8-bit or 24 bit high-resolution bitmaps** - Use GIF, JPEG or PNG images with any width and height.
  - **8-bit and animated alpha channels** – Shape textures by using transparent GIF or PNG textures or use the `Texture` object's alpha channel scene to create animated 8 bit alpha channel maps.

- **3D object material** - Specify following material properties

  - Ambient intensity
  - Diffuse color
  - Specular color
  - Emissive color
  - Glossiness
  - Opacity
  - Reflection maps

- **Animation features**

  - **Full control over the animation** - Animations can be time driven, but there is also the possibility to fully control the fractional progress of the animation.
  - **Matrix animations** - Animation of the position, rotation and scale of objects.
  - **Vertex animations** - Animation of each vertex position (morphing)
  - **Dynamic normal** - When an object is morphed, using vertex animation, recalculated vertex normals (dynamic normals) make sure the lighting is correctly displayed.
  - **Material properties animation** - Specify animations for any material property, e.g. opacity, object color, glossiness etc.
  - **Light animation** - Specify animations for light source direction, position, color and intensity.
  - **Camera animation** - Specify animations to move the camera views.
  - **Hierarchal animation control** - When an animation is specified on an object consisting of a hierarchal object structure the animations are separately controllable for each sub-object.

- **User interaction features**

  - **5 different modes of user interaction** - You can allow the user to interact in any of the modes Zoom, Pan, Rotate, Interact and Normal.
  - **Keyboard-controlled navigation** - Specify hotkeys for Zoom, Pan and Rotation.
  - **True 6 DOF (degrees of freedom)** - No restrictions as to where and how you translate camera positions and object positions.
  - **Camera restrictions** - Optional camera rotation restrictions, and zoom-in and zoom-out restrictions.
  - **Two camera rotation modes** – Choose to rotate cameras around the world axis or the local camera axis.
  - **Multiple cameras** - Specify multiple cameras and let the user switch between their viewpoints.

- **Streaming features**

  - **Object streaming** – Have 3D objects streamed into the presentation.
  - **Texture streaming** - You can choose to let the textures start streaming after the objects have finished loading or start streaming when its associated object is loading.
  - **Animation streaming** - Have object animations streamed into the presentation

- **Streaming order** – The order of streamed objects and animations are fully controlled.

- **Encryption** - Encrypts the 3D model to protect from unauthorized alterations.

- **Compression features** - All 3D data are compressed, generally achieving 90-95% file size compression compared to the VRML-file format.

  - Compresses mesh data
  - Compresses texture coordinate data
  - Compresses vertex animation data

- **3D file format/modeling program support** - Supports the VRML 97/2.0 file format, which allows you to work in all 3D authoring tools with the capability of exporting to the VRML 97/2.0 format. Mainstream content creation tools such as *3ds ma, Plasma* and *Maya* support VRML 97/2.0. The exporters in *3ds max, Plasma* and *Maya* includes the following data:

  - Mesh information
  - Material settings
  - Textures
  - Color vertices
  - Animations (matrix- and vertex animations)
  - Lights
  - Cameras

3.2 VRML support

The VRML 97/2.0 support in WF-3D is not complete. The nodes supported are a subset of the nodes in VRML97/2.0.

The following VRML 97/2.0 nodes are supported:

Anchor	Appearance	Color	ColorInterpolator
Coordinate	CoordinateInterpolator	DirectionalLight	Group
ImageTexture	IndexedFaceSet	LOD	Material
NavigationInfo	Normal	NormalInterpolator	OrientationInterpolator
PixelTexture	PointLight	PointSet	PositionInterpolator
ScalarInterpolator	Shape	Switch	TextureCoordinate
TextureTransform	TimeSensor	TouchSensor	Transform
Viewpoint			

The VRML 97/2.0 ROUTE formalism is supported. All ROUTEs between two supported nodes are handled.

Nodes that are not supported:

AudioClip	Background	Billboard	Box
Collision	Cone	Cylinder	CylinderSensor
ElevationGrid	Extrusion	Fog	FontStyle
IndexedLineSet	Inline	MovieTexture	PlaneSensor
ProximitySensor	Script	Sound	Sphere
SphereSensor	Spotlight	Text	VisilibitySensor

The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, intranets, and on local client systems. A very brief description of how VRML works would be to say that the format consists of a set of nodes, where the nodes are the fundamental components of the VRML file. Nodes are abstractions of various real-world objects and concepts. Examples include spheres, lights, and material descriptions. Nodes contain fields and events. Messages may be sent between nodes along routes.

Note: You need no knowledge about VRML in order to use or create 3D contents for WF-3D, you only need to know how to export to VRML from your authoring tool. For those who are interested to know more about VRML, the whole VRML97 standard can be found at http://www.vrml.org/technicalinfo/specifications/vrml97/index.htm.

4 Introduction

WF-3D is an add-on (plug-in) to WireFusion, which extends it with advanced web-3D capabilities.

In conjunction with WireFusion, WF-3D gives you a web-3D solution more powerful than anything else on the market today. You get one of the smallest 3D renders (about 30kb) available and one of the most advanced, both when talking about quality, features and performance. You also get a renderer with enormous capabilities and possibilities. And as the technology is based on Java, you will also reach the major part of all Internet users worldwide without the need of a browser plug-in, no matter which platform or browser they are using. WF-3D is also one of the most affordable solutions available today, especially when taking into account the time it takes to develop a complex web-3D solution. With its unique programming interface, nothing beats WireFusion and WF-3D today when it comes to development time! With one license you can create and publish as many presentations you like. There are no site licenses or annual fees.

Note: WF-3D does not use Sun Microsystems Java3D. WF-3D is an in-house developed technology, which requires no 3D hardware acceleration cards or additional software drivers.

One of the great benefits with WF-3D is the capability to have interactive and animated textures. You can even load a previously saved project and apply it as a texture on any texture. This is very useful when, for example, creating interactive product simulations. You could create product simulations in 2D, e.g. a mobile phone with full logic and interactivity. Then you could add the display, from the 2D phone project, with all the logic as a texture on your 3D phone display, and in a snap you would have a “fully” working 3D phone.

WF-3D is delivered to you as a fully working tryout version, with no time limitations or feature limitations. You can even evaluate it together with a tryout version of WireFusion. Projects created with the tryout version can later be loaded into a registered version, all for your convenience.

All presentations made with a tryout version will have a time delay of five seconds before starting. They also have an icon with a link to the WireFusion web page and a text in the HTML code saying "*Tryout Version - For evaluation purposes only*". These restrictions are removed when your register the software.

Three powerful objects are included in WF-3D, and they are installed in a tab named 3D (Figure 3):

- `3D Scene`: Imports 3D models and helps to configure and adjust them to your project with an easy GUI.
- `Texture`: Replaces static textures on your 3D models with interactive and animated textures.
- `Texture Array`: Animates textures and lightmaps.

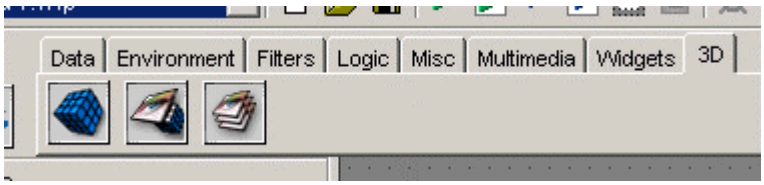

Figure 3. 3D Scene, Texture and TextureArray

Please send comments and feedback regarding this manual or the software to *development@demicron.com*

5 Creation and export of 3D models

5.1 General creating

WF-3D imports 3D models specified in the *Virtual Reality Modeling Language* (VRML97 or VRML2.0) standard, which can be exported from the major part of all 3D authoring tools on the market today, including *3ds max*, *Plasma, 3D Studio VIZ*, *Maya*, *Lightwave*, *trueSpace*, *Amapi*, *Carrara*, *3DMeNow* and many more. All major 3D scanners also have the possibility to export to VRML and so does 2D-to-3D conversion software like *Realviz ImageModeler*.

Even though there is no preset limit in WF-3D of how large your scene can be or how many polygons you can use in your models, you should always try to make it as simple as possible when creating it in your 3D authoring tool. By reducing the number of polygons, the final rendering performance increases and the file size get smaller. This will increase the user experience, so try to find a good balance between size and quality while creating the models. There is a *crease angle* (smoothing) method in WF-3D, which helps to improve the smoothness of objects and hence removes the need of high polygon models. WF-3D also supports *LOD* (Level of Detail), which can be used to construct an object that alters its geometric complexity, or level of detail based on how close the camera is to the object.

Further, you do not create 3D animations inside WireFusion, you create them directly in your 3D authoring tool. You can make object mesh animations (morphing) and you can animate cameras and light sources, everything is included with the VRML export and supported by WF-3D. Then you set the animation time, program the animation events and create touchsensors from inside WireFusion and WF-3D. There is hence no need to use the *VRML Helpers*, which are included in, for example, *3ds max* for the creation process of animations.

The philosophy is to let you stay in your favorite 3D authoring tool as long as possible; to create models and animations there. Then you import the models to WF-3D and WireFusion for final tuning and to add interactivity and logic before publishing to the web.

Note: This manual uses *3ds max* to demonstrate different processes.

Note: Consult the user manual for your specific 3D authoring tool for information regarding model creation and VRML export.

5.2 The VRML Export

Everything you create in your 3D authoring tool might not come along with the export (depending on the VRML exporter or the VRML standard) or isn't supported by WF-3D. Below you will find information about features supported by both VRML97 and WF-3D, which will be exported:

Mesh: Mesh information for objects. VRML primitives are not supported by WF-3D though.

Material properties: Ambient intensity, Diffuse color, Specular color (is a combination of *Specular Color* and *Specular Level* from *3ds max*), Emissive color (called *Self Illumination Color* in *3ds max*), Glossiness and Opacity

Textures: Object bitmaps specified in the JPEG, GIF or PNG format. It is recommended to optimize the texture size and quality before exporting.

Animation: Matrix (translation) animations for objects, cameras and lights and vertex (mesh) animations for objects.

Cameras: Camera positions, directions and perspectives. *Target Camera* or *Free Camera* made in *3ds max* are supported.

Lights: Light positions, directions, intensities and colors. *Target Direct* or *Free Direct* for directional lights and *Omni* for point lights made in *3ds max* are supported.

Then there are additional settings supported by WF-3D, which aren't included with the export, e.g. *Texture opacity* and *Lightmaps*. You will learn more about them later.

5.3 The exporting procedure

Once your model and your animations are created in your 3D authoring tool, then it is time to export to VRML. The procedure of exporting is quite straightforward and is explained by the following example (made in *3ds max*).

Example

We want to create a box with a texture on it and then export the box to VRML97.

Step 1

Create a symmetric box and position it in the world center (Figure 4).

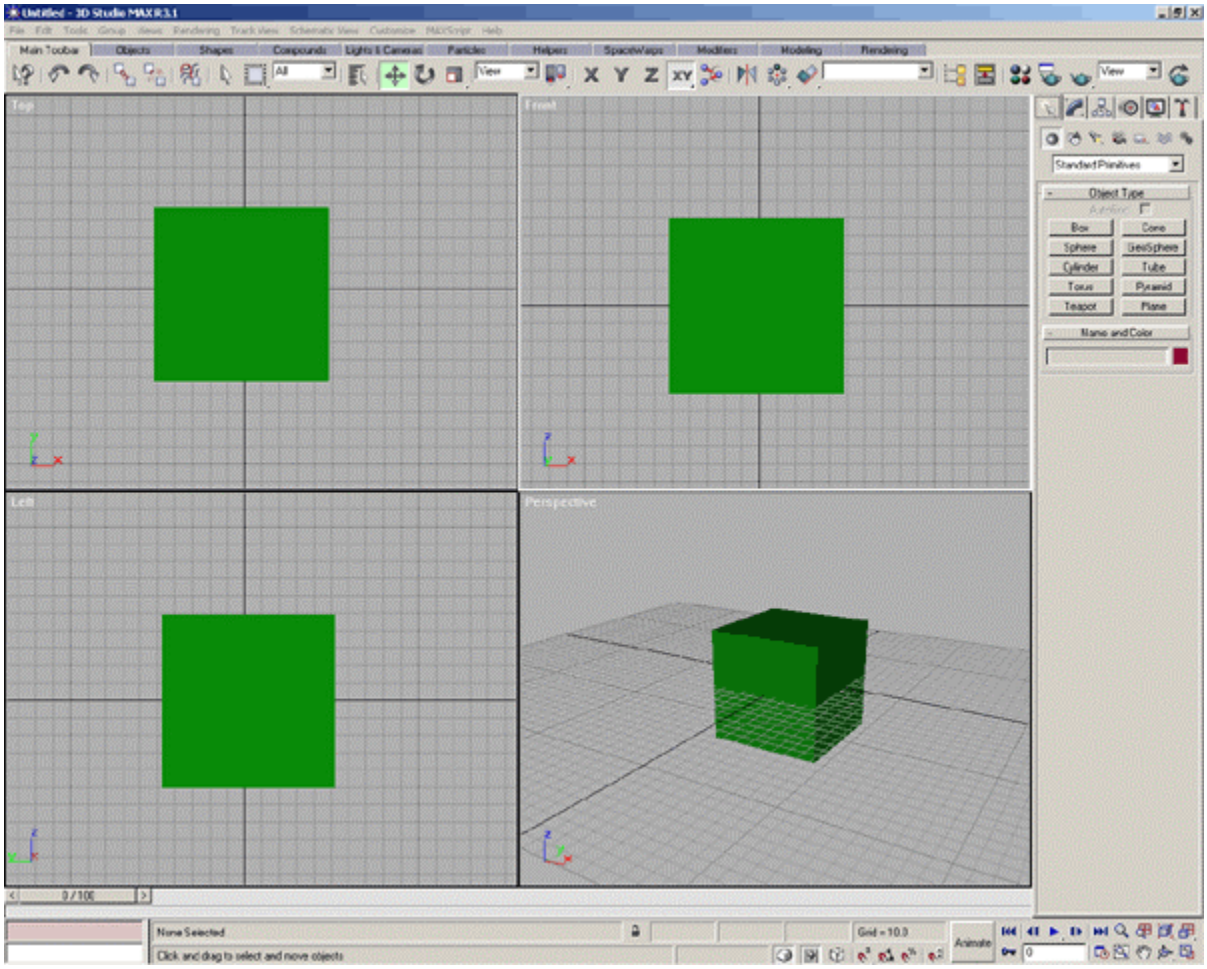

Figure 4. A box placed in the world center

Step 2

Insert a *Target Camera* and position it similar to Figure 5.

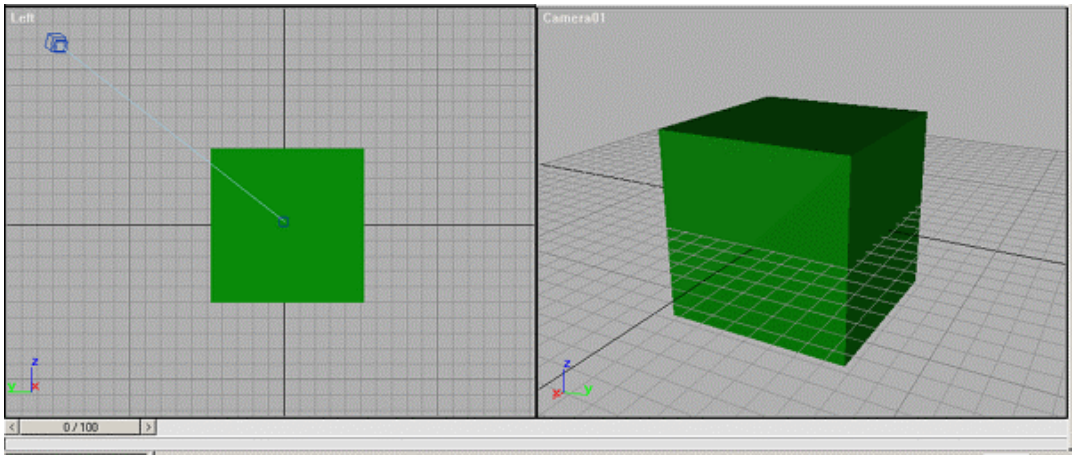

Figure 5. A Target Camera inserted

Step 3

Add a texture to the box (Figure 6).

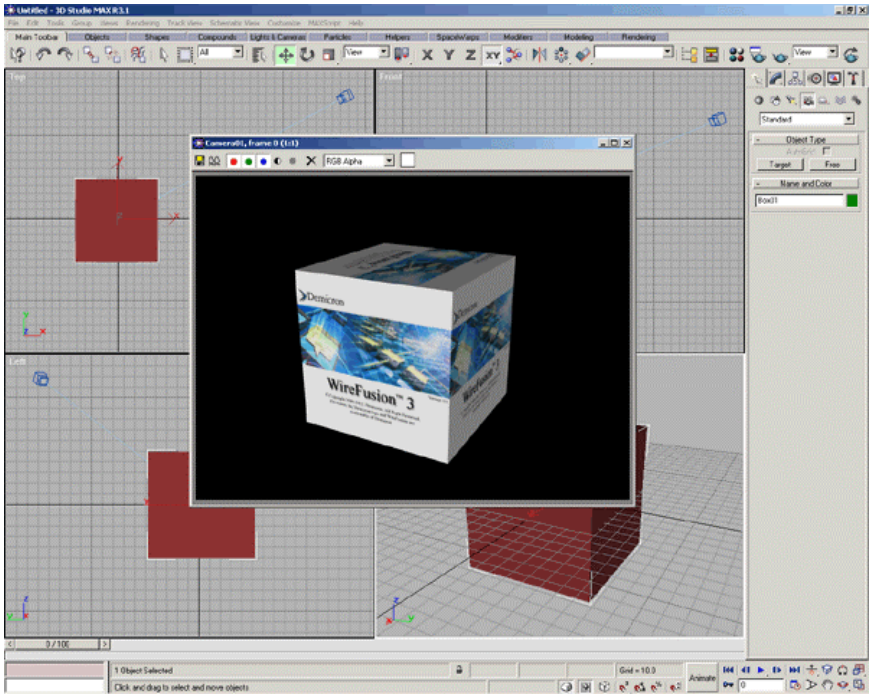

Figure 6. A texture added to the box

Step 4

To export, choose *File* > *Export...* Name the file and save it as type *VRML97 (*.WRL)*. Click the Save button (Figure 7).

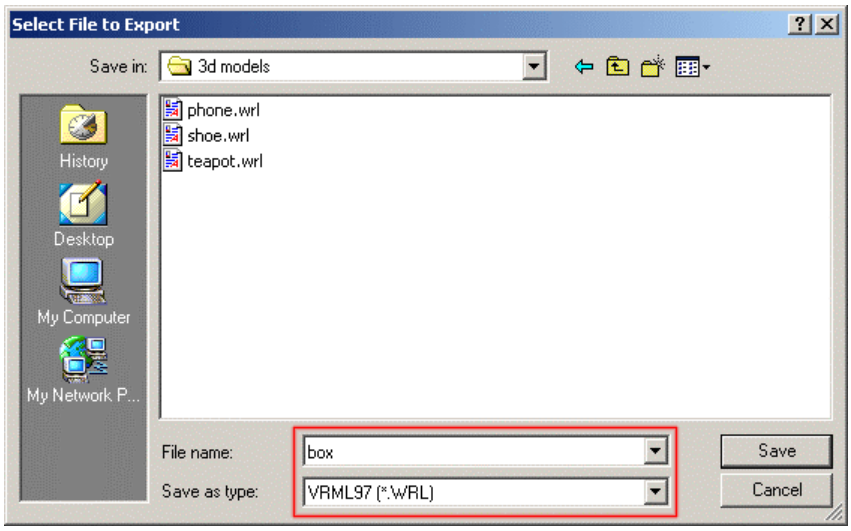

Figure 7. Exporting the box to VRML97

Step 5

After you have clicked the Save button, the *3ds max VRML97 Exporter* dialog appears. Here you can choose some settings for the VRML export, but as a rule you should leave all the checkboxes unchecked most of the times (Figure 8). Click OK and your model is ready to be imported into WF-3D. Make sure to have your textures in the same folder as your VRML files when importing to WF-3D.

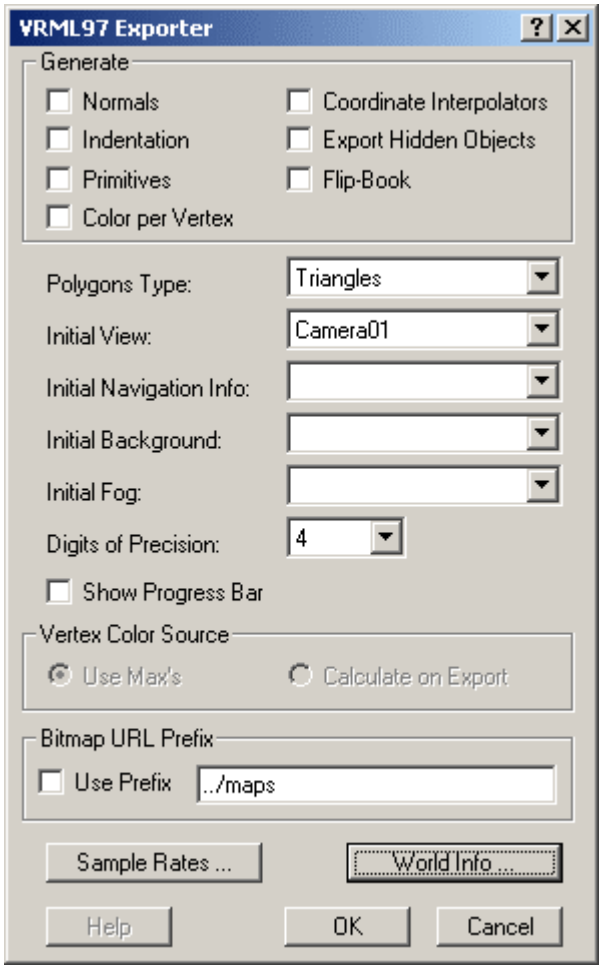

Figure 8. The VRML97 Exporter dialog

In the *3ds max VRML97 Exporter*, you might want to mark the following checkbox:

Coordinate Interpolators: Exports animation effects that involve actual modifications of the mesh objects, and not just move, rotate, and scale. This option can generate large files, because the exporter has to calculate the position of every vertex for this kind of animation. Mark this checkbox only when you have created mesh animations.

In the *3ds max VRML97 Exporter*, you especially **don't** want to mark the following checkboxes:

Normals: Generates real normals for objects. This option generates unnecessarily large files, as the normals are calculated in WF-3D.

Primitives: Exports to VRML primitives. It specifies for example a sphere with its radius. This is not supported in WF-3D.

The above settings are normally the only ones you have to think of when exporting from *3ds max*. You can read about the other choices in the *3ds max* help.

Note: All VRML exporters don't have the same nomenclature as *3ds max*, but the above information should give you some guidance.

6 3D Scene object

The `3D Scene` is the core object of WF-3D. It's the object that imports your VRML file and displays the 3D model, and also where you make all the 3D settings.

6.1 Importing VRML

When you are ready in your 3D authoring tool and have exported to VRML, then it's time to import your 3D model to WF-3D.

There is a slight difference between the `3D Scene` and the other objects in WireFusion. When you drop the `3D Scene` in the *Script Area* a loading dialog appears and you are supposed to browse for your 3D model (VRML file). After loading the VRML file, the *Properties* dialog will open. We illustrate this with an example.

Example

We want to import the 3D box, exported from *3ds max* (see above).

Step 1

In WireFusion, insert a `3D Scene` object into your project, choose *Objects* > *3D* > *3D Scene*

When the loading dialog opens, browse for your 3D model (Figure 9) and click Open to load it. Make sure to have any textures placed in the same folder as the VRML file in order to have them loaded properly. Supported VRML extensions are: *wrl*, *wrz* (compressed) and *wrl.gz* (compressed).

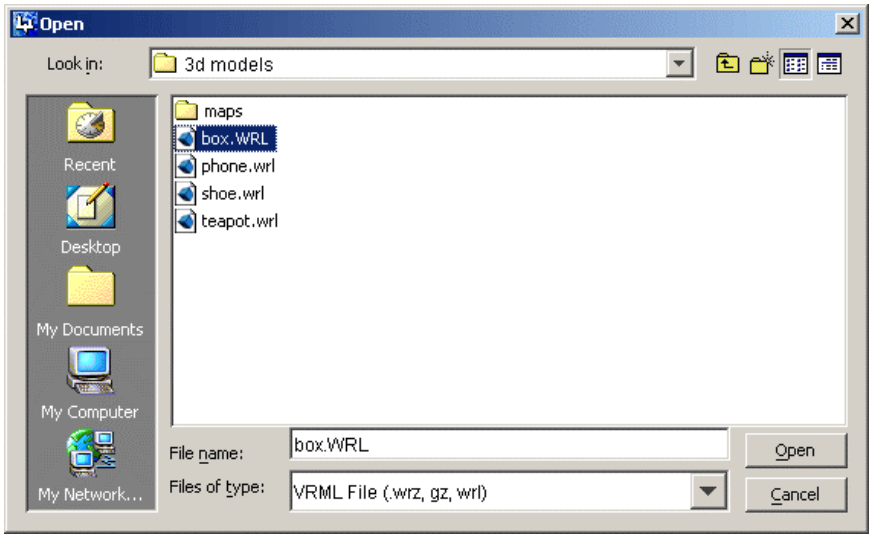

Figure 9. `3D Scene` loading dialog

Step 2

The `3D Scene` *Properties* dialog opens and the model is imported (Figure 10).

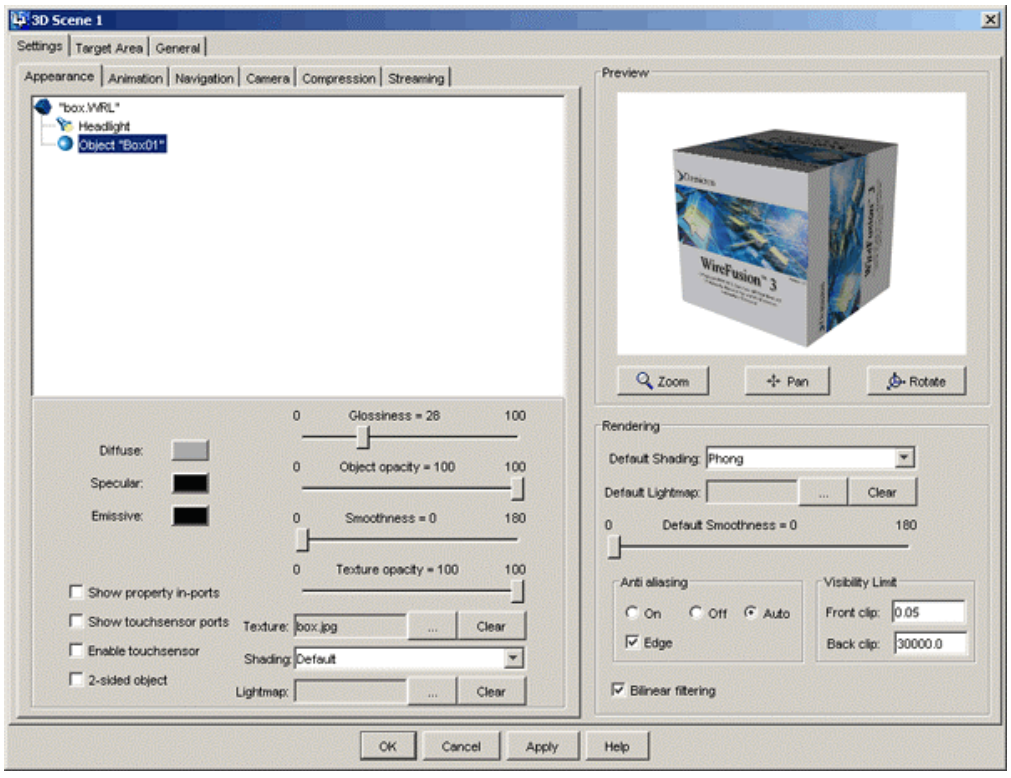

Figure 10. `3D Scene` *Properties* dialog

Note: When you have imported a model into the `3D Scene` object, then you can't reload an updated version of the model again. You have to insert a new `3D Scene` and load the model again. Therefore, make sure that you are done with your modeling and creation of animations in your 3D authoring tool before doing too much work in WF-3D.

6.2 The 3D Scene user interface

The `3D Scene` differ quite a lot from the other WireFusion objects with its extensive user interface. It is almost like a small program. Nevertheless, everything you need in order to configure the 3D model is placed in the *Settings* tab and its six sub-tabs. *Target Area* settings are, as usual, found in the *Target Area* tab (Figure 11) and general information about the object and the comments text field are found in the *General* tab.

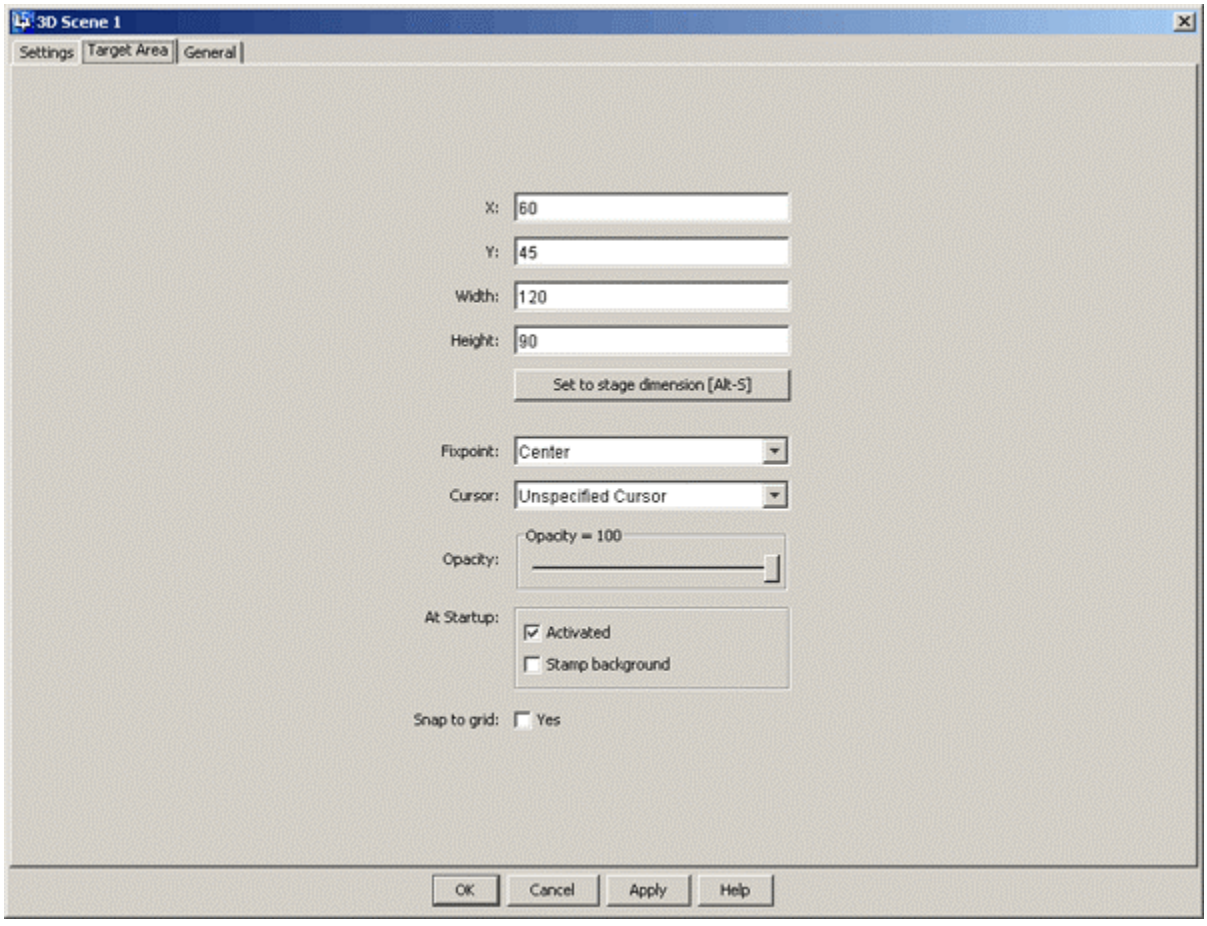

Figure 11. The `3D Scene` Target Area tab

6.2.1 Preview and Rendering settings

All the sub-tabs have the same right-hand side, a *Preview* window and some general *Rendering* settings for your models (Figure 12).

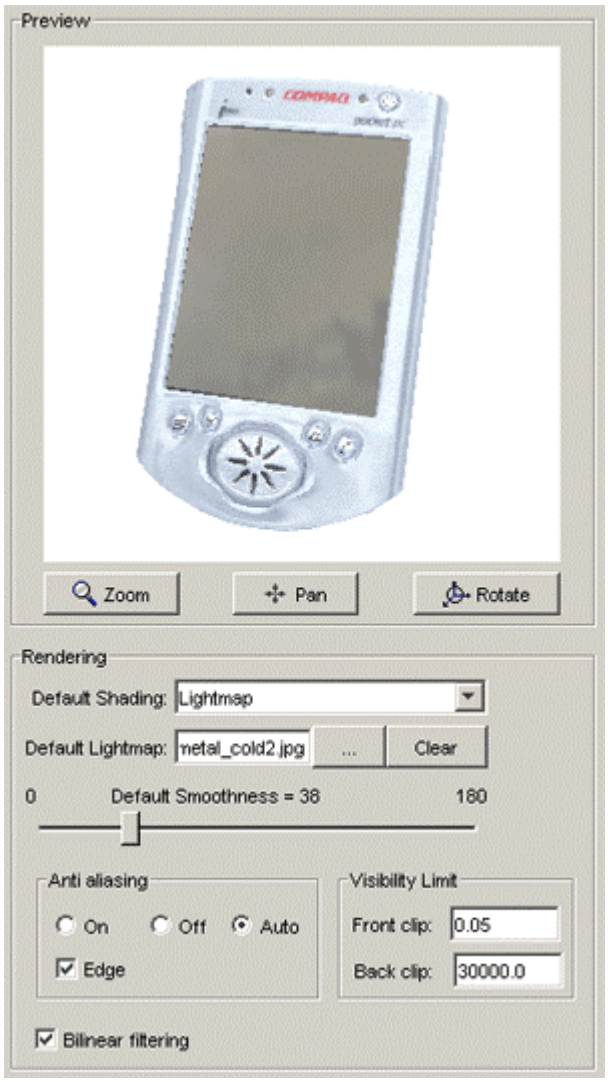

Figure 12. The *Preview* window and some general settings

Preview

You can use the *Preview* window to navigate your model. By default the *Rotate* mode is activated. To switch navigation mode, click the buttons *Zoom*, *Pan* or *Rotate*. Use the mouse to navigate (left-click and drag).

The *Preview* window has the same proportion as the *Target Area* for the `3D Scene` (default is 4:3). To have a quadratic *Preview* window, make the *Target Area* quadratic. Go to the *Target Area* tab and change the width and height values. To have the preview window updated, press the Apply button.

Zoom: (button)
Click to change the preview navigation mode to zoom.

Pan: (button)
Click to change the preview navigation mode to pan.

Rotate: (button)
Click to change the preview navigation mode to rotate.

Note: These modes are only for the *Preview* navigation window and will not set any default navigation for the final presentation. See *Navigation* tab for more information about navigation.

Rendering

In the *Rendering* section you set some general settings for your model, you choose for example the shading method and whether to use anti-alising and bilinear filtering or not.

Default Shading: (choice)
There are five default shading options; *Constant*, *Flat*, *Gouraud*, *Phong* and *Lightmap*. All objects in your model will have this shading by default. However, you have the option to set an individual shading method for each individual object in your model (see the Appearance/Objects section).

Constant: (alternative)
No lighting is performed. Useful with texture-mapped surfaces when the texture image is preferred to reproduce on the object as close to the original color as possible.

Figure 13. Constant shading

Flat: (alternative)
Calculates the lighting for each facet and gives it a constant color. No smoothing information is displayed. Use this mode to provide a faceted look to your geometry.

Figure 14. Flat shading

Gouraud: (alternative)
Calculates the lighting for each polygon vertex. The resulting colors are interpolated over the polygon surface to give it a smooth appearance. Gouraud shading removes the faceted look of objects but highlights on glossy objects may be deformed and bright or dark intensity streaks, called Mach bands, may appear on the surface.

Figure 15. Gouraud shading

Phong: (alternative)
Performs shading calculations for each screen pixel. To get the normal used for the shading calculation you interpolate the vertex normals over the polygon surface. This shading correctly shows highlights on glossy objects and generally gives the object a smoother look compared to Gouraud shading.

Figure 16. Phong shading

Lightmap: (alternative)
By mapping a texture, showing the surrounding of a 3D scene, onto 3D objects so it doesn't move with the object, but with the changes in the view, you simulate reflections of the surrounding on the objects. This can, for example, be used to increase the metallic feel of objects, and can also be used as a way of lighting the objects without specifying light sources. If chosen, all other light sources are deactivated.

Figure 17. Lightmap shading

The above shading settings can be programmed to alter dynamically in your presentation by using the *In-ports* > *Rendering* > *Shading* (Figure 18).

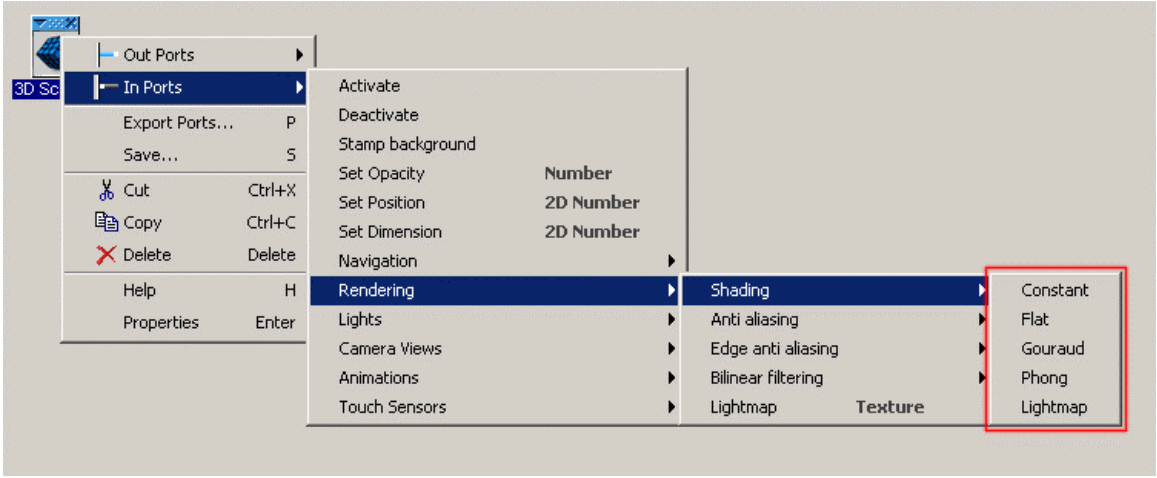

Figure 18. Shading in-ports, found in all `3D Scene` objects

Default Lightmap: (loading dialog)
"..." loads a default lightmap, and "Clear" removes the lightmap image. Applicable if the *Lightmap* shading method is selected in the *Default Shading* menu (see above). Supports JPEG, GIF or PNG format with any width and height.

Note: A loaded lightmap will be included in the published presentation, even if it's not used. So make sure to remove it (Clear) if you don't use it.

Tip: A lightmap is usually static, but you can have animated lightmaps, or you can replace a lightmaps, during a presentation by using the `Texture` and `Texture Array` object. The new lightmap information is send from the texture object to the `3D Scene` *In-ports* > *Rendering* > *Lightmap* (Figure 19).

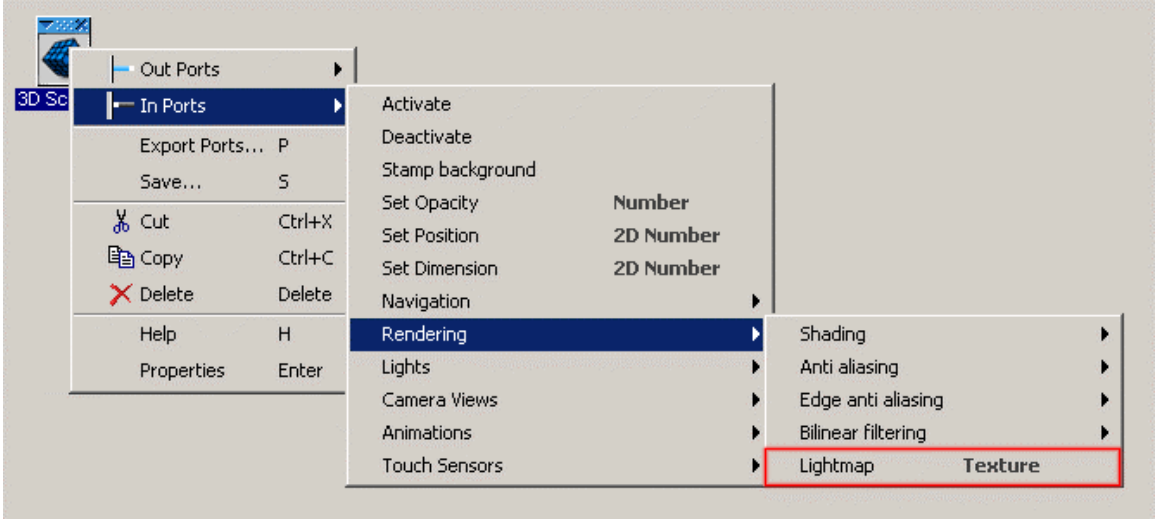

Figure 19. Lightmap in-port, found in all `3D Scene` objects

Default smoothness: (slider)
The *Default smoothness* option sets the crease angle (between 0 and 180 degrees) for all objects in your model. However, individual crease angles can be set for each individual objects (see below).

Figure 20. Comparison between different smoothness values

Anti aliasing: (group)
Aliasing is caused by the sampling of smooth data onto a screen consisting of discrete pixels. The result is the visible stair-stepping or jaggies at the edges of the object polygons. Anti-aliasing is the method to remove this and results in smoother edges and sharper images. Three modes of anti-aliasing is supported, always On, always Off and Auto (automatic switching between On and Off). Since anti-aliasing is a processor-intensive operation the Auto option is preferred, where the method is temporarily turned off when running animations, interacting with the model or if a texture animation is running.

On: (radiobutton)
Full-scene anti-aliasing is always ON, even if an animation is running, if the 3D model is moved or if parameters are changed.

Off: (radiobutton)
Full-scene anti-aliasing is always OFF.

Auto: (radiobutton)
Full-scene anti-aliasing is automatically turned OFF when the 3D model is moved, or an animation is running, or parameters are changed (i.e. animations

through in-ports, like changing the opacity dynamically for example), else it is turned ON.

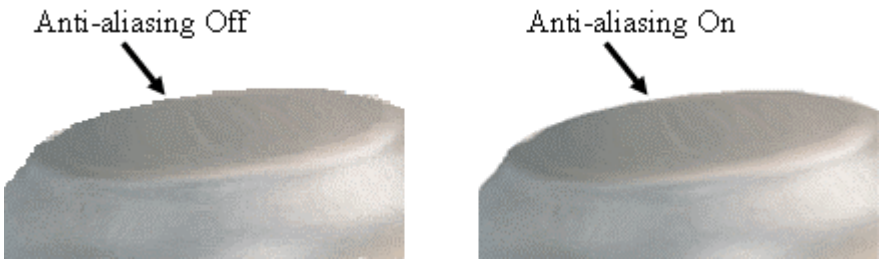

Figure 21. Comparison between anti-aliasing Off and On

The above anti-aliasing settings can be programmed to alter dynamically in your presentation by using the *In-ports* > *Rendering* > *Anti-aliasing* (Figure 19).

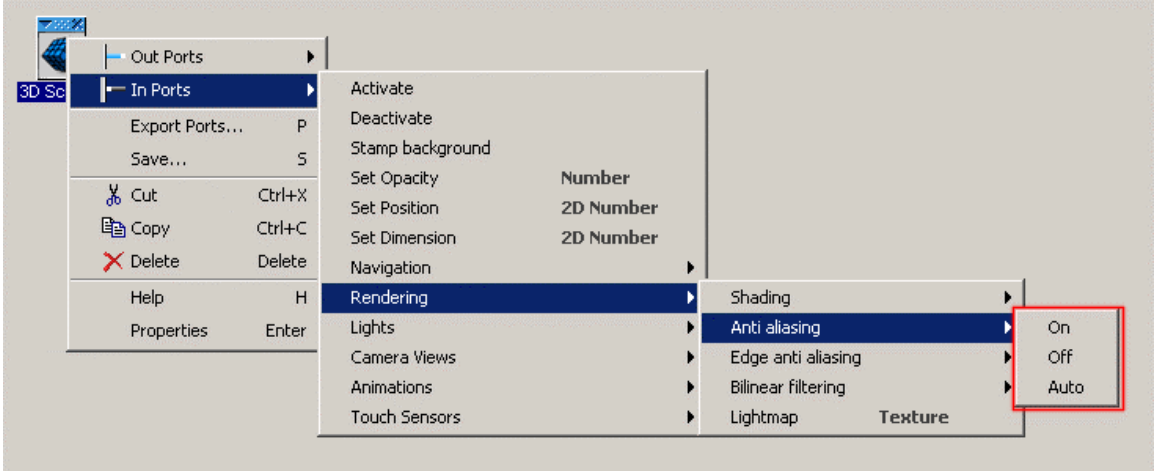

Figure 22. Anti-aliasing in-ports, found in all `3D Scene` objects

Edge anti-aliasing: (group)
As described above, aliasing is caused by the sampling of smooth data onto a screen consisting of discrete pixels. The result is the visible stair-stepping or jaggies at the edges of the object polygons. Anti-aliasing is the method to remove this and results in smoother edges and sharper images. However, the full-scene anti-aliasing consumes a lot of CPU and cannot be used in realtime (unless a very fast computer is used). The edge anti-alising method can however be used to get smoother edges, and hence better quality, even in realtime, i.e. when moving your models or having animated textures.

On: (radiobutton)
Edge anti-aliasing is always ON, even if an object or texture animation is running.

Off: (radiobutton)
Edge anti-aliasing is always OFF.

The above edge anti-aliasing settings can be programmed to alter dynamically in your presentation by using the *In-ports* > *Rendering* > *Edge anti-aliasing* (Figure 23).

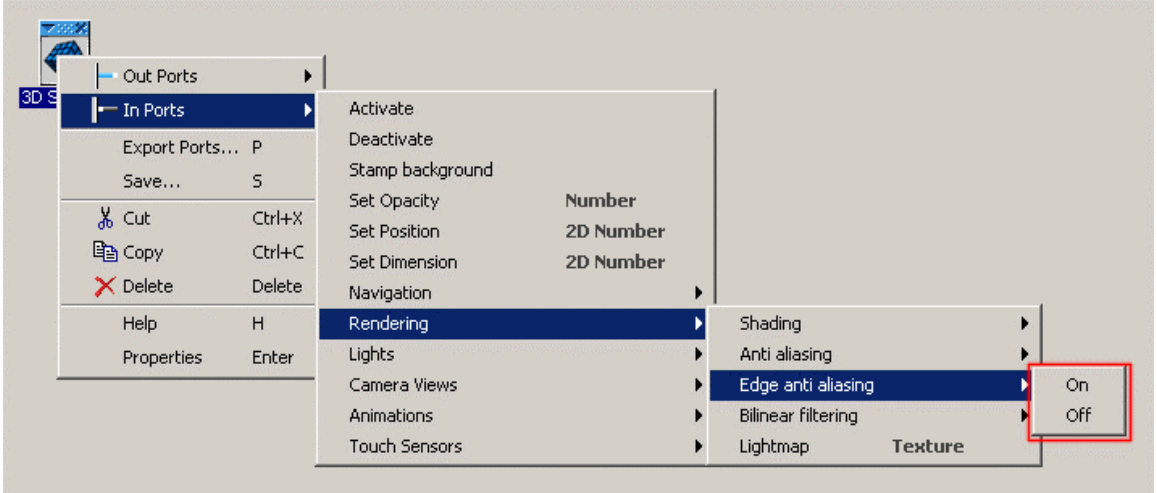

Figure 23. Edge anti-aliasing in-ports, found in all `3D Scene` objects

Bilinear filtering: (ckeckbox)
Mark this checkbox to activate bilinear filtering (smoothing) for all textures and lightmap images in the scene.

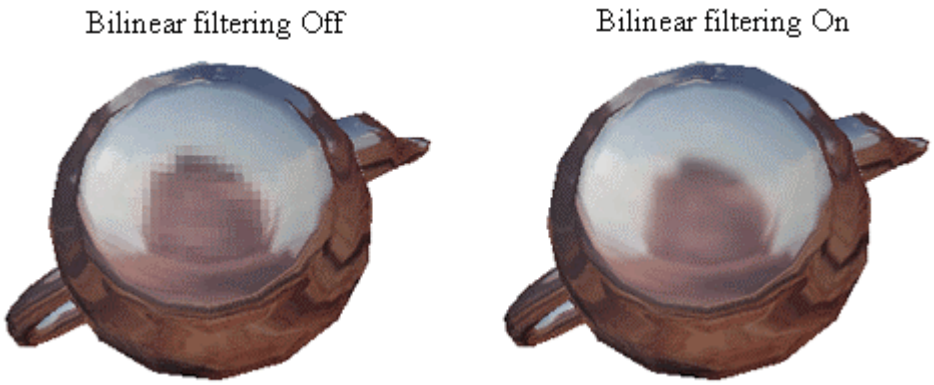

Figure 24. Comparison between bilinear filtering Off and On

The above bilinear filtering settings can be programmed to alter dynamically in your presentation by using the *In-ports* > *Rendering* > *Bilinear filtering* (Figure 25).

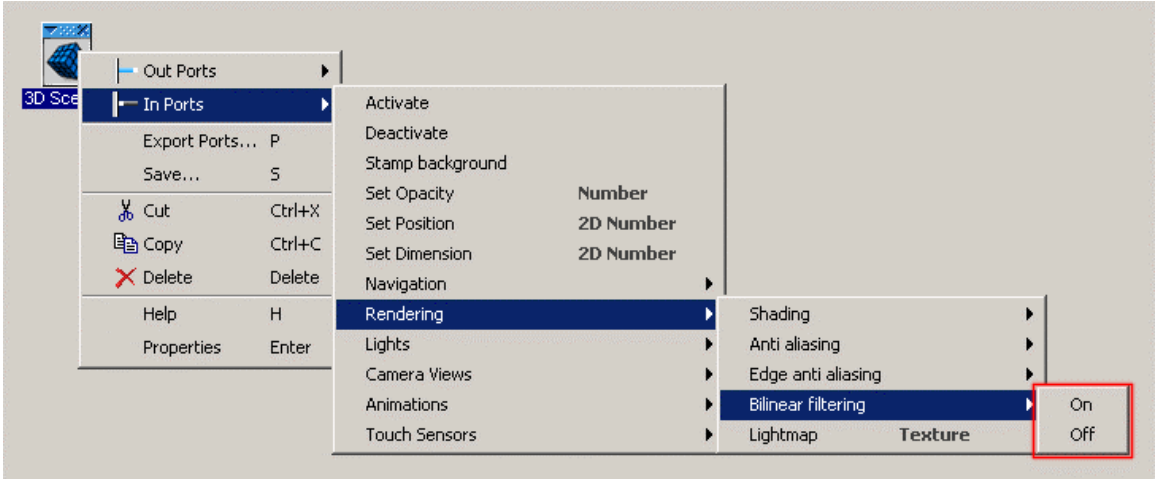

Figure 25. Bilinear filtering in-ports, found in all `3D Scene` objects

Visibility limit: (Group)

Front Clip: (Number)
Specifies the front (near) clipping distance between the display and the 3D model. There are no exact values for this setting, as it depends on the size of the 3D world and the 3D objects. You have to test it out for yourself.

Back Clip: (Number)
Specifies the back clipping distance. You can hence have an object disapear in the distance. There are no exact values for this setting, as it depends on the size of the 3D world and the 3D objects. You have to test it out for yourself.

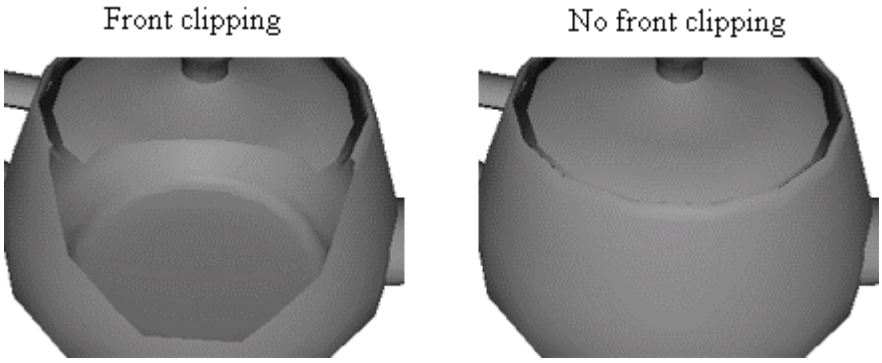

Figure 26. Front clipping vs. no front clipping

6.2.2 Appearance

In the *Appearance* tab (Figure 27) you specify and change object-, material- and light properties. The list window displays both objects and lights (Headlight, Directional and Omni).

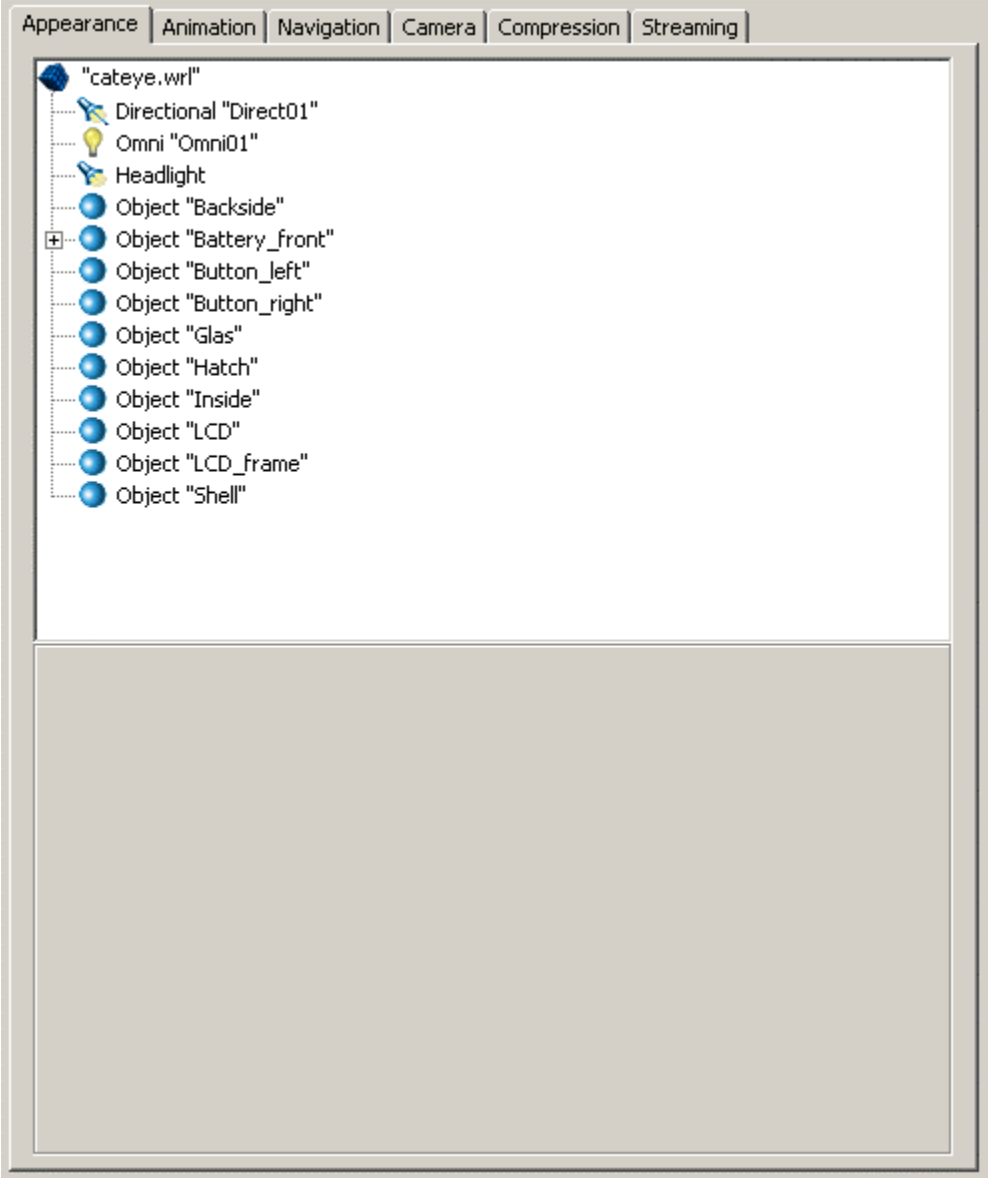

Figure 27. Different light sources listed

6.2.2.1 Lights

A 3D scene can have multiple colored and dynamic light sources. Two different types of light sources are supported, *Directional* light, which simulates light sources that are far away from the 3D objects and therefore has only a direction in the scene, and *Omni* light, which is a light that illuminates uniformly in all directions from a specified point in the 3D scene. All scenes have a *Headlight* automatically, which works as a *Directional* light.

You create your light sources in your 3D authoring tool, where you also animate the light source positions and directions if desired. Colors and intensities can then be adjusted from the *Appearance* tab in WF-3D. Click the light source you want to change and a property dialog will appear (Figure 28).

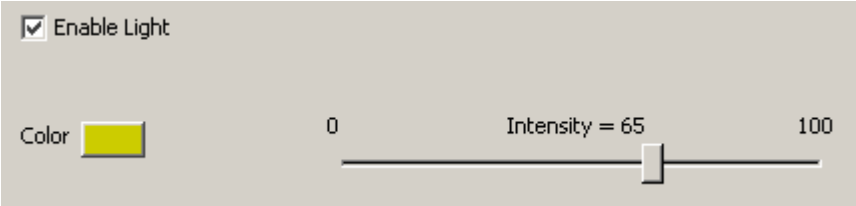

Figure 28. Light source settings

Enable Light: (checkbox)
Mark this checkbox to enable/activate the light.

Color: (color dialog)
Sets the light source color.

Intensity: (slider)
Sets the light source intensity. 0=minimum, 100=maximum

The above light settings can be programmed to alter dynamically in your presentation by using the *In-ports* > *Lights* > *"Light-name"* (Figure 29).

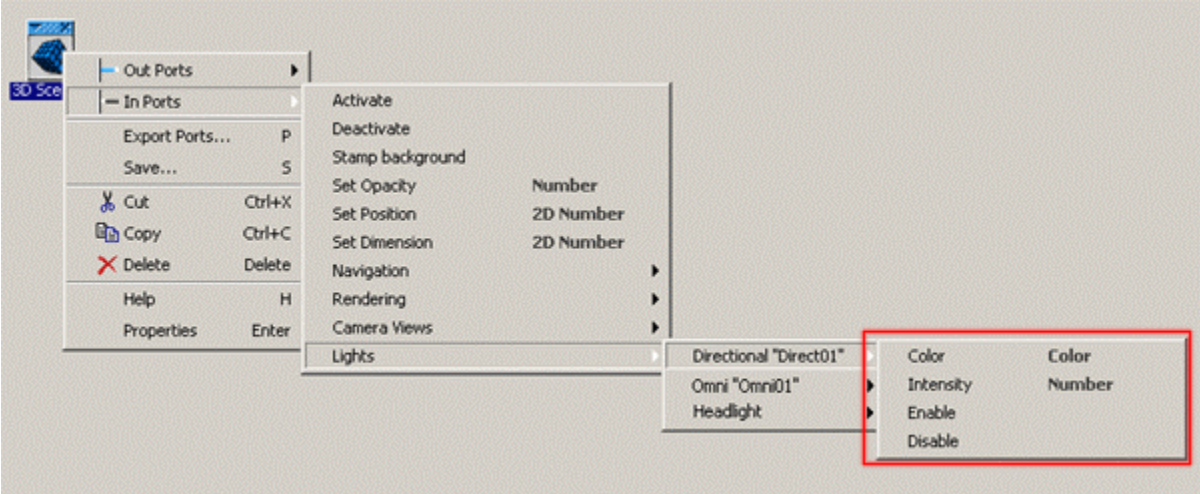

Figure 29. Lights in-ports

6.2.2.2 Objects

To work with an object, select the object from the list or click on the object in the *Preview* window. Objects are listed in the hierarchy from parent to child, as they were linked in the 3D authoring tool where they were created. Click the object you want to work with and a property dialog will appear below (Figure 26).

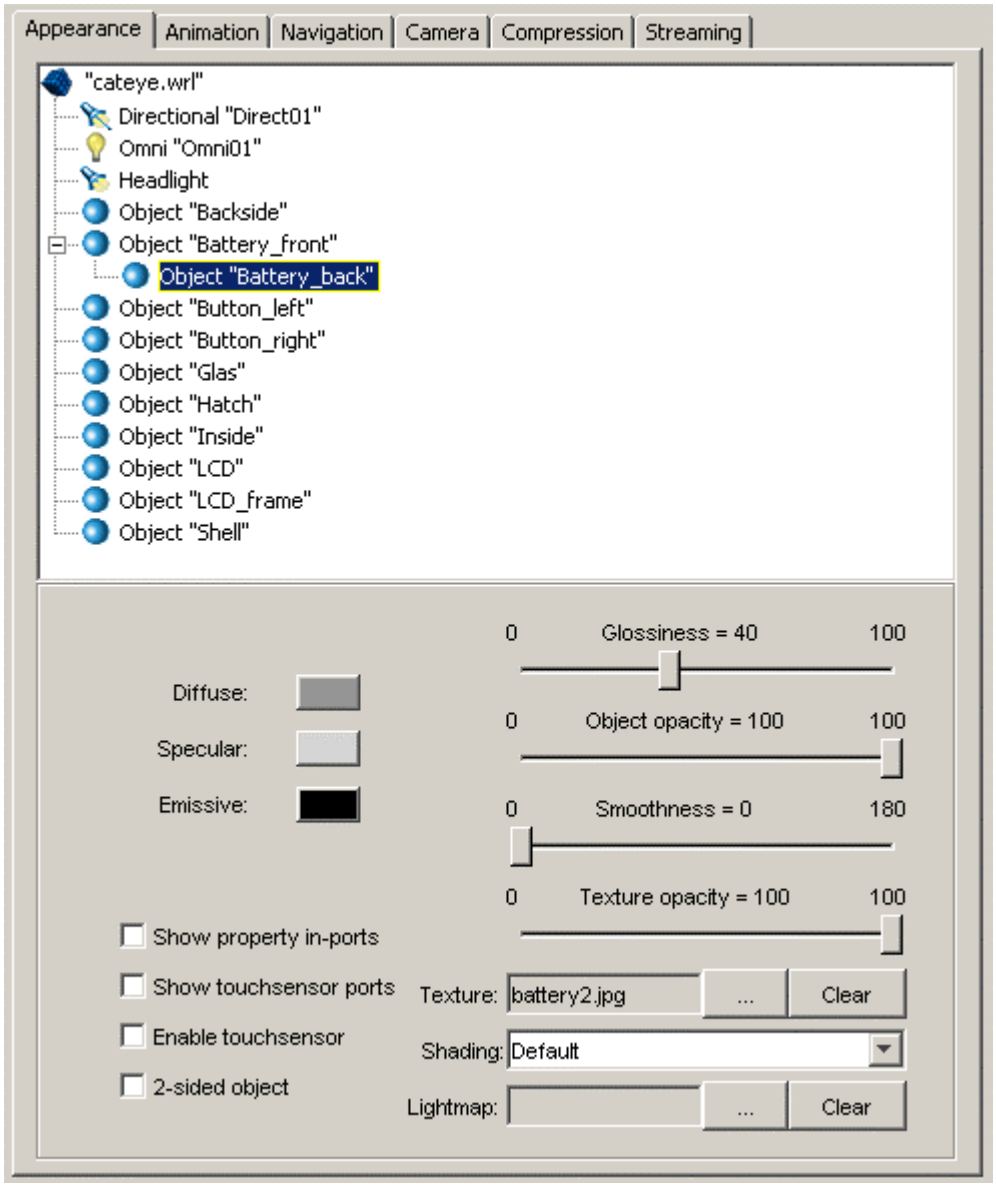

Figure 30. Object settings

Diffuse: (color dialog)
Sets the object diffuse color. The diffuse color is the color that the object reflects when illuminated by "direct daylight".

The diffuse color can be animated, see *Show property in-ports* below.

Specular: (color dialog)
Sets the specular color. Shows a highlight on an object if *Gouraud* or *Phong* shading is used and if the object has a smoothing angle (Smoothness) larger than 0. The size of the highlight is set with the *Glossiness* slider. (The specular color is exported from *3ds max* as combination of *Specular Color* and *Specular Level*).

The specular color can be animated, see *Show property in-ports* below.

Emissive: (color dialog)
Sets the emissive color (called *Self-illumination Color* in *3ds max*). Makes the object self-illuminating with this color.

The emissive color can be animated, see *Show property in-ports* below.

Glossiness: (slider)
Sets the shininess value or the reflection level for an object. If lightmaps are used, then the slider sets the reflection level from 0 to 100. 0 = no reflection, 100 = full reflection
For object shininess, the slider increases or decreases the size of a highlight in eight (8) steps (slider value 0,1,2,4,7,13,27,51) if *Gouraud* or *Phong* shading is used.

The glossiness value can be animated, see *Show property in-ports* below.

Object Opacity: (slider)
Sets the opacity level of the object. 0 = transparent, 100 = opaque

The object opacity can be animated, see *Show property in-ports* below.

Smoothness: (slider)
Sets the object crease angle in degrees (between 0 and 180 degrees). Overrides the *Default Smoothing* settings.

Texture Opacity: (slider)
Sets the opacity level of the object texture (if a texture is available). 0 = transparent, 100 = opaque

The texture opacity can be animated, see *Show property in-ports* below.

Texture: (loading dialog)
Shows the current texture (if any was mapped in the 3D authoring tool). "..." loads a new texture, and "Clear" removes the texture image. Supports JPEG, GIF (transparent) and PNG (transparent) format with any width and height.

Tip: A texture is usually static, but you can have animated and interactive textures, or you can replace a texture, during a presentation by using the `Texture` and `Texture Array` object. The new texture information is send from the texture object to the `3D Scene` *In-ports* > *Objects* > *"Object-name"* > *Texture* (Figure 31).

Note: These ports are only visible if the *Show property in-ports* checkbox is marked (see below).

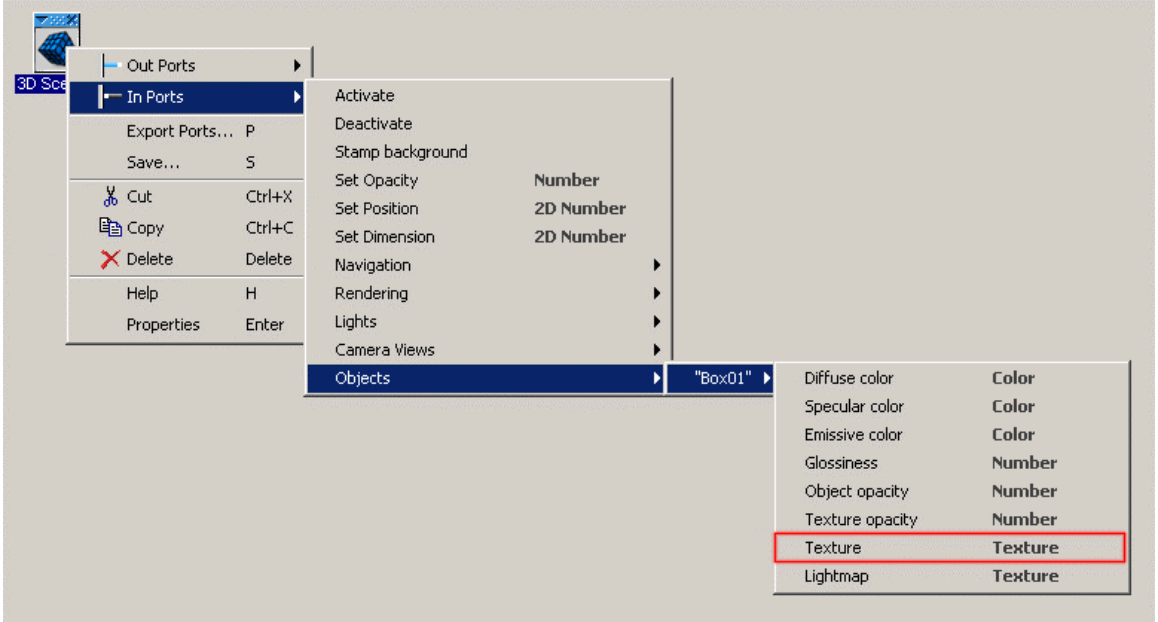

Figure 31. Texture in-port, working for all 3D objects with a mapped texture

Shading: (choice)
Sets shading method for the selected object (overrides the *Default Shading*). There are six options; *Default, Constant*, *Flat*, *Gouraud*, *Phong* and *Lightmap*. The *Default* option uses the *Default Shading* option set in the *Rendering* section. The other five options are the same as described above.

Lightmap: (loading dialog)
"..." loads a lightmap, which will be applied on the selected object if the *Lightmap* shading method is selected in the *Shading* menu (see above). If a default lightmap is set,

then this lightmap will override the default map. Supports JPEG, GIF or PNG format with any width and height.

Tip: A lightmap is usually static, but you can have animated lightmaps, or you can replace a lightmaps, during a presentation by using the `Texture` and `Texture Array` object. The new lightmap information is send from the texture object to the `3D Scene` *In-ports* > *Rendering* > *Lightmap* (Figure 19). Note that these ports are only visible if the *Show property in-ports* checkbox is marked (see below).

Note: A loaded lightmap will be included in the published presentation, even if it's not used. So make sure to remove it (Clear) if you don't use it.

Note: Lightmaps do not follow the VRML export from *3ds max* (or other 3D authoring tools) and must therefore be added manually.

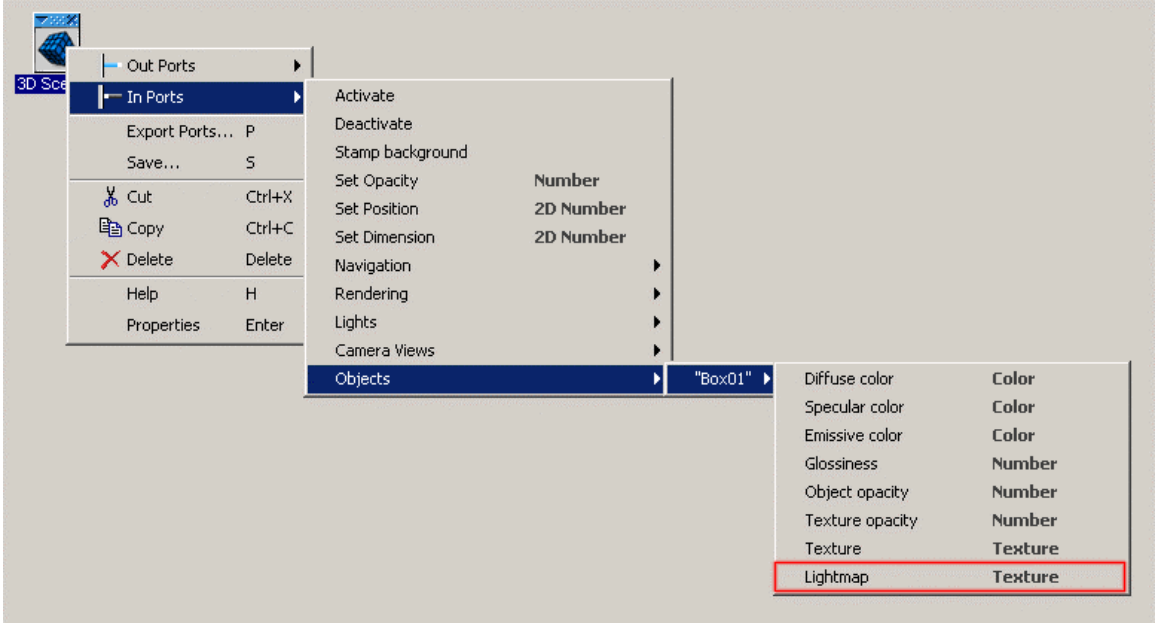

Figure 32. Lightmap in-port, working for all objects

Show property in-ports: (checkbox)
Mark this checkbox to have new in-ports, for the following features, shown in the `3D Scene` object: *Diffuse color*, *Specular color*, *Emissive color*, *Glossiness*, *Object opacity*, *Texture opacity*, *Texture* and *Lightmap* (you find information for each property above). This is necessary when you want to animate a property. The ports can be found at `3D Scene` *In-ports* > *Objects* > *"Object-name"* (Figure 33)

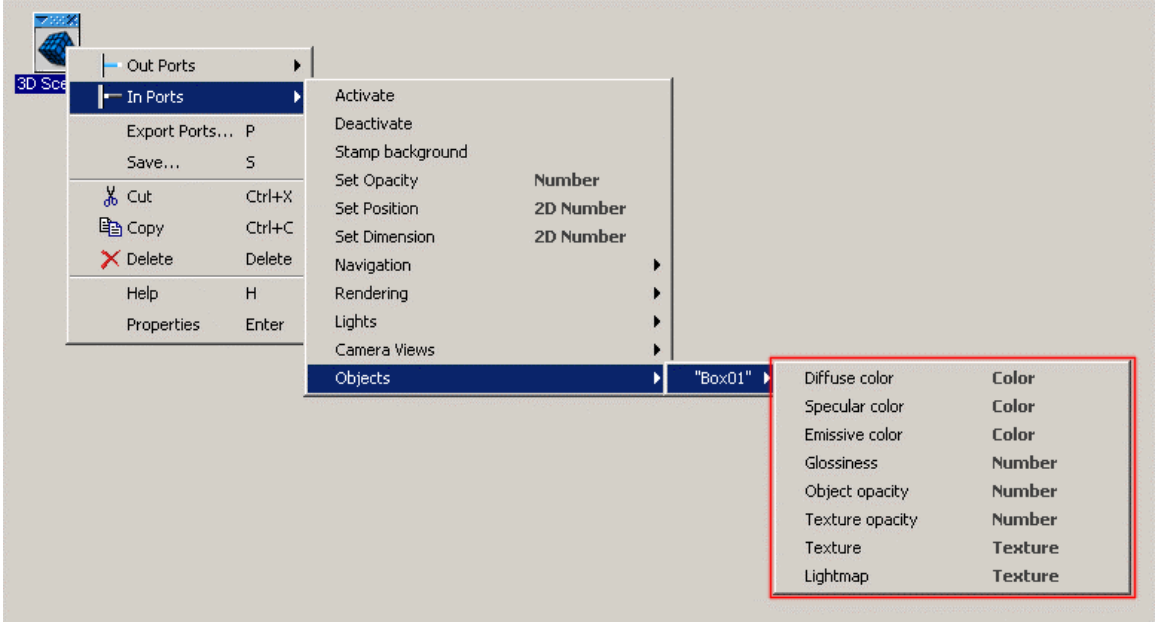

Figure 33. Properties shown as in-ports for the object "Box01"

Show touchsensor ports: (checkbox)
Mark this checkbox in order to show touchsensor ports for the object. Two new in-ports (Figure 34) and four out-ports (Figure 35) will be visible in the `3D Scene` object.

In-ports > Touch Sensors > "Object-name" > Enable:
Enables the touchsensor.

In-ports > Touch Sensors > "Object-name" > Disable:
Disables the touchsensor.

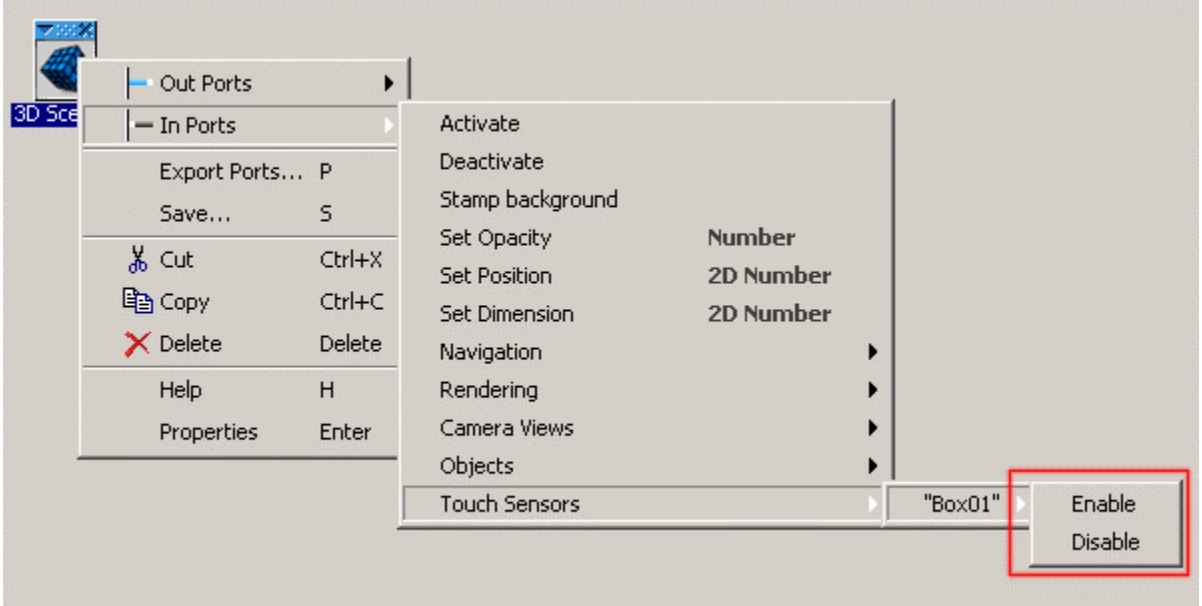

Figure 34. Touchsensor in-ports

Out-ports > Touch Sensors > "Object-name" > Mouse Press:
Fires when mouse button is pressed over the touchsensor.

Out-ports > Touch Sensors > "Object-name" > Mouse Release:
Fires when mouse button is released over the touchsensor.

Out-ports > Touch Sensors > "Object-name" > Mouse Roll Over:
Fires when mouse cursor enters the touchsensor.

Out-ports > Touch Sensors > *"Object-name" > Mouse Roll Out*:
Fires when mouse cursor exits the touchsensor.

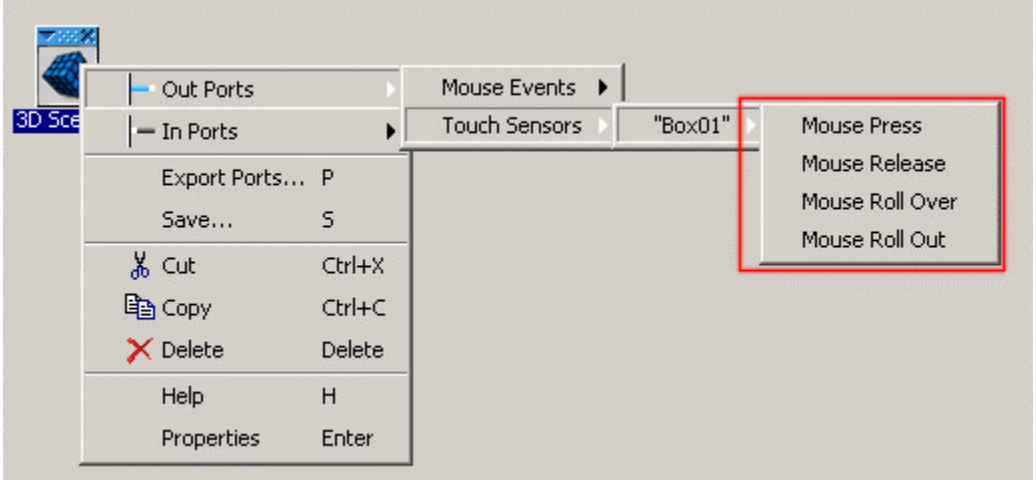

Figure 35. Touchsensor out-ports

Enable touchsensor: (checkbox)
Makes and enables the object to a touchsensor. You have to mark the checkbox *Show touchsensor ports* (see above) in order to send events to/from from the touchsensor. The mouse cursor will be changed to a handpointer when placed over an enabled touchsensor.

2-sided object: (checkbox)
Mark this checkbox to have object faces two-sided.

6.2.3 Animation

In the *Animation* tab (Figure 32) you preview both matrix (translation) and vertex (mesh) animations. Supported animation types are: object position, object scaling, object rotation, camera position, camera rotation, camera target, light position and mesh animations.

Note: You do not create any animations in WF-3D, that you to do in your 3D authoring tool.

There is also support for hierarchal animations. When an animation is specified on an object consisting of a hierarchical object structure the animations are separately controllable for each sub-object. This means that specifying an animation like the waving of a hand, you can run each finger's animation separately.

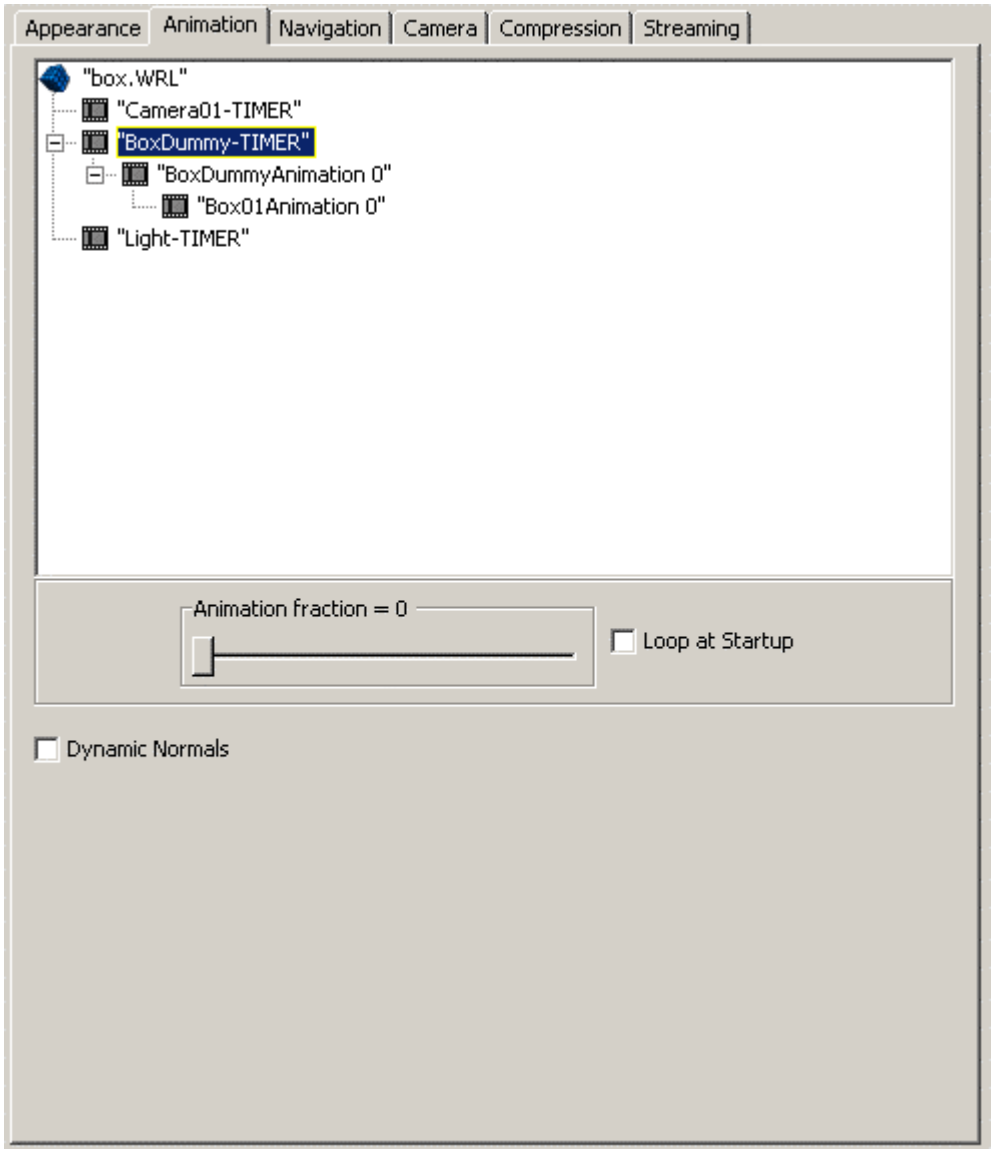

Figure 36. The *Animation* tab with some animations

Animation fraction: (slider)
Sets the fraction (in percentage) of a selected animation (select from the list) between your *Start Time* and *End Time* (*Start Time* specifies the beginning of your active time segment in *3ds max* and *End Time* the end of your active time segment). 0 = beginning, 100 = end

In-ports are automatically created for all animations. To run an animation (or set an animation fraction), send numbers between 0 and 100 to the in-port *Set Animation Fraction* for the specific animation (Figure 37).

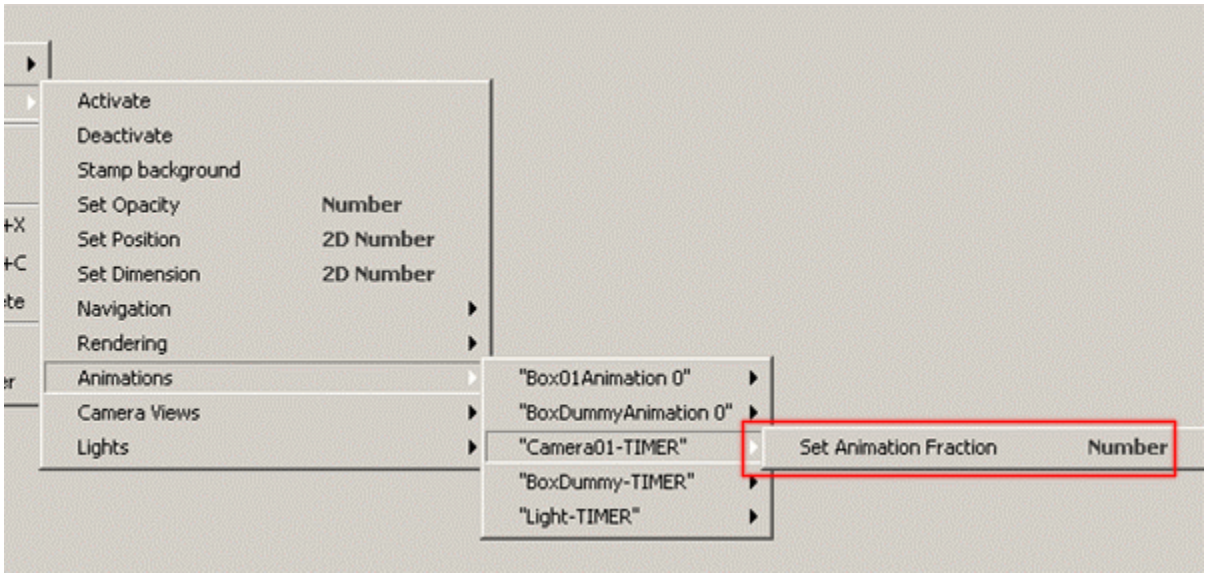

Figure 37. Animation in-port *Set Animation Fraction*

Loop at Startup: (checkbox)
Mark this checkbox if you want the animation to run and loop at the presentation startup.

Tip: You cannot set the loop time with this option. To have a time-controlled loop, use a `Progressor` object instead, connected to the *Set Animation Fraction* port.

Dynamic Normals: (checkbox)
Mark this checkbox to use dynamic normal calculation. When an object is morphed (vertex animation) you may want to recalculate the vertex normals to make sure the lighting is correctly displayed.

Note: *Dynamic Normals* can be very CPU consuming, as the normals have to be recalculated for every animation frame, and should therefore only be used in situations when really needed.

6.2.4 Navigation

In the *Navigation* tab (Figure 38) you configure and customize navigation modes for different occasions.

Note: In WF-3D you don't actually rotate, pan and zoom the model. The model is fixed and you move the cameras around the world center and the camera targets.

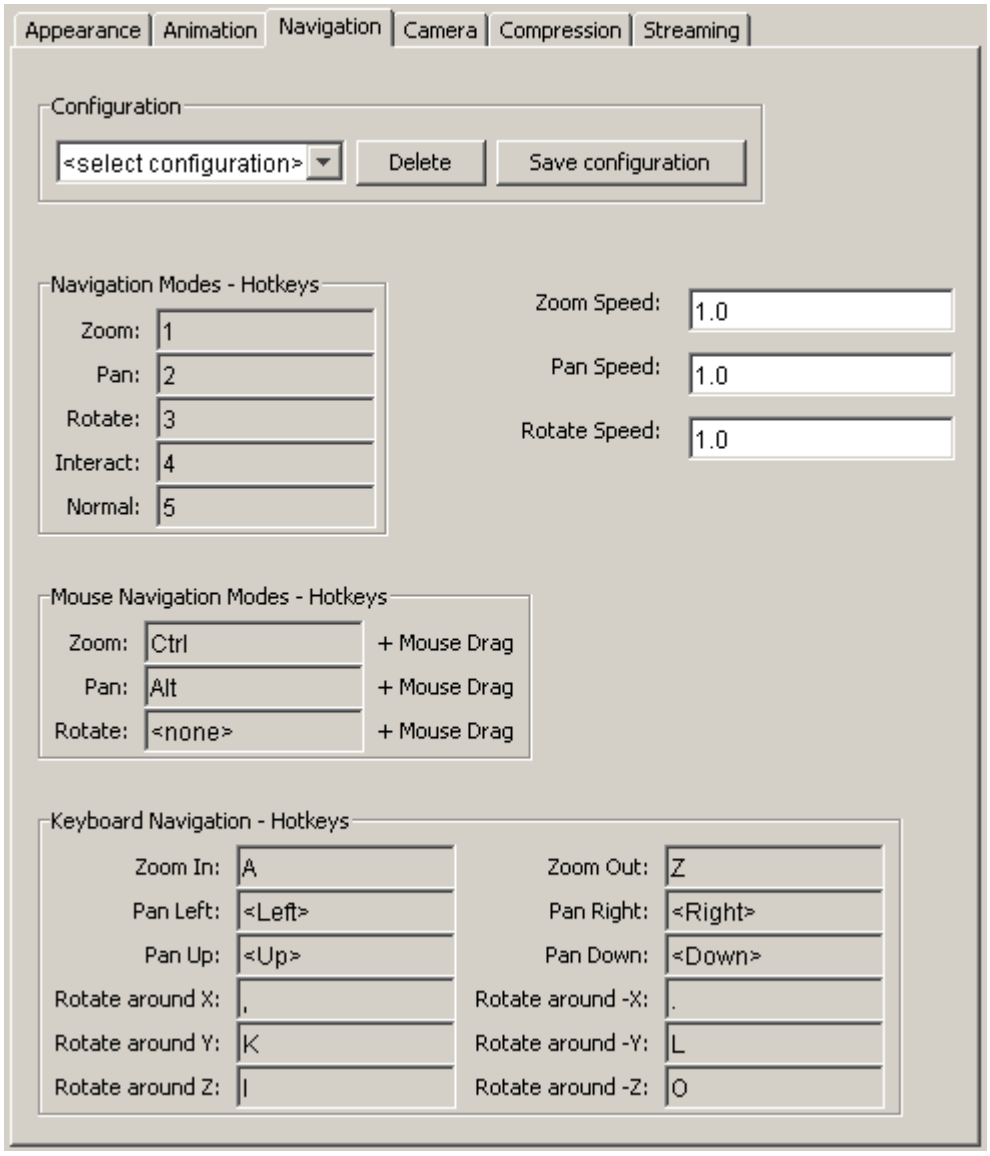

Figure 38. The *Navigation* tab

Configuration: (group)
Save and load your favorite navigation configurations.

Select configuration: Choose between saved navigation configurations.

Delete: Delete a saved navigation configuration.

Save configuration: Save a navigation configuration.

Different 3D models have different navigation speeds. This normally depends on the size of the models compared to the 3D world when they were created. You can therefore adjust the navigation speed for *Zoom*, *Pan* and *Rotate* so they suit your requirements.

Zoom speed: (number)
Sets the zoom speed. (To test new values in the *Preview* window, click the *Apply* button)

Pan speed: (number)
Sets the pan speed. (To test new values in the *Preview* window, click the *Apply* button)

Rotate speed: (number)
Sets the rotate speed. (To test new values in the *Preview* window, click the *Apply* button)

Navigation Modes - Hotkeys: (group)
There are five different navigations modes. By using a navigation mode you make the navigation easier for the viewer. For example, if *Rotation* mode is used, then it is only possible to rotate the model using the mouse (left-click and drag). If *Pan* mode is used, then it is only possible to pan the model, and so on.

> **Zoom:** (configurable)
> Choose a hotkey to use for switching to zoom mode. Click the field and press a hotkey (or a hotkey combination, like Ctrl+hotkey).

> **Pan:** (configurable)
> Choose a hotkey to use for switching to pan mode. Click the field and press a hotkey (or a hotkey combination, like Ctrl+hotkey).

Rotate: (configurable)
Choose a hotkey to use for switching to rotate navigation mode. Click the field and press a hotkey (or a hotkey combination, like Ctrl+hotkey).

Interact: (configurable)
Choose a hotkey to use for switching to interact mode. Normally used when an interactive `Texture` object has replaced a texture. Click the field and press a hotkey (or a hotkey combination, like Ctrl+hotkey).

Normal: (configurable)
Choose a hotkey to use for switching to normal navigation mode. In *Normal* mode, which is the default, it is possible to temporarily change navigation mode with the *Mouse Navigation Modes - Hotkeys* (see below).

The navigation modes can be programmed to alter dynamically in your presentation by using the *In-ports > Navigation > Mode* (Figure 39).

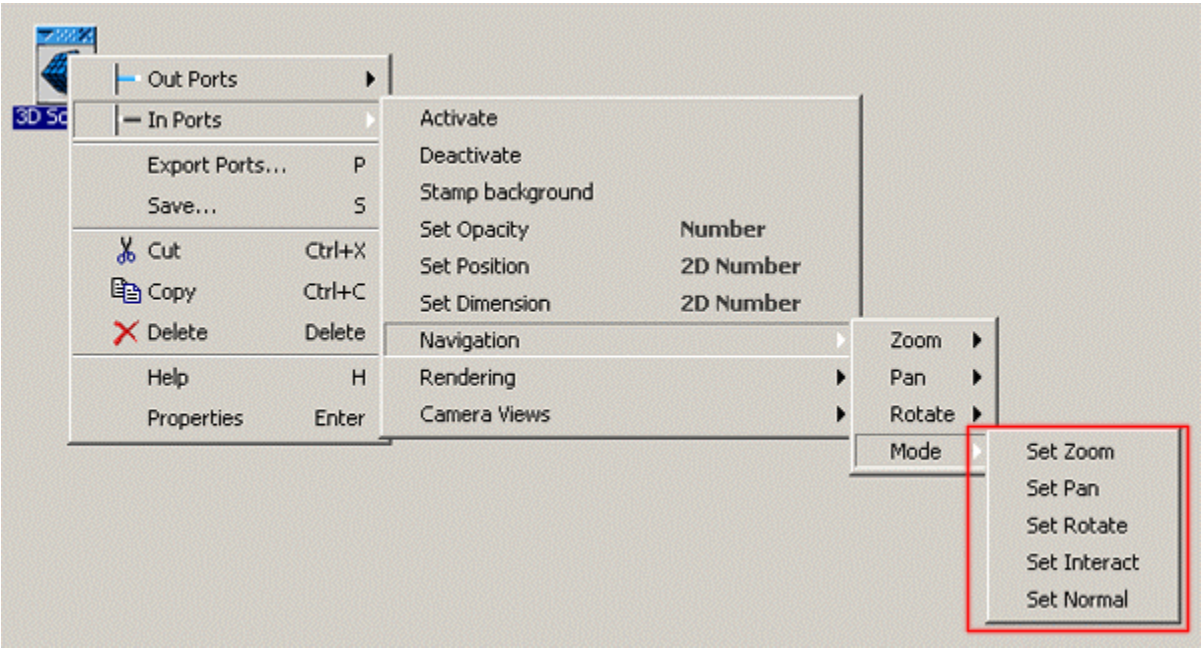

Figure 39. Navigation Modes in-ports

Mouse Navigation Modes - Hotkeys: (group)

If the navigation mode *Normal* is used (see above) then it is possible to temporarily change the navigation mode, for example to *Zoom* mode, by pressing a hotkey and using the mouse (left-click and drag), when releasing the hotkey the navigation returns to the *Normal* mode again.

Zoom: (configurable)
Choose a hotkey, to work in conjunction with the left mouse button, for zooming.
Works when navigation mode *Normal* is used (see above)

Pan: (configurable)
Choose a hotkey, to work in conjunction with the left mouse button, for panning.
Works when navigation mode *Normal* is used (see above)

Rotate: (configurable)
Choose a hotkey, to work in conjunction with the left mouse button, for rotating.
Works when navigation mode *Normal* is used (see above)

Keyboard Navigation Modes - Hotkeys: (group)

Zoom In: (configurable)
Choose a hotkey for zooming in.

Zoom Out: (configurable)
Choose a hotkey for zooming out.

Zooming in and out can be programmed in your presentation by using the *In-ports* > *Navigation* > *Zoom* (Figure 40). Each incoming pulse make the model zoom one step. The steps further depend on the *Zoom Speed* (see above).

Figure 40. Zoom in and zoom out ports

Pan Left: (configurable)
Choose a hotkey for panning the camera left.

Pan Right: (configurable)
Choose a hotkey for panning the camera right.

Pan Up: (configurable)
Choose a hotkey for panning the camera up.

Pan Down: (configurable)
Choose a hotkey for panning the camera down.

Panning of the model be programmed in your presentation by using the *In-ports* > *Navigation* > *Pan* (Figure 41). Each incoming pulse make the model pan one step. The steps further depend on the *Pan Speed* (see above).

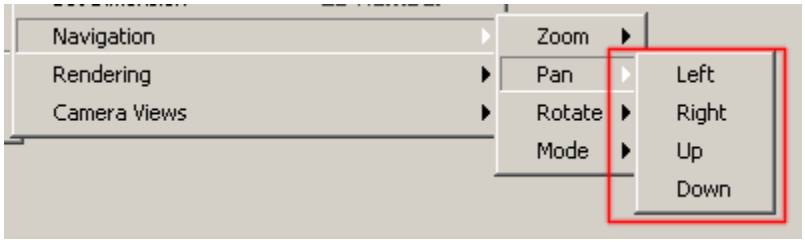

Figure 41. Panning ports

Rotate around X: (configurable)
Choose a hotkey for a positive rotation of the camera around the x-axis.

Rotate around -X: (configurable)
Choose a hotkey for a negative rotation of the camera around the x-axis.

Rotate around Y: (configurable)
Choose a hotkey for a positive rotation of the camera around the y-axis.

Rotate around -Y: (configurable)
Choose a hotkey for a negative rotation of the camera around the y-axis.

Rotate around Z: (configurable)
Choose a hotkey for a positive rotation of the camera around the z-axis.

Rotate around -Z: (configurable)
Choose a hotkey for a negative rotation of the camera around the z-axis.

Rotation of the model be programmed in your presentation by using the *In-ports* > *Navigation* > *Rotate* (Figure 42). Each incoming pulse make the model rotate one step. The steps further depend on the *Rotate Speed* (see above).

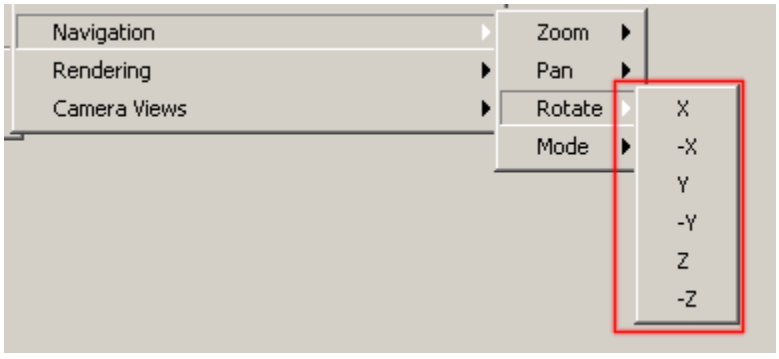

Figure 42. Rotating ports

6.2.5 Camera

In the *Camera* tab (Figure 43) you can add and delete cameras, change camera positions and *Field Of View* (FOV) values, switch between camera rotation modes and set camera restrictions. All cameras in your project are listed in this tab, even cameras created in your 3D authoring tool.

Note: In WF-3D you don't actually rotate, pan and zoom (scale) the model. The model is fixed in the 3D world and you move the camera around the world center and the camera target. Therefore, make sure to place your models and camera targets in the world center in your 3D authoring tool, in order to get a proper rotation around your model in WF-3D.

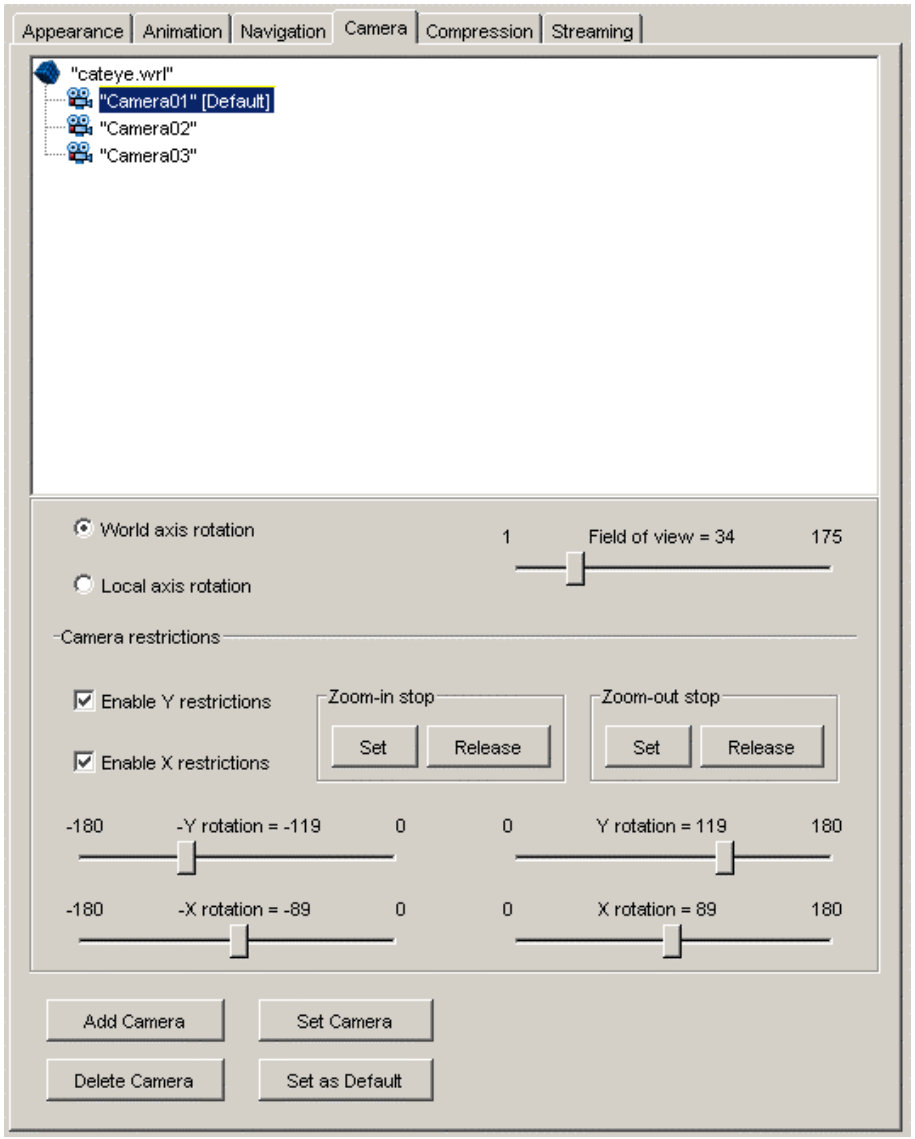

Figure 43. The *Camera* tab

World axis rotation: (radiobutton)
Choose this option to rotate the camera around the world center and the world axis (Figure 44), rather than the local camera axis.

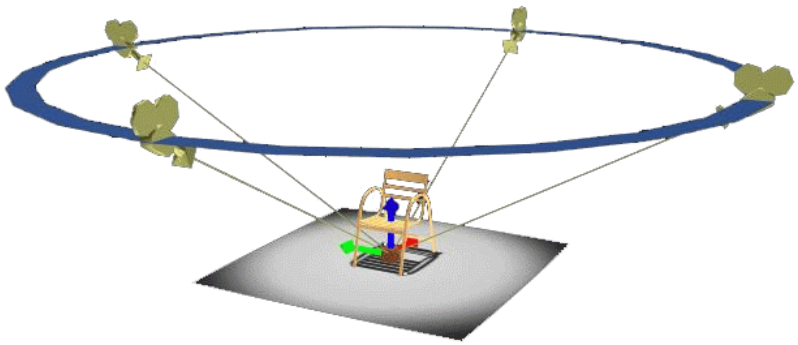

Figure 44. Camera rotation around world axis

Local axis rotation: (radiobutton)
Choose this option to rotate the camera around the world center and the local camera axis (Figure 45), rather than the world axis.

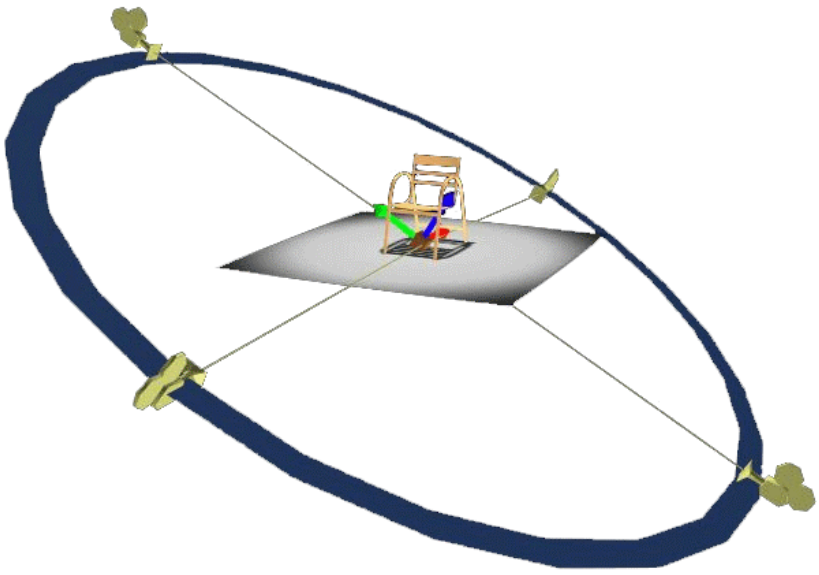

Figure 45. Camera rotation around local camera axis

Field of view: (slider)
Sets the Field Of View (perspective) value in degrees. Select a camera in the list and set the slider. 1 = minimum, 175 = maximum

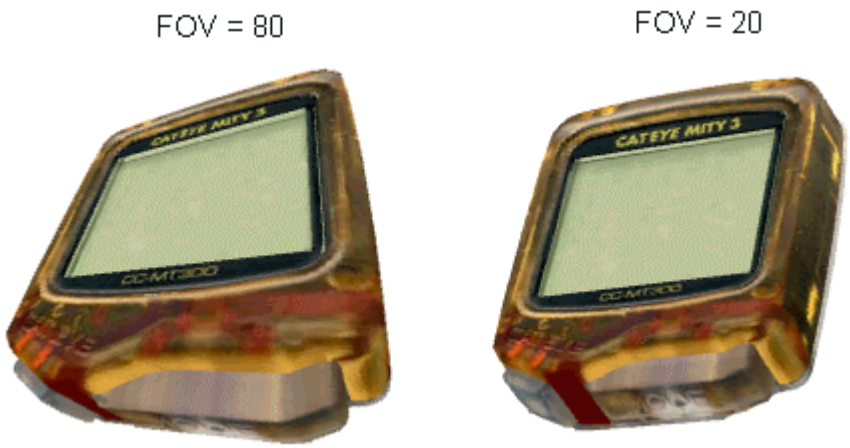

Figure 46. Comparison between high and low perspective

Camera restrictions

In the *Camera restrictions* section you can set zoom and rotation constrains for the selected camera.

Zoom-in stop

The *Zoom-in stop* feature allows you to set a stop for the camera when zooming in. The default setting is set to the world center.

Set: (button)
Sets the zoom-in limit for the chosen camera and for the present camera view, as seen in the *Preview* window.

Release: (button)
Releases zoom-in limit for the chosen camera.

Zoom-out stop

The *Zoom-out stop* feature allows you to set a stop for the camera when zooming out. The default setting is set to the infinity.

Set: (button)
Sets the zoom-out limit for the chosen camera and for the present camera view, as seen in the *Preview* window.

Release: (button)
Releases zoom-out limit for the chosen camera.

Enable Y restrictions: (checkbox)
Enables camera restrictions for rotation around the Y-axis. Use *Y-rotation* sliders to set angles.

Enable X restrictions: (checkbox)
Enables camera restrictions for rotation around the X-axis. Use *X-rotation* sliders to set angles.

-Y rotation: (slide)
Sets the camera rotation restriction around the Y-axis (in the negative part of the X-axis). Only applicable if *World axis rotation* is chosen. -180 (degrees)=no restriction, 0 (degrees)=full restriction

Y rotation: (slide)
Sets the camera rotation restriction around the Y-axis (in the positive part of the X-axis). Only applicable if *World axis rotation* is chosen. 0 (degrees)=full restriction, 180 (degrees)=no restriction

-X rotation: (slide)
Sets the camera rotation restriction around the X-axis (in the negative part of the Y-axis). Only applicable if *World axis rotation* is chosen. -180 (degrees)=no restriction, 0 (degrees)=full restriction

X rotation: (slide)
Sets the camera rotation restriction around the X-axis (in the positive part of the Y-axis). Only applicable if *World axis rotation* is chosen. 0 (degrees)=full restriction, 180 (degrees)=no restriction

Add Camera: (button)
Adds a new camera to your scene (a new camera will appear in the list).

Delete Camera: (button)
Deletes a camera from your scene. Select a camera in the list and click the *Delete* button.

Set Camera: (button)
Sets the current *Preview* position to the selected camera.

Set as Default: (button)
Sets the selected camera to the default camera, i.e. the camera view that will be used at the presentation startup. Demarked with [Default] after the camera name in the list.

There are two in-port types that are created automatically for all your cameras; one that switches camera view and one that controls the zoom (Figure 47).

To program the zoom, use the *In-ports* > *Camera* > *Set zoom fraction*

The zoom fraction in-port zooms between the two zoom constrains, *Zoom-in stop* (max. zoom-in) and *Zoom-out stop* (max. zoom-out). The in-port receives values between 0 and 100, and works for all the cameras in a scene. The value 100 sets the camera to the *Zoom-in stop* position (max. zoom-in), while the value 0 sets the camera to the *Zoom-out stop* position (max. zoom-in). If no zoom constrains have been set, then the value 100 sets the max. zoom-in to the world center and value 0 to the double default camera distance.

To switch between different camera views in a presentation, use the *In-ports* > *Camera* > *Set "Camera-name"*

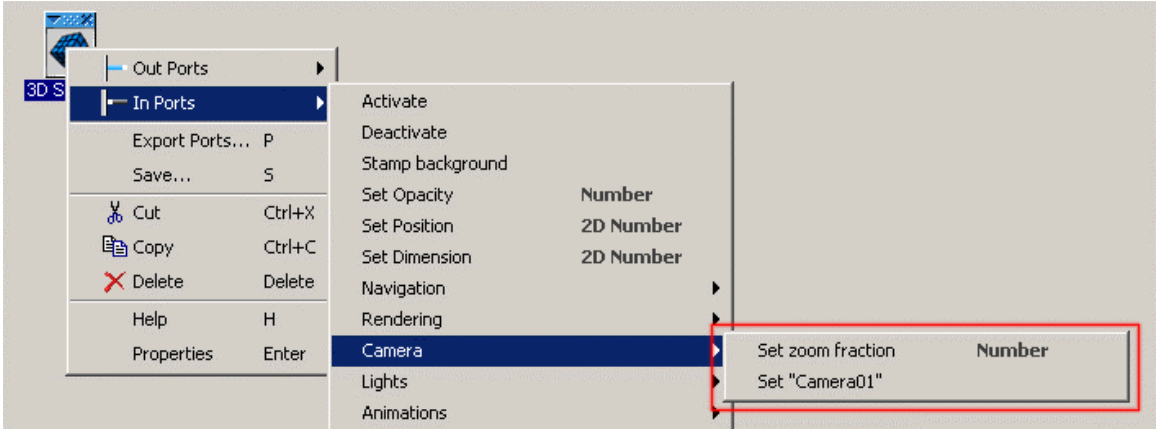

Figure 47. Camera in-ports

6.2.6 Compression

In the *Compression* tab (Figure 48) you set the compression ratio for your model. All 3D data are compressed; mesh data, texture coordinate data and vertex animation data. Generally, a file size compression of 90-95 % is achieved compared to the original (uncompressed) VRML-file format.

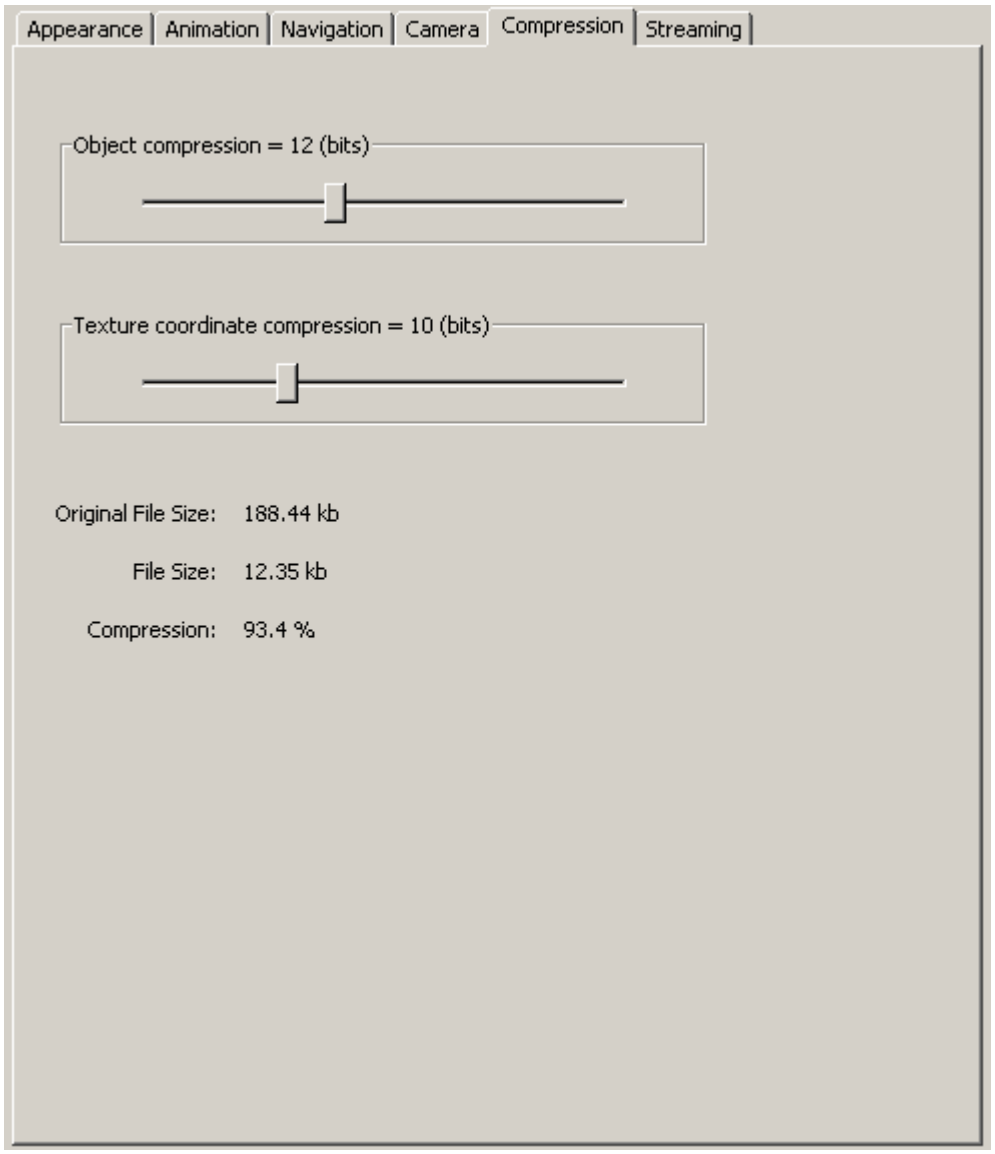

Figure 48. The *Compression* tab

Object compression: (slider)
Compresses the object coordinates precision. Lower values give smaller files, but also lower quality. The default settings are normally the best.

Texture coordinate compression: (slider)
Compresses the texture coordinates precision. Lower values give smaller files, but also lower quality. The default settings are normally the best.

Original File Size: (information)
Indicates the original VRML file size in kilobytes.

File Size: (information)
Indicates the new and compressed file size in kilobytes.

Compression: (information)
Indicates the compression ratio between the original file and the compressed file.

6.2.7 Streaming

In the *Streaming* tab (Figure 49) you control if the 3D model should be streamed into the presentation, and if so, the order the objects should be streamed. Streaming can increase the user experience, as the presentation will start quicker.

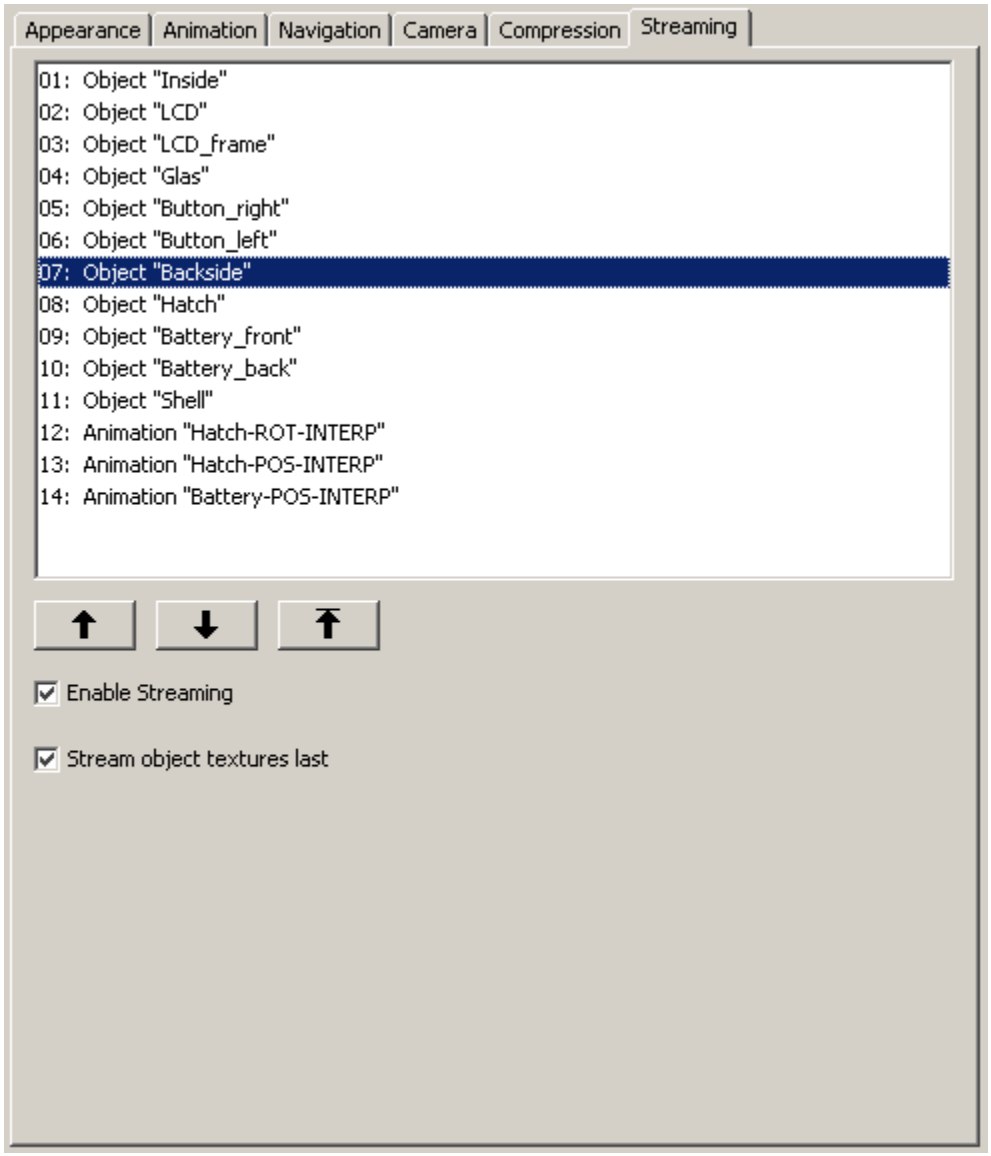

Figure 49. The *Streaming* tab

Enable Streaming: (checkbox)
Mark this checkbox to enable streaming of the model. The objects and animations will be streamed in the listed order. To reorder, mark an item and then click the up and down arrows.

If you choose not to stream the objects, then the complete 3D model and its textures will appear in the WireFusion *Loading Manager* (*Project* > *Loading Manager*...).

Stream object textures last: (checkbox)
Mark this checkbox to have the textures streamed last, after the objects have been loaded. If not checked, each texture will be streamed together with its corresponding object.

7 Texture object

Most 3D presentations seen on the web today only have static textures. Normally you can rotate, zoom and pan the objects. Some also have the possibility to run animations, e.g. changing a color or opening a hatch. But you very seldom see, for example, a digital watch with a working digital display. A display that has working time, a working stopwatch, working alarms etc. This is possible with WireFusion and WF-3D!

The `Texture` object, included in WF-3D, is used for creating advanced textures, advanced in the matter that they can contain logic, animations and interactivity. The `Texture` object has a normal WireFusion *Stage* (and an *Alpha Channel* if needed), on which you can create a presentation. It could for example be the display of a mobile phone, a touch screen for a handheld computer or the display of a digital watch. The applications are almost endless, and the functionality and the complexity is decided by you and your imagination, and of course, how good you are at using WireFusion.

Shortly, this is how it works:

- Let's say you want to create a handheld computer (PDA) with a working touch screen.
- You create your 3D model in your 3D authoring tool, as usual. Make sure that the PDA has a separate "display" object with a texture mapped to it.
- When imported to the `3D Scene` object, make the in-ports visible for the "display" object by marking the *Show property in-ports* checkbox.
- Use a `Texture` object and send its stage contents to the "display" in-port, *Texture*.
- All the logic and interactivity programmed on the `Texture` object's *Stage* will be visible on the "display" object, instead of the initial static texture that was mapped in the 3D authoring tool.

We illustrate the process of creating an interactive texture with a simple example.

Example

We want to replace the static texture of a 3D box with a `Texture` object, on which we want to write a text.

1. Insert a `3D Scene` and load a 3D box with a texture on it. Load 'box.wrl'
2. In the *Appearance* tab, mark the *Show property in-ports* checkbox for the box object. Click OK to close the dialog. Set the `3D Scene` *Target Area* to the same size as the *Stage* dimension (Alt+S).

3. Insert a `Texture` object. When its dialog opens, click OK to close it.
4. Connect: 'Texture 1', *Out-ports > Texture Pushed [Texture]* to '3D Scene 1', *In-ports > Objects > 'Box01' > Texture [Texture]*
5. Open the `Texture` object's local menu and choose *Explore Texture* to jump to the `Texture` *Stage*
6. In the empty `Texture` *Stage*, insert a `Keyboard` object and a `Text Window` object.
7. In the `Text Window` dialog, choose font size 20. Click OK to close the dialog. Set the `Text Window` *Target Area* to the same size as the *Stage* dimension (Alt+S).
8. Connect: 'Keyboard 1', *Out-ports > Key Pressed [Text]* to 'Text Window 1', *In-ports > Append Text [Text]*

Step 1

Insert a `3D Scene` object into a new and empty project, *Objects > 3D > 3D Scene*

Load a 3D box, which has a texture on it. Load 'box.wrl' found at:

- *[Path]/WireFusion 3/resources/3d_models/boxes/*

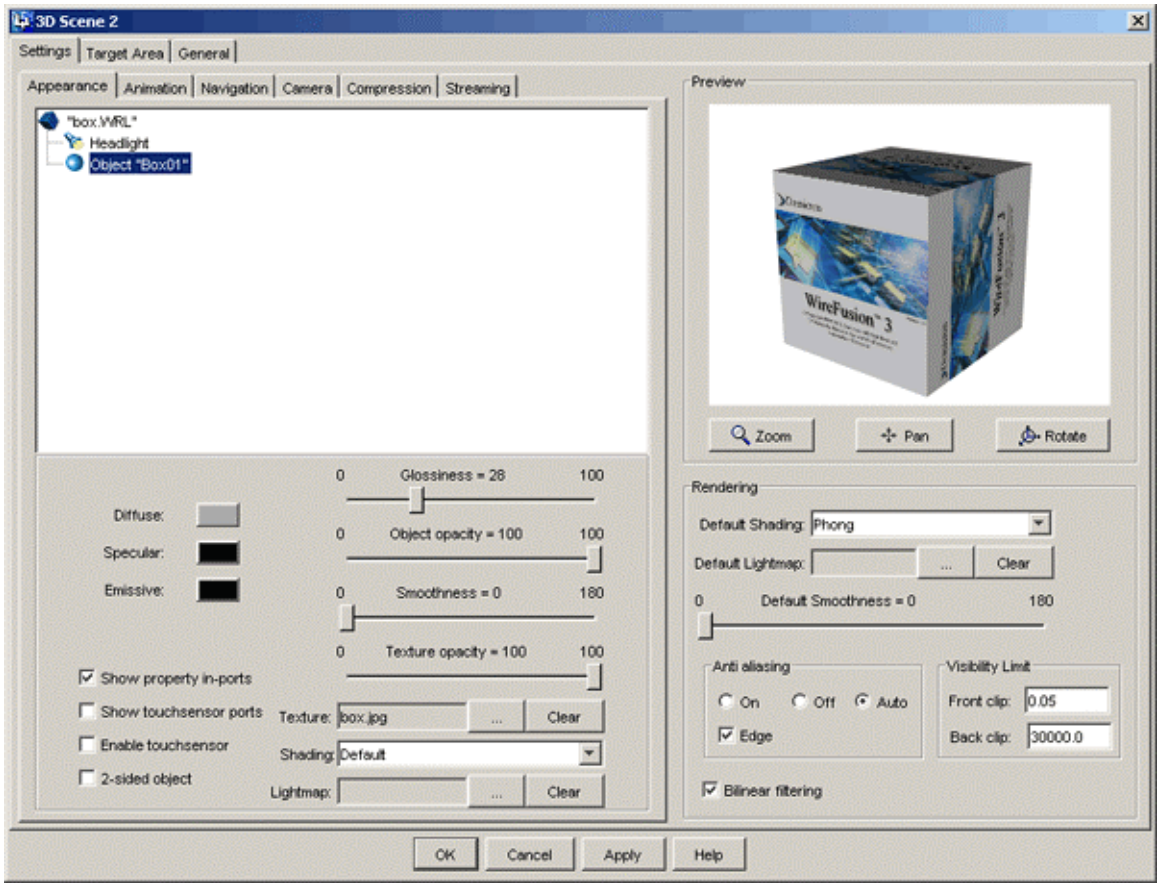

Figure 50. 'box.wrl' loaded

Step 2

In the *Appearance* tab, select the 'Box01' object in the list, and then mark the *Show property in-ports* checkbox (Figure 51) in order to make its properties visible as in-ports (Figure 50).

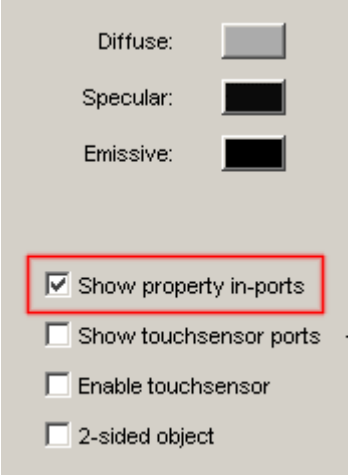

Figure 51. The *Show property in-ports* checkbox

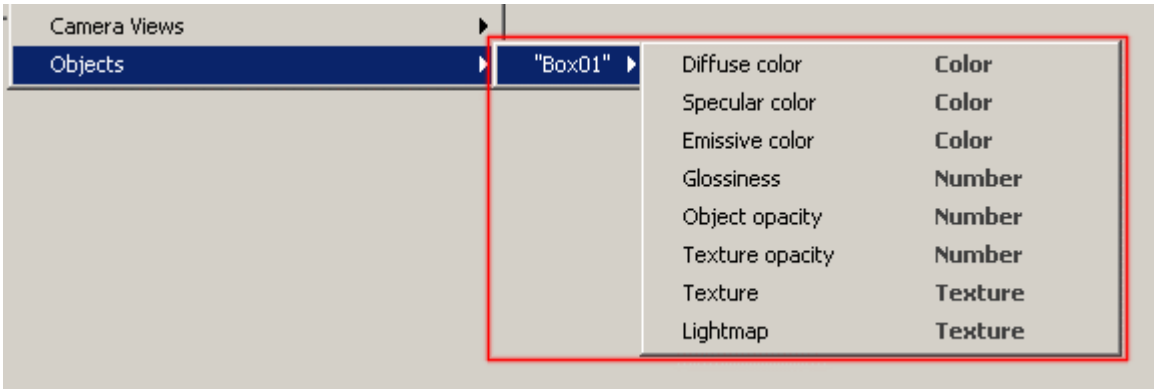

Figure 52. 'Box01' object's properties visible as in-ports

Set the '3D Scene 1' *Target Area* to the same size as the *Stage* dimension (240x180). Select the '3D Scene 1' object and press Alt+S on your keyboard to resize its *Target Area*.

Step 3

Insert a `Texture` object into your project, *Objects* > *3D* > *Texture*

When its dialog (Figure 53) opens, click OK to close it.

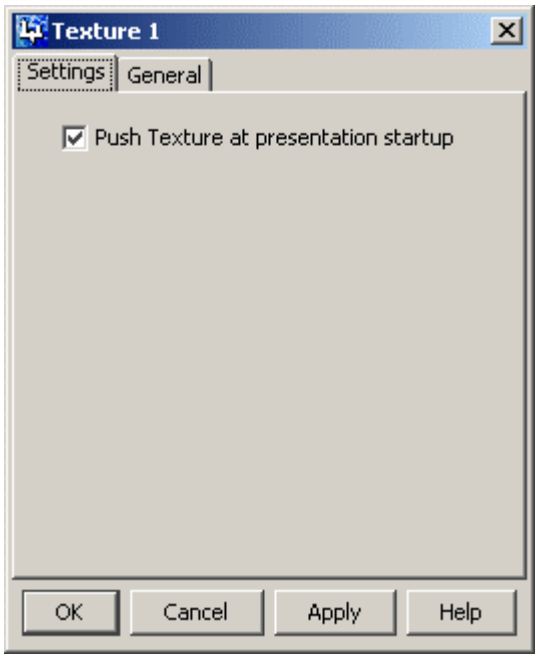

Figure 53. The `Texture` dialog

Note: The `Texture` object has a checkbox named *Push Texture at Presentation startup*. Mark this to have the `Texture` *Stage* sent out through the out-port *Texture Pushed* at the presentation startup. If not checked you'll need to trigger the in-port *Push Texture* in order to send out its content.

Step 4

Replace the texture mapped on the 3D box with the *Stage* of 'Texture 1' (which at this moment still is empty).

Connect:

- 'Texture 1', *Out-ports* > *Texture Pushed [Texture]* to '3D Scene 1', *In-ports* > *Objects* > *'Box01'* > *Texture [Texture]*

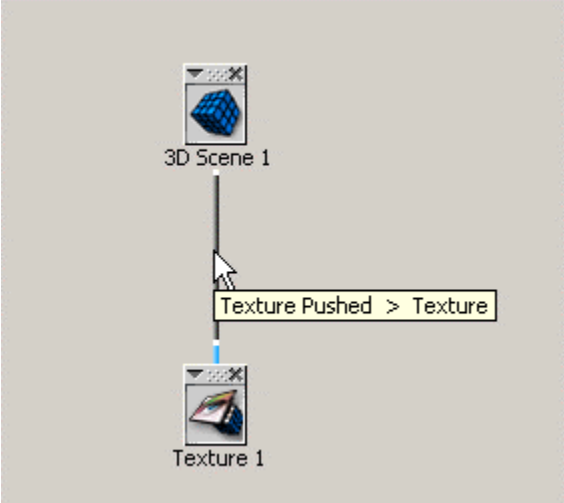

Figure 54. 'Texture 1' connected to '3D Scene 1'

Step 5

Open the `Texture` *Stage* by choosing *Explore Texture* from the 'Texture 1' object menu (Figure 55).

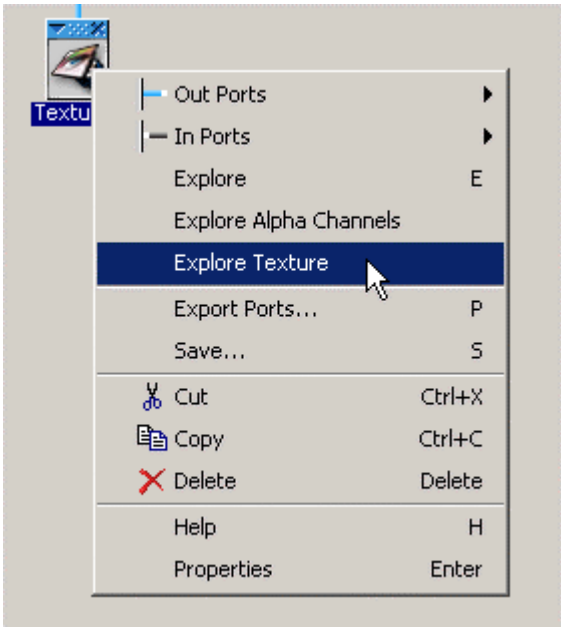

Figure 55. Jumping to the `Texture` *Stage*

Step 6

In the empty `Texture` *Stage*, insert a `Keyboard` object, *Objects* > *Enviroment* > *Keyboard*

and a `Text Window` object, *Objects* > *Widgets* > *Text Window*

Step 7

When the `Text Window` is dropped, its dialog opens up. Choose font size 20 and then click OK to close the dialog (Figure 56).

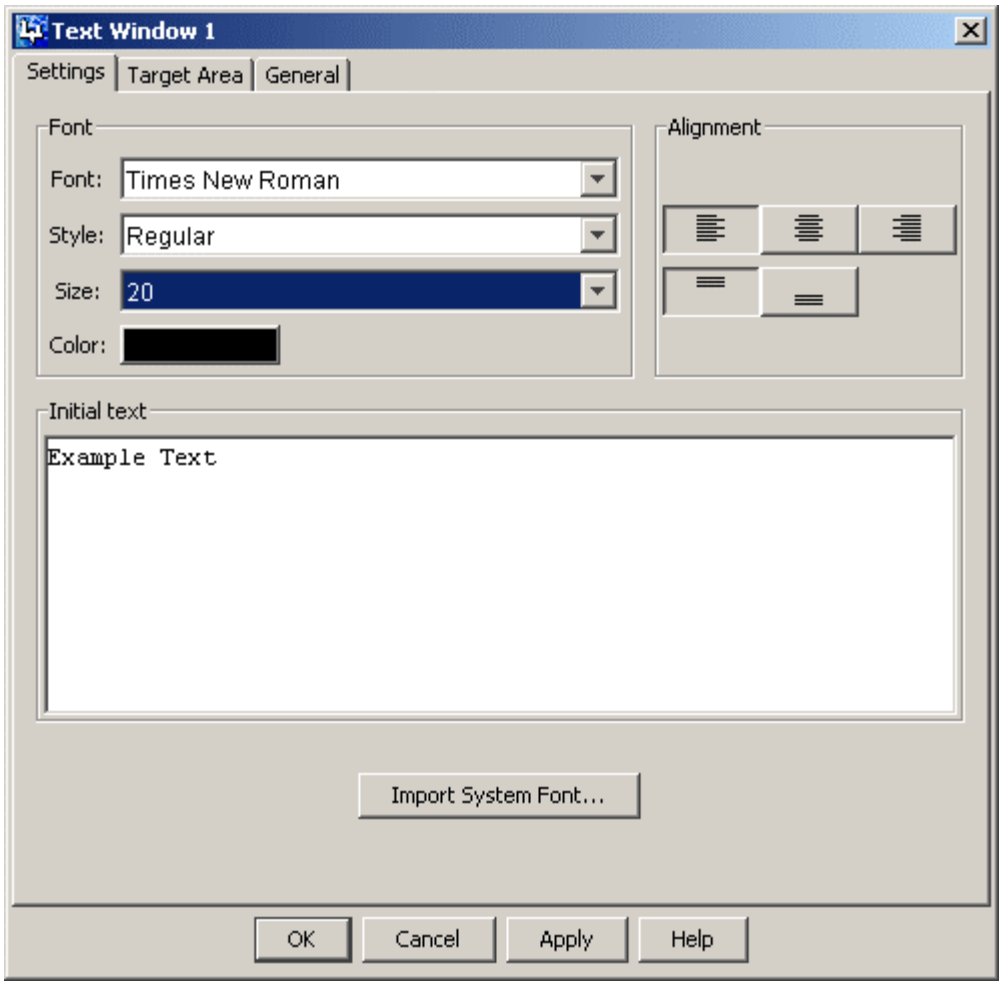

Figure 56. `Text Window` dialog

When the dialog is closed, set the `Text Window` *Target Area* to the same size as the *Stage* dimension (Alt+S) (Figure 57).

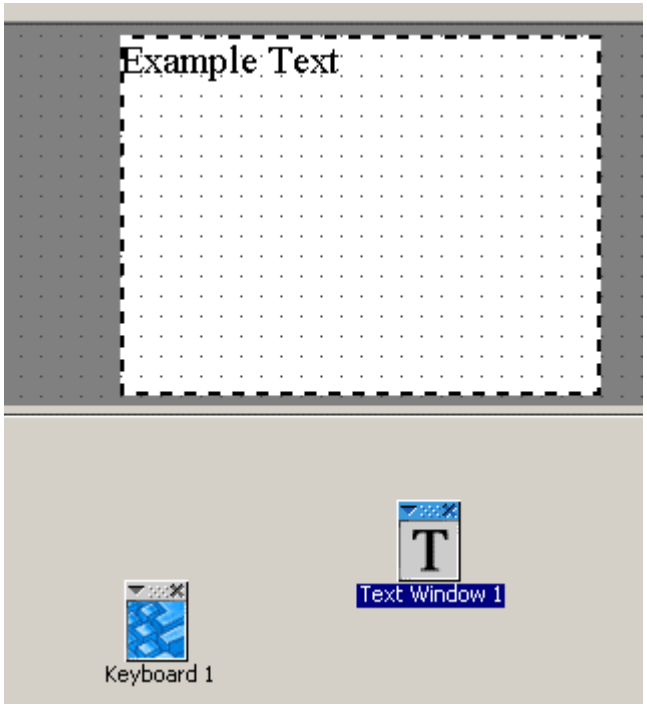

Figure 57. `Text Window` with its *Target Area* resized

Step 8

Now we want to send keyboard events (text) to the `Text Window` object.

Connect:

- 'Keyboard 1', *Out-ports* > *Key Pressed [Text]* to 'Text Window 1', *In-ports* > *Append Text [Text]*

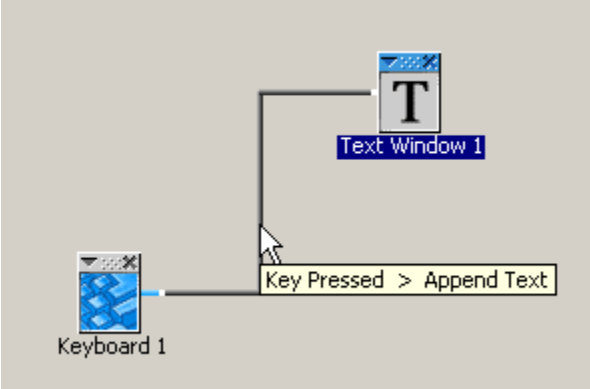

Figure 58. 'Keyboard 1' connected to 'Text Window 1'

OK, done!

Press F9 to preview the presentation. The texture is now replaced with the white stage and the default text (Example Text) that was entered in the `Text Window` (Figure 59).

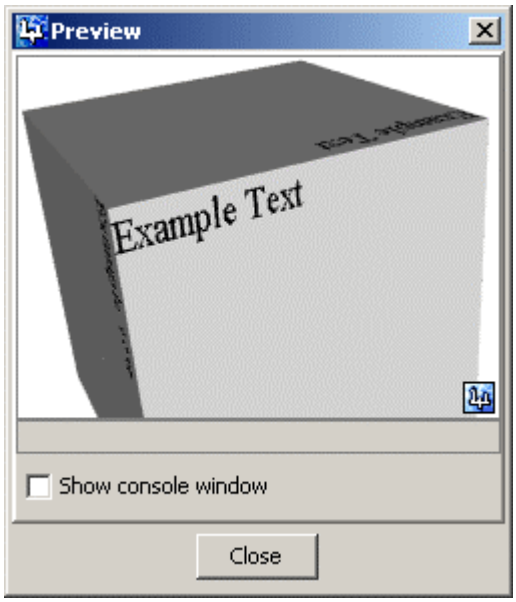

Figure 59. The original texture bitmap replaced with the `Texture` object *Stage*

To interact with the texture you have to change navigation mode to *Interact*. The default hotkey for this is key "4" on your keyboard (not on the numpad however). After you have pressed "4", then you can write text on your keyboard that will be printed on the box (Figure 60).

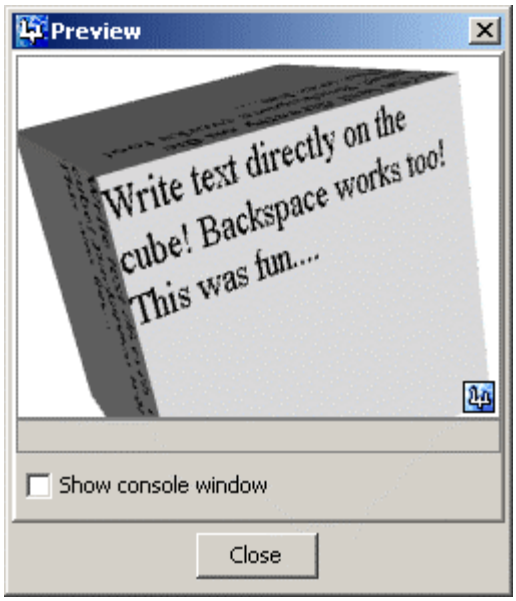

Figure 60. Text written on the box

To change back the navigation mode to Normal, press key "5" on your keyboard.

Note: The navigation hotkeys can be configured in the *Navigation* tab in the `3D Scene` object.

8 Texture Array object

The `Texture Array` object stores an array of images, which can be sent to a `3D Scene` object and replace either a texture or a lightmap.

Normally the `Texture Array` is used to switch between a collection of textures for a specific object. For example, it could be different textures on a sofa or different colors/textures on a mobile phone (Figure 61).

Figure 61. Different textures on a sofa

The `Texture Array` object is very much alike the `Image Array` object, when you drop it in the *Script Area* a dialog which lets you load and sort images appears (Figure 62).

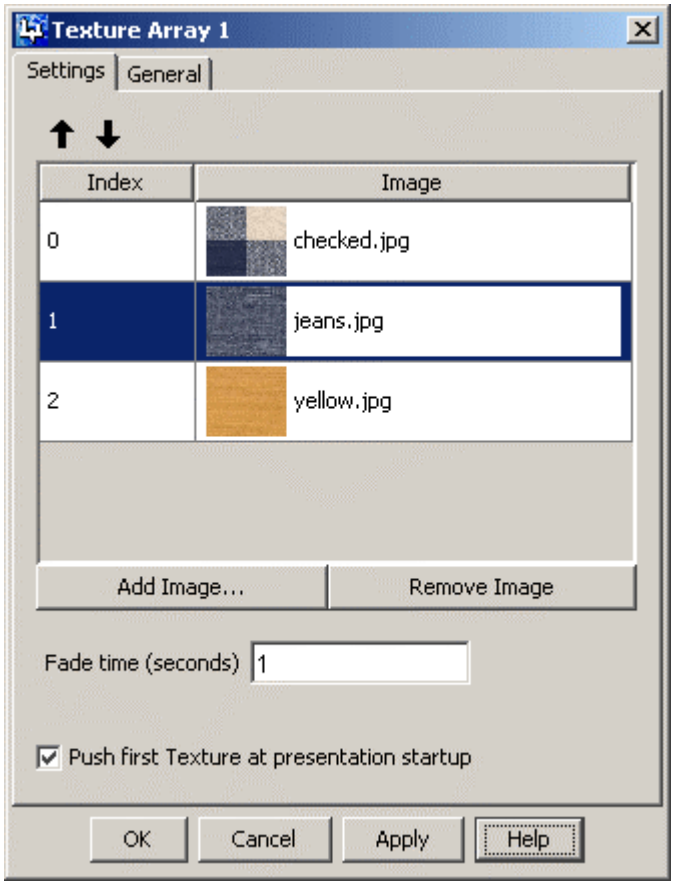

Figure 62. The `Texture Array` settings

Add Image...: (loading dialog)
Adds a new image to the array. Supported formats are JPEG, GIF and PNG. When loaded, use the *Up* and *Down* arrows to sort the images. All texture images have a unique *Index* number, found in the left column, which is used when requesting the texture. To load a new image, double click the small preview image.

Remove Image: (button)
Removes a texture image from the array. Select an image and click the *Remove Image* button to delete it.

Fade time (seconds): (number)
Sets the fade time between the image transitions in seconds. 0 = default.

Push first Texture at presentation startup: (checkbox)
Fires the "Texture Pushed" out-port at presentation startup. Sends the first texture image in the array, i.e. with index 0.

We illustrate the process of adding a collection of textures to an object with an example.

Example

We want to be able to switch between five different colors (textures) on a bicycle computer, using a `Texture Array` object.

1. Insert a `3D Scene` and load the 3D bicycle computer. Load 'cateye.wrl'
2. In the *Appearance* tab, mark the *Show property in-ports* checkbox for the object called "Shell". Click OK to close the dialog. Set the `3D Scene` *Target Area* to the same size as the *Stage* dimension (Alt+S).
3. Insert a `Texture Array` object. When its dialog opens, load the textures 'yellow.jpg', 'blue.jpg', 'green.jpg', 'red.jpg' and 'purple.jpg'. Set the fade time to 1 second. Click OK to close the dialog.
4. Connect: 'Texture Array 1', *Out-ports > Texture Pushed [Texture]* to '3D Scene 1', *In-ports > Objects > 'Shell' > Texture [Texture]*
5. Insert a `JavaScript Link` object. Name the *Link Name* 'ShellColor'. Click OK to close the dialog.
6. Connect: 'JavaScript 1', *Out-ports > Number received from JavaScript [Number]* to 'Texture Array 1', *In-ports > Push Texture [Number]*

Step 1

Insert a `3D Scene` object into a new and empty project, *Objects > 3D > 3D Scene*

Load the 3D bicycle computer. Load 'cateye.wrl' found at:

- *[Path]/WireFusion 3/resources/3d_models/bicycle_computer/*

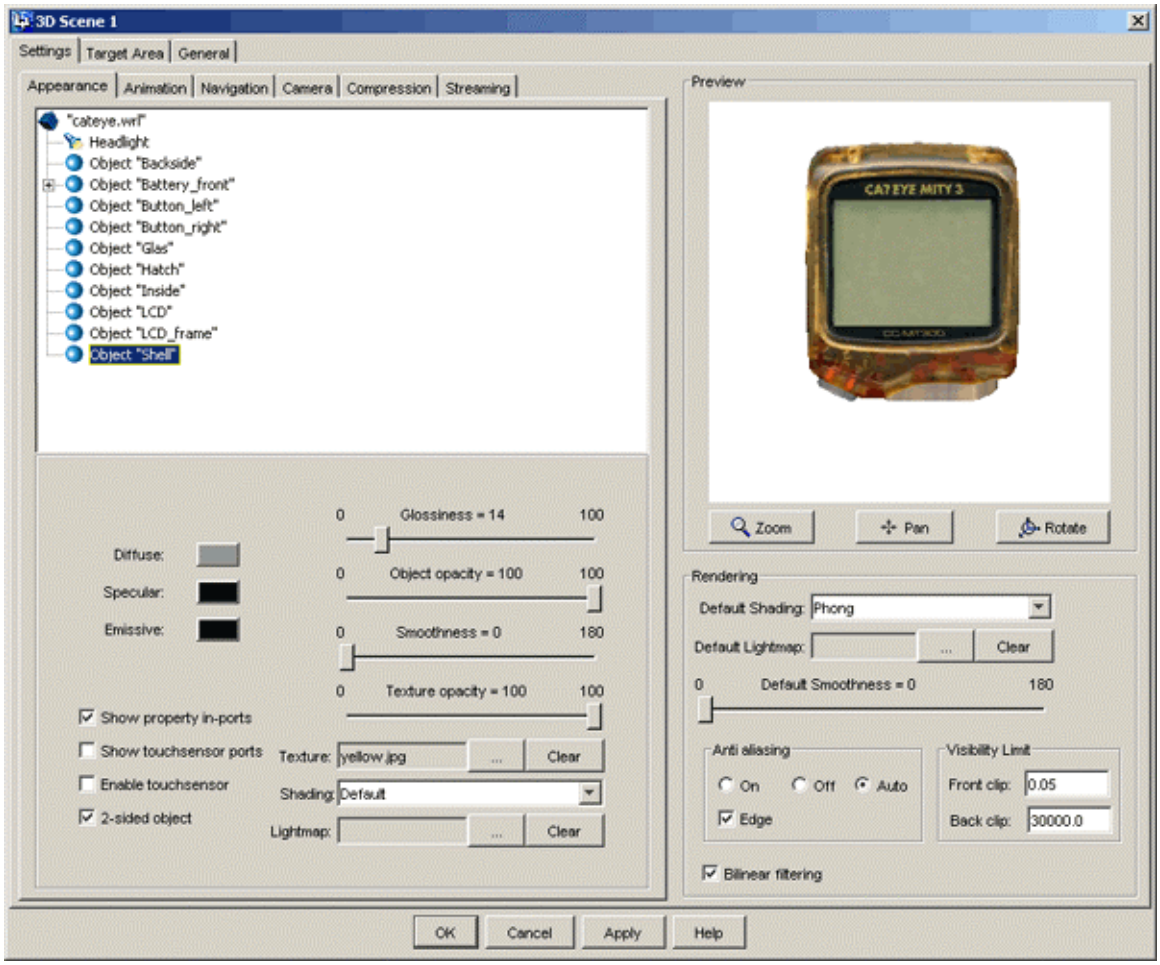

Figure 63. 'cateye.wrl' loaded

Step 2

In the *Appearance* tab, select the 'Shell' object in the list, and then mark the *Show property in-ports* checkbox in order to make its properties visible as in-ports.

Step 3

Insert a `Texture Array` object into the project, *Objects* > *3D* > *Texture Array*

In its Property dialog, add images by clicking the *Add Image...* button. Load the images 'yellow.jpg', 'blue.jpg', 'green.jpg', 'red.jpg' and 'purple.jpg' found at:

- *[Path]/WireFusion 3/resources/3d_models/bicycle_computer/*

Then set the *Fade time* value to 1 (one) second (Figure 64). Click the OK button when ready to close the dialog.

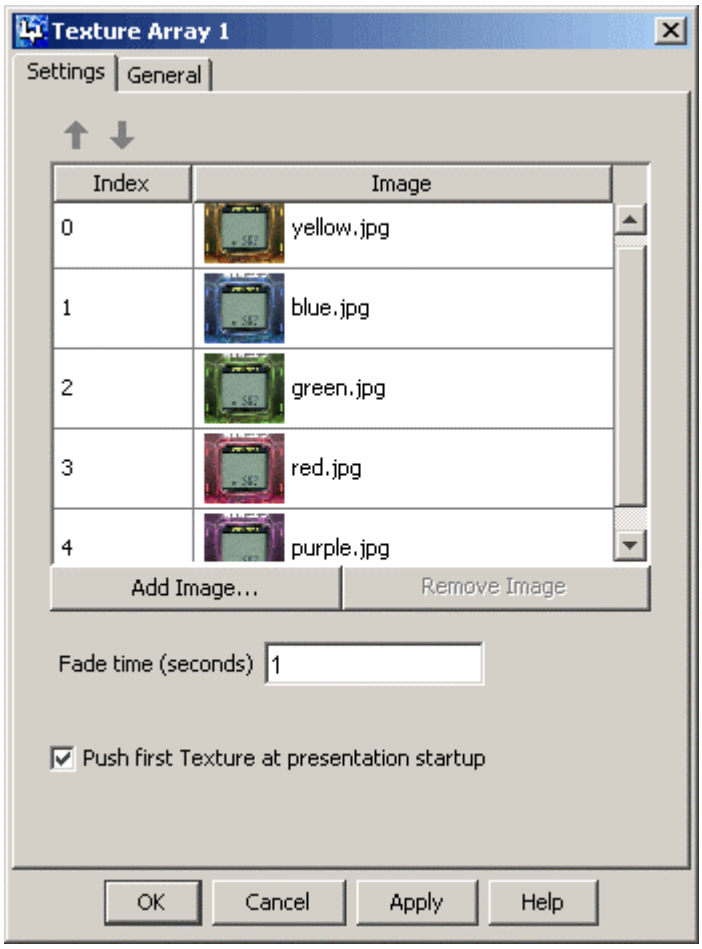

Figure 64. The different color image loaded

Step 4

Replace the default 'Shell' texture with the textures added to the 'Texture Array 1'.

Connect:

- 'Texture Array 1', *Out-ports > Texture Pushed [Texture]* to '3D Scene 1', *In-ports > Objects > 'Shell' > Texture [Texture]* (Figure 65)

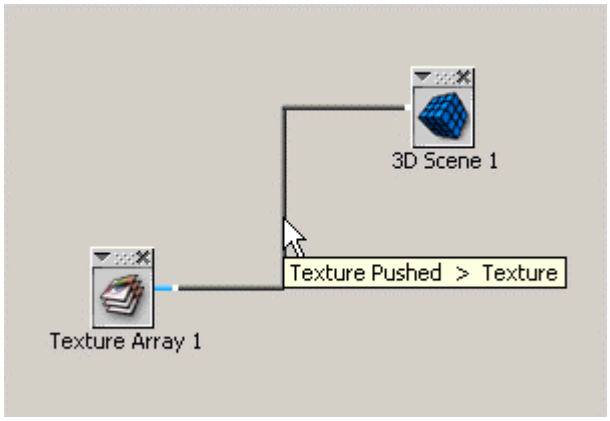

Figure 65. 'Texture Array 1' connected to '3D Scene 1'

Step 5

We will use JavaScript to send index numbers from the HTML page into the presentation in order to switch between the different textures stored in the `Texture Array`.

Insert a `JavaScript Link` object into the project, *Objects* > *Environment* > *JavaScript Link*

In its *Property* dialog, select *JavaScript to Presentation* and change the *Link Name* to 'ShellColor' (Figure 64). Select *Add linked TextField to HTML code*.

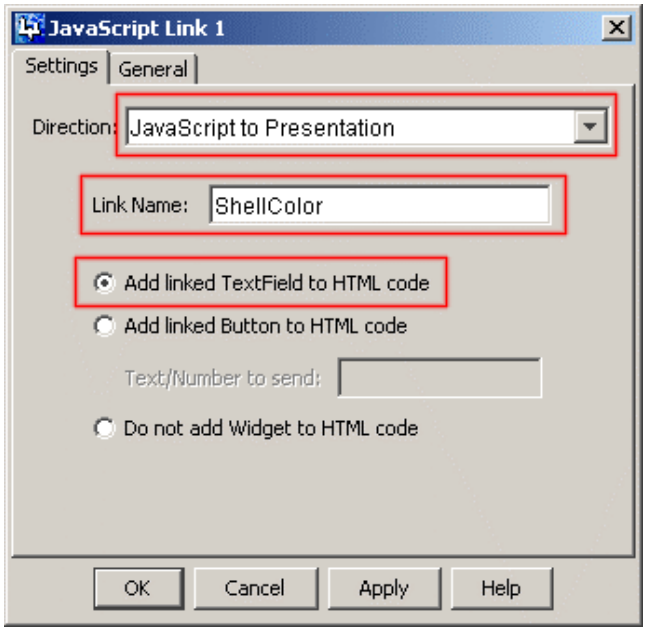

Figure 66. The `JavaScript Link` dialog

Step 6

Send the numbers coming from the HTML page (using JavaScript) to the `Texture Array`.

Connect:

- 'JavaScript Link 1', *Out-ports > Number received from JavaScript [Number]* to 'Texture Array 1', *In-ports > Push Texture [Number]* (Figure 67)

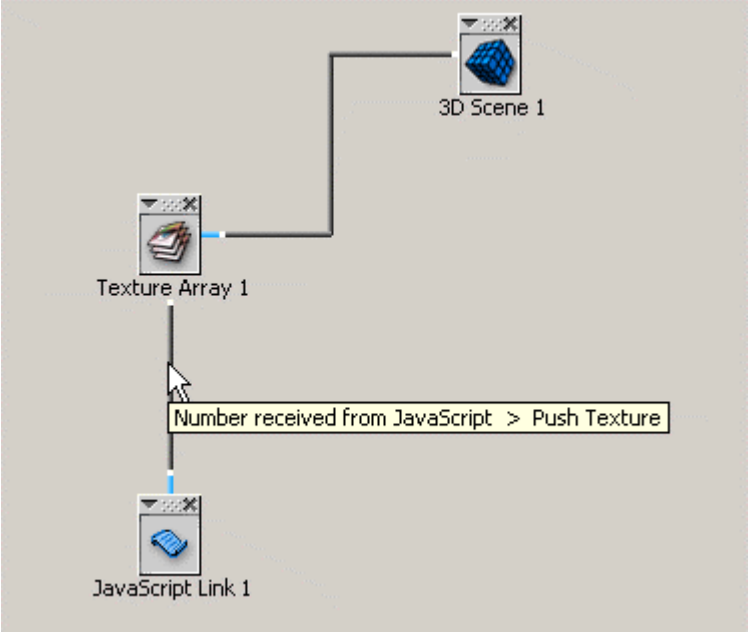

Figure 67. 'JavaScript Link 1' connected to 'Texture Array 1'

OK, done!

Press Ctrl+F9 to preview the presentation in the browser (which is needed as the JavaScript is used).

Enter a number between 0 and 4, which are the index numbers for the images stored in the `Texture Array` (Figure 64), and then click the *Send to ShellColor* button to send the value to the presentation (Figure 68).

Figure 68. Switching textures using `Texture Array` and JavaScript

9 Optimization tips

The performance of your 3D presentation can differ a lot, depending of how you optimize it. There are several aspects to think of while developing and some have already been mention above, but we will enlighten the most important issues here and also give some general tips.

Note: Make sure to read the section *Optimizing your presentation* in the *Working in WireFusion* manual, which takes up some general optimization tips for WireFusion presentations.

In the creation process (in your 3D authoring tool):

- Try to use as few polygons in your models as possible, to ensure small 3D files and fast performance. A polygon reducer can be used to optimize the number of polygons.

- If a high polygon object is still required, use the *Level of Detail* (LOD) function. In *3ds max* you'll find the LOD function under Helpers > VRML97. Read the *3ds max* manual for instructions.

- Try to use as few light sources as possible (or none), as the shading has to be calculated for each light and this consumes CPU. Use the *Lightmap* shading option instead for complicated lighting or use “baked textures” (a method where the 3D artist renders out a bitmap built upon the desired lights).

- Use only JPEG, GIF or PNG images for textures. Try to have them optimized in file size and dimension before exporting to VRML (it is possible to change textures from WF-3D if needed). Use for example Adobe ImageReady for image optimization, which has a nice *Save for Web* feature.

- Position your model in the world center for proper rotation in 'WD-3D' and add at least one Camera.

- Make sure to create your model so it’s oriented correctly in the 3D space, i.e. make sure to have a top view of your model in the *Top viewport*, a front view in the *Front viewport* etc. This to ensure correct navigation in WireFusion.

- When exporting to VRML97, don't export to *Primitives* (not supported in WF-3D) and don't generate *Normals* (creates unnecessarily large files).

In WF-3D:

- Make sure you are ready in your 3D authoring tool before importing and doing too much work in the `3D Scene` object, as you can't reload a VRML file into the `3D Scene` object.
- Use *Smoothness* to give your low polygon models a smoother look.
- In most cases the anti-aliasing option *Auto* is best suited.
- Use *Edge anti-aliasing* for smoother edges.
- Use *Bilinear filtering* for smooth textures and reflections.
- In the *Navigation* tab, adjust the zoom, rotation and pan speed for a better user experience.
- In the *Camera* tab, add camera restrictions for easier navigation and hence better user experience.
- In the *Streaming* tab, enable streaming to have the presentation started quicker.