

**SONY**

Community Place Conductor 2.0  
User's Manual

---

Copyright (c) 1998 Sony Corporation, All rights reserved  
Community Place is a trademark of Sony Corporation.

The use and reproduction of this software is permitted only within accordance with the terms of the Licence Agreement. The contents of this publication may be modified at any time without notice, and represent no affirmation to product specification or pursuit of obligations by Sony Corporation.

No part of this publication may be reproduced, transmitted, transcribed, stored, or translated to any other languages in any form or by any means, without prior written permission of Sony Corporation.

Windows, Windows NT are trademarks or registered trademarks of Microsoft Corporation in the United States and other countries. Netscape Navigator is the trademark of Netscape Communications Corporation, USA. Other product and company names mentioned herein may be the trademarks or registered trademarks of their respective owners.

# Contents

<b>Contents</b> .....	<b>i</b>
<b>Community Place Conductor 2.0</b> .....	<b>1</b>
Installing Conductor .....	2
Uninstalling Conductor .....	3
Starting or Quitting Conductor .....	3
Using Conductor .....	4
Reference .....	5
<b>Main Window</b> .....	<b>6</b>
File menu .....	6
New .....	6
Open, Reopen .....	6
Save .....	7
Save As .....	7
Close .....	7
Exit .....	7
Edit menu .....	7
Undo .....	7
Redo .....	7
Cut .....	7
Copy .....	8
Paste .....	8
Delete .....	8
Keyframe .....	8
View menu .....	8
Run menu .....	8
Run .....	8
Pause .....	8
Stop .....	9
Script menu .....	9
Expert .....	9
Build .....	9
Rebuild .....	9
Options menu .....	9
Environment .....	9
Help menu .....	12
VRML97 Spec .....	12

Show CLASSPATH .....	12
About .....	13
Changing the mouse mode .....	14
The Tool Bar .....	14
<b>3D Perspective View Window .....</b>	<b>17</b>
Reading a VRML file .....	17
Selecting an object .....	18
Displaying your eye position/direction .....	18
Creating an object .....	18
Walk through a world .....	19
Moving an object .....	20
Rotating an object .....	20
Scaling an object .....	21
Pop-up menu .....	21
<b>Appearance Window .....</b>	<b>24</b>
Editing the Material node .....	24
Editing the Texture node .....	25
Editing the TextureTransform node .....	26
<b>Attribute Window .....</b>	<b>27</b>
Editing a DEF name .....	28
Editing attributes .....	28
Wizard tab .....	28
Editing nodes within PROTO .....	29
Pop-up menu .....	30
Editing the Script node .....	31
<b>Keyframe Editor Window .....</b>	<b>32</b>
Starting or Quitting the Keyframe Editor .....	32
Selecting the Source .....	33
Enlarging/reducing or scrolling the display .....	33
Specifying the fraction .....	33
Locking the display .....	33
Seeing how animation works on the screen .....	33
Loop checkbox .....	34
Selecting multiple Keyframes .....	34
Pop-up menu .....	34
Keyframe operations (Scaling and Smoothing) .....	35
<b>Library Window .....</b>	<b>38</b>

The types of library .....	38
Dragging and Dropping .....	38
Starting the related application .....	39
Displaying Help in the Script library .....	39
Changing the Library folder .....	40
Customizing an icon .....	40
Formats supported in each library .....	41
Adding objects to a library .....	41
<b>Message Window .....</b>	<b>43</b>
Warning message .....	43
Message from Java compiler .....	43
<b>Multi User Window .....</b>	<b>44</b>
Creating a Multi User World .....	44
Configuring Multi User Server .....	44
Specifying the Avatar .....	45
<b>Parallel View Window .....</b>	<b>49</b>
Displaying and changing your eye position/direction .....	49
Pop-up menu .....	50
Creating an object .....	51
Walk through a virtual world .....	51
Moving an object .....	52
Rotating an object .....	52
Scaling an object .....	53
<b>Route Window .....</b>	<b>54</b>
Text display mode: .....	54
Adding a route .....	55
Deleting a route .....	55
Visual display mode: .....	55
Automatically creating and modifying a DEF name .....	57
Pop-up menu .....	58
<b>Scene Graph Window .....</b>	<b>59</b>
Switching the current scope .....	59
Switching the PROTO scope .....	60
Selecting a node .....	61
Displaying the Help information .....	61
Pop-up menu .....	61
<b>Script Expert Window .....</b>	<b>66</b>

Using Script Expert .....	66
<b>Text Editor Window .....</b>	<b>68</b>
Starting the Text Editor .....	68
Quitting Text Editor .....	68
Pop-up menu .....	69
<b>Texture Mapping Window .....</b>	<b>71</b>
Color sheet .....	71
Texture Coordinate sheet .....	73
Color Mode dialog .....	75
Change Texture Mode dialog .....	78
Texture Mapping Projection dialog .....	80
<b>Object Customize Wizard Operating Instructions .....</b>	<b>82</b>
Modifying VRML objects .....	82
Entering necessary information in the Wizard configuration file .....	83
<b>Script Reference .....</b>	<b>89</b>
Introduction .....	89

## **Community Place Conductor 2.0**

Community Place Conductor ("Conductor" in the following pages) is the authoring tool that conforms to VRML 2.0. Conductor 2.0 has the following features:

- Conforms to VRML 2.0.
- Drag and drop to add a node.
- Enables you to specify values for node attributes.
- Enables you to specify textures for sound and objects.
- Enables you to switch between the Execution and Edit modes.
- Conforms to Java (JRE1.1.3, JDK1.1.3, or later).
- Enables you to edit independently in each scope by switching the current scope.
- The Script Expert function to automatically create a script node.
- Enables you to add/delete routes while displaying it on the screen.
- Enables you to edit keyframes.
- Enables you to edit PROTO.
- Supports USE.

This manual describes each function of Conductor. For the information of how to create a world with Conductor, see Tutorial .

## ■ Installing Conductor

Before starting Conductor, check to see if your system meets the following requirements.

table 1: Required Environment

	<b>Recommended Environment</b>
Hardware	PC/AT-compatible machines operated under Windows 95/Windows NT 4.0 or later
CPU	Pentium 90MHz or higher
Memory	32MB or more
Free disk space	20MB or more (40MB required at the time of installation)
Display resolution	800x600 pixels or higher
Display colors	65536 colors or more
Software	Netscape Navigator for Windows (Ver. 3.0 or later) or Microsoft Internet Explorer (Ver. 3.0 or later) *
JAVA**	JDK1.1.3 or later
Sound	Sound board + a speaker or headphones

\* Necessary when displaying Help or Anchor.

\*\* Necessary when creating contents in Java.

**Note:** Conductor may not operate properly with some hardware. For details, see "Restrictions."

## ■ Uninstalling Conductor

You can easily and safely uninstall Conductor in the following procedure:

1. Click the Start button.
2. Open Program, and select Community Place Conductor.
3. Select Uninstall from the menu.

**Note:** A part of the temporary files or the files you created after installing Conductor may not be deleted by uninstalling Conductor. In such a case, delete all the Conductor directories left. Before re-installing Conductor, make sure that all the Conductor directories are completely deleted.

## ■ Starting or Quitting Conductor

### ● Starting Conductor

Choose from the Start menu in the order of Program, Community Place Conductor, and Community Place Conductor. To newly create a world, choose the File/New menu. To load an existing VRML 2.0 file, choose from the File/Open menu.

**Note:** The Conductor is not compatible with the VRML 1.0 files. Convert all the files you use to the VRML 2.0 format with [vrml1to2](http://vs.sony.co.jp/Download/vrml1to2.htm) (<http://vs.sony.co.jp/Download/vrml1to2.htm>).

### ● Quitting Conductor

Choose File/Exit from the menu. If you are working on the file, you will be prompted to save the changes.

## ■ Using Conductor

Main window (Manages the overall procedure for using Conductor.)

3D Perspective View window (Displays objects in the perspective projection mode.)

Appearance window (Specifies the color and texture of an object.)

Attribute window (Specifies the attribute of a node.)

Keyframe Editor window (Edits Interpolator.)

Library window (Manages the library.)

Message window (Displays various messages.)

Multi User Window (Specifies the multi user world.)

Parallel View window (Displays objects in the parallel projection mode.)

Route window (Specifies route for events.)

Scene Graph window (Displays the tree structure of a world.)

Script Expert window (Creates a Script model.)

Text Editor window (Used to edit a Java file.)

Texture Mapping window (Edits Texture Mapping.)

Each window can be displayed or closed when necessary.

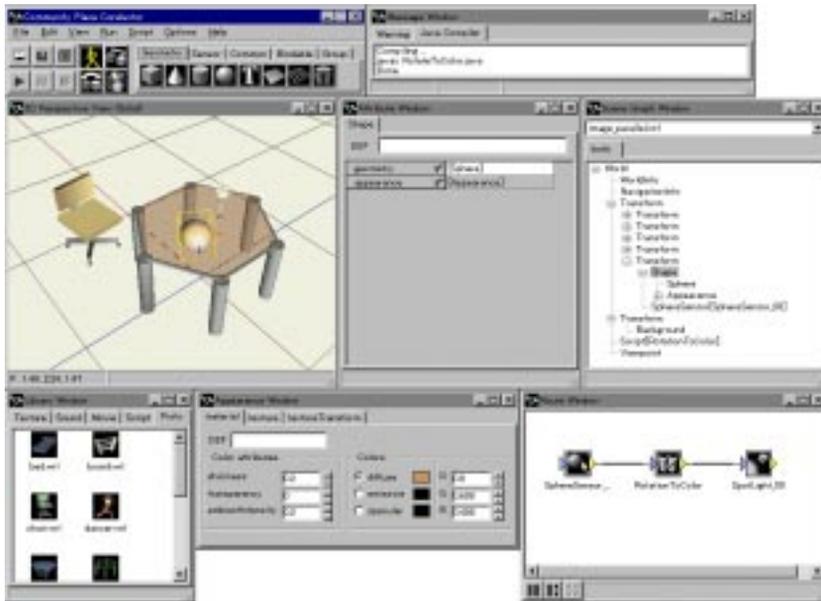


Figure 1: Using Conductor

## ■ Reference

VRML97 Specifications  
<http://vag.vrml.org/VRML97>

Virtual Society on the Web site  
<http://vs.sony.co.jp>

## Main Window

This window allows you to newly create, read, store, and run a world, change the mouse mode, switch to the keyframe edit mode, and add/delete objects.

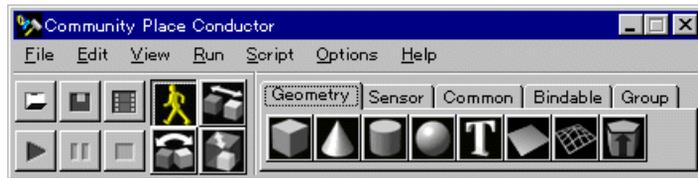


Figure 2: Main window

### ■ File menu

- **New**

Click New on the File menu. This opens a new file and displays windows, such as a 3D Perspective Window (to be described "3D View window" in the following pages) to operate a world.

- **Open, Reopen**

To load a world from a file, press the Open button  the File menu. The File Open dialog box appears on the screen. Specify the name of the file you want to open. To load the file previously loaded, click Reopen on the File menu. The list of all the files appears on the screen. You can select the file you want to load out of the list.

The following are other methods to opening a file. Shortcuts files are not supported in any of the cases below.

- Drop a VRML file on the Conductor icon.
- Execute Conductor and drop a VRML file on the Main Window.
- Load from command line ( ex: cpcondct foo.wrl )

**Note:** The Conductor is not compatible with the VRML 1.0 files. Convert all the files you use to the VRML 2.0 format with [vrml1to2](http://vs.sony.co.jp/Download/vrml1to2.htm) (<http://vs.sony.co.jp/Download/vrml1to2.htm>).

- **Save**

Saves the world to a file. It is also done by pressing the Save button  .

- **Save As**

To save the file under a new name, click Save As on the File menu. The File Save Dialog Box appears on the screen. Type the name under which the file is to be saved.

- **Close**

To close a world, click Close on the File menu.

- **Exit**

To quit Conductor, click Exit on the File menu. If you are working on the file, you will be prompted to save the changes when you click Close or Exit. To save the changes, click Yes.

**Note:** Do not save files in a temporary directory ("C:\Program Files\Sony\Community Place Conductor\tmp" by default). All the files contained within the temporary directory will be deleted at the time of starting or quitting Conductor.

## ■ Edit menu

- **Undo**

If you wish to undo your work, click Undo on the Edit menu (or simultaneously press Ctrl and z keys).

**Note:** You can undo your work as many times as you want, but there are some functions that you cannot undo.

- **Redo**

If you wish to bring back the work that you have cancelled by using Undo, click Redo on the Edit menu (or simultaneously press Shift and Ctrl and z). Redo is only available right after using Undo.

- **Cut**

To delete a node, select the node you want to delete, then click Cut on the Edit menu (or simultaneously press Ctrl and x keys). You can select the node to delete from within the 2D/3D View Window or the Scene Graph Window. You can also paste the node you have cut.

- **Copy**

To copy a node, select the node you want to copy, then click Copy on the Edit menu (or simultaneously press Ctrl and x keys). You can also paste the node you have copied.

- **Paste**

If a node has been loaded onto the Cut or Copy buffer, you can paste it. Only the currently selected node can be pasted. If it cannot be pasted (because of inconsistent node type), a warning message will appear.

**Note:** The Inline node cannot be copied.

- **Delete**

Deletes a node.

- **Keyframe**

Opens the Keyframe Editor window to edit the Interpolator node. Please refer to the "Keyframe Editor" section of the manual for more information.

## ■ View menu

This menu allows you to display the Conductor windows. The menu items of the window currently displayed are checkmarked.

## ■ Run menu

- **Run**

To check to see how the world works on the screen, click Run on the View menu or press the Play button 

**Note:** While running, only the Main window, 2D/3D View window, Message window, and Java Console window can be used.

- **Pause**

To temporarily stop the world, click Pause on the View menu or press the Pause button . To resume playing back the world, press the Pause or Play button again.

- **Stop**

To stop playing the world and return to edit mode, press the Stop button  .

## ■ Script menu

- **Expert**

Activates the Script Expert Window that lets you create a Script model. For more information, please refer to the "Script Expert Window" section of this manual.

- **Build**

Compiles only the modified script files.

- **Rebuild**

Recompiles all the script files.

**Note:** Script menu can only be used when JDK 1.1.x is installed. It is not compatible with JDK 1.2.x.

## ■ Options menu

- **Environment**

The following Environmental settings dialog boxes (Figure 3 ) will appear.

### General page

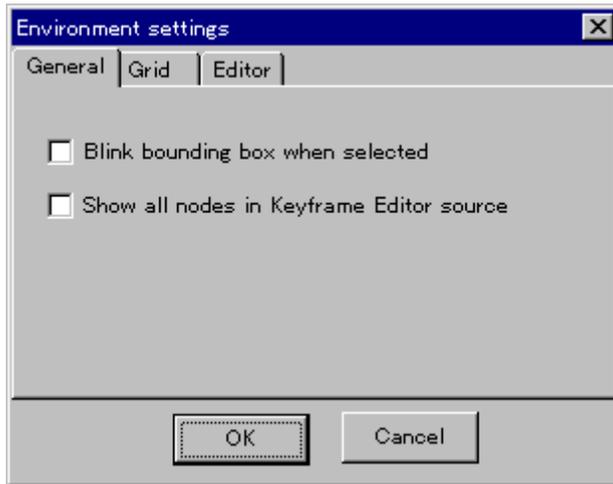


Figure 3: General page

#### Blink bounding box when selected:

Check this to blink the bounding box around the selected object.

#### Show all nodes in Keyframe Editor source:

Check this to display all the specifiable nodes within the Source selection combo box within the Keyframe Editor Window.

## Grid page

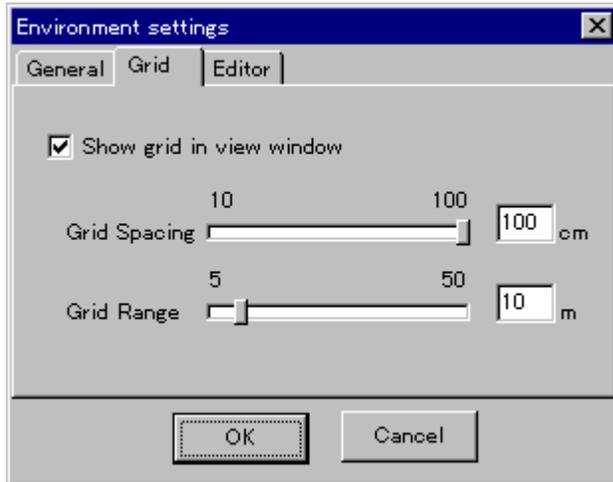


Figure 4: Grid page

## Show grid in view window:

Check this to show grid lines in the 2D/3D View Window. When checked, interval and area of the grid can be configured using the slider bar or entering values in the edit box.

Editor page

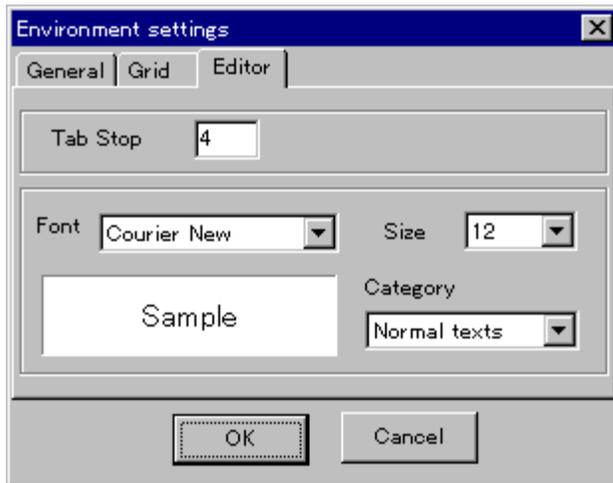


Figure 5: Editor page

In this page, number of tab stops, font, size, colors for each Category of text (normal text, keyword, comment, background colors) can be configured.

## ■ Help menu

### ● VRML97 Spec

Displays the specifications of VRML97 within the HTML browser.

Note:1) Netscape Navigator 3.0 or later or Microsoft Internet Explorer 3.0 or later is required to display the specifications.

Note:2) For the copyrights of VRML97 online manual, please read the attached "vrml97copyright.html".

Note:3) If Help menu is selected while Netscape Navigator or Microsoft Internet Explorer isn't running, it will open the browser without showing the Help file. To view Help, select the Help menu once again.

### ● Show CLASSPATH

Displays the class path information when the Sun JRE 1.1.x or JDK 1.1.x has been installed.

Note: The class path information will not be displayed when JRE or JDK is not detected.

Note: You will need JDK to compile a script. Please install version 1.1.x for JRE and JDK. JDK 1.2.x cannot be used.

- **About**

Displays the About dialog box shown below. This dialog shows the version number of Conductor, the information about the Java that has been recognized by Conductor.

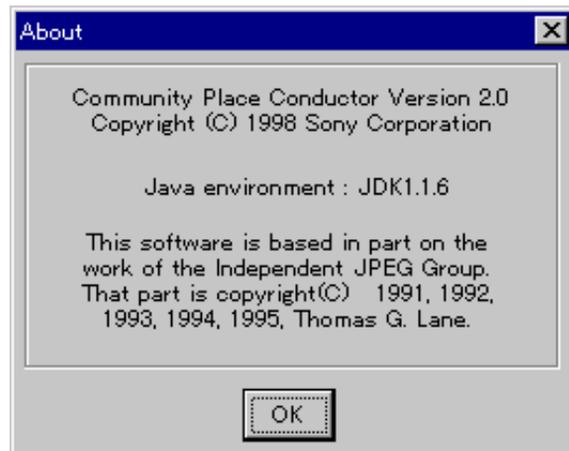


Figure 6: About dialog

## ■ Changing the mouse mode

Conductor provides you with the following mouse modes to select from. Switch by clicking one of the corresponding buttons:

table 2: Mouse Modes

 Walk	Used to move your eye position/direction within the 2D/3D View window.
 Move	Used to move an object.
 Rotate	Used to rotate an object.
 Scale	Used to change the scale of an object.

Only a selected object can be moved, rotated, or scaled. You can select an object from within the 3D View Window or the Scene Graph Window. Selecting an object, the box enclosing it will start blinking (when the blinking option is selected).

## ■ The Tool Bar

To add a node such as geometry or sensor to a world, choose the node that you want to add on the Tool Bar (figure 4), and then click on the 3D View window. (Please refer to the tutorial for usage)



Figure 7: Tool bar

table 3: Geometry

Button	Node
 Box	Cube
 Cone	Cone
 Cylinder	Cylinder
 Sphere	Sphere
 Text	Text
 IndexedFaceSet	Polygon (a quadrilateral by default)
 ElevationGrid	Land surface (a plane by default)
 Extrusion	A 3D figure extruded from a 2D figure

table 4: Sensor

Button	Node
 ProximitySensor	Sensor to detect a viewpoint in a region
 TimeSensor	Sensor of time
 TouchSensor	Sensor to detect a hit on the object (for example, a mouse click)
 PlaneSensor	Sensor to detect the dragged distance as amount of movement on the xy-plane on the local coordinate system
 SphereSensor	Sensor to detect the dragged distance as amount of rotation.
 VisibilitySensor	Sensor to detect whether the region is included in the visible range

table 5: Common

Button	Node
 DirectionalLight	Parallel light source
 PointLight	Point light source
 SpotLight	Spotlight
 WorldInfo	Defines world information

table 6: Bindables

Button	Node
 Fog	Sets a fog
 Viewpoint	Adds a viewpoint
 Background	Sets a background
 NavigationInfo	Sets a navigation information

table 7: Group

Button	Node
 Anchor	Links to an URL
 Billboard	Automatically rotates an object to keep a specific direction against the viewpoint.
 Collision	Detects collision of the viewpoint and an object
 Group	Group
 Transform	Defines the local coordinate system
 Inline	Reads data from another file

## 3D Perspective View Window

The 3D View window displays all the nodes as the surface models in the 3D perspective projection view.

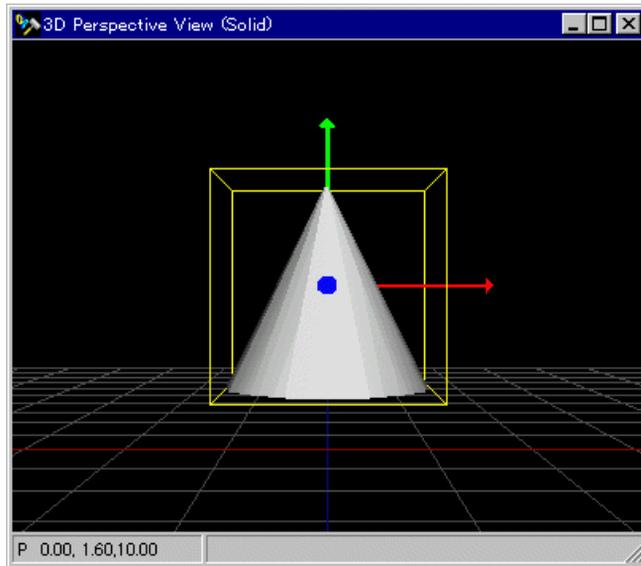


Figure 8: 3D View Window

### ■ Reading a VRML file

You can add an object to a scene by dropping a VRML file into the 3D View window.

**Note:** There are some restrictions on dropping a file to the 3D View window. Please refer to the restriction page (doc/restrict/restrict-e.txt) for more information.

## ■ Selecting an object

If you select an object, the box enclosing the object starts blinking to indicate that it is currently selected. The local coordinate for the selected object will also start blinking. (Red represents the local Y-axis, blue the Z-axis, and green the X-axis. The arrow marks indicate the positive directions of the axes.)

## ■ Displaying your eye position/direction

The values of your eye position/direction currently set are indicated on the sub-panel at the lower-left corner. You can change the display as follows by clicking on this panel:

table 8: Parameters shown on the eye position panel

Parameters	Variables	Example
Eye position in world coordinates	(P x, y, z)	P 0.00, 0.00, 4.16
Distance from the eye position to the center position of the selected node	(D d)	D 8.83
Eye direction in world coordinates	(V x, y, z)	V -0.00, -0.00, -1.00

## ■ Creating an object

You can create an object by using the buttons on the Main Window or dragging and dropping icons from the Library Window. The object will be created at the center position of the 3D View Window, just in front of the eye position regardless of where the mouse button was released.

### • To create a node from the Main Window>

1. Select the library you want to create an object out of from the Primitive bar in the Main Window.
2. Press the left mouse button on the object you want to create; the mouse button will be highlighted and kept caved-in.
3. Drag the mouse pointer onto the 3D View Window.

4. Left-click here again to display the selected object.

Note:1) Objects cannot be added to some nodes. When creating an object, please check which node is selected in the Scene Graph Window.

Note:2) Nodes that are created with Transform, such as Geometries and Light Sources, will be shown on the 3D View Window, but invisible nodes such as Sensors and Scripts will not be shown. Created nodes can be viewed on the Scene Graph Window.

## ■ Walk through a world

Before walk through the 3D View, select the Walk  in the Conductor window.

You can move in every direction basically by dragging with the left mouse button pressed or by using the arrow keys on the keyboard.

table 9: Walk

Movement	Modifier keys	Mouse	Arrow key	Eye position	Eye direction
Move forward		Drag upward	up	moves	does not move
Move backward		Drag downward	down	moves	does not move
Turn right		Drag rightward	right	does not move	moves
Turn left		Drag leftward	left	does not move	moves
Look up	Shift key	Drag upward	up	does not move	moves
Look down	Shift key	Drag downward	down	does not move	moves
Move upward	Shift + Ctrl keys	Drag upward	up	moves(*)	does not move
Move downward	Shift + Ctrl keys	Drag downward	down	moves(*)	does not move

table 9: Walk

Movement	Modifier keys	Mouse	Arrow key	Eye position	Eye direction
Move rightward	Shift + Ctrl keys	Drag rightward	right	moves(#)	does not move
Move leftward	Shift + Ctrl keys	Drag leftward	left	moves(#)	does not move

Note 1: <moves (\*)> indicates that the eye position will move only in the direction of Y-axis.

Note 2: <moves (#)> indicates that the eye position will not move in the direction of Y-axis, but will move in the directions of X and Z-axes only on the surface vertical to the eye direction.

## ■ Moving an object

Before moving an object in the 3D View mode, select the  Move mouse mode in the Main window.

You can move only the currently selected object by dragging it with the left mouse button. At the position where you want to place the object, release the left mouse button. The object can be moved only on the surface that goes through the original center position of an object and is vertical to the eye direction.

Note: You may not be able to select an object when it is located too far away.

## ■ Rotating an object

Before rotating an object in the 3D View mode, select the  Rotate mouse mode in the Main Window.

You can rotate only the currently selected object by dragging the mouse around it with the left button pressed. The object can be rotated only on the axis that goes through the center position of the object. To stop rotating the object, release the left mouse button.

## ■ Scaling an object

Before scaling an object in the 3D View mode, select the  Scale mouse mode in the Main Window.

You can scale only the currently selected object by dragging it with the left mouse button pressed. If you have satisfactorily enlarged or reduced the object, release the left mouse button. An object will be scaled with its center position on the central axis.

**Note:** With the mouse, you can only scale the object *proportionally* in every directions.

## ■ Pop-up menu

The pop-up menu can be displayed by right-clicking on the 3D View.

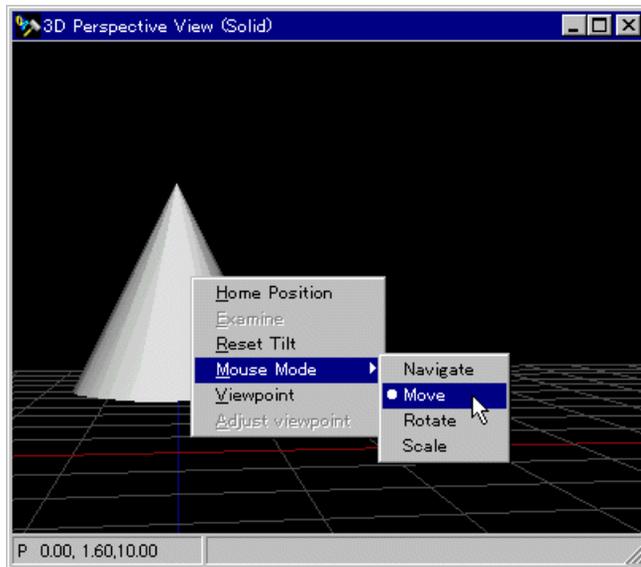


Figure 9: Pop-up menu in the 3D View

The Pop-up menu provides the following sub-menus:

- **Home Position**

Used to locate your position when you get lost or when you want to return to the start position.

**Note:** The Home Position corresponds to that specified in the Viewpoint node within the VRML file or that specified by default (0, 0, 0).

- **Examine**

Used to monitor ws around the object currently selected. This menu is effective only in the 3D View mode. You can use the menu in the following procedure:

1. Specify the Walk  mouse mode in the Conductor window.
2. Select the object you want to monitor.
3. Select Examine ON from the pop-up menu.

**Note 1:** Examine On can be selected only when a node that can be monitored is selected.

With Examine ON, the eye direction changes so that the object you want to monitor appears at the center position of the window.

**Note 2:** The distance from the object never changes since the eye position does not change.

4. Dragging the mouse with its left button pressed around the object will change your eye position/direction. This enables you to monitor the object from every angle.

**Note 3:** With Examine ON, 3D View normally does not allow you to walk through a world. The Parallel View allows you only to walk.

**Note 4:** In the Examine mode, you cannot select other objects. To select other objects, exit the Examine mode.

To quit monitoring the object, select Examine OFF from the pop-up menu. With Examine OFF, the background color of the eye position indicator panel returns to gray, and your eye position/direction last selected will be restored.

- **Reset Tilt**

Restores your original view direction (to the parallel).

- **Mouse Mode**

Enables you to select the mouse mode (Walk, Move, Rotate, and Scale).

- **Viewpoint**

Displays all the Viewpoint nodes existing within the world. You can swiftly move to your desired location by selecting one of the displayed Viewpoints.

Note: The Viewpoint menu shows the strings specified in description. Viewpoint is not set in the newly created world.

- **Adjust Viewpoint**

This menu is available only when Viewpoint is selected. The currently selected eye position/direction can be reflected in Viewpoint.

## Appearance Window

Used to specify the color and texture of the object currently selected. This window enables you to edit the Material node, ImageTexture node, TextureTransform node, which are included in the Appearance node. For the details of how to use each node, see "Help/VRML97 Specifications."

### ■ Editing the Material node

Click the material tab to display the Material sheet on which you can specify the attributes included in the Material node.

**Note:** Press the Enter key after typing the name of a node within the DEF name edit box.

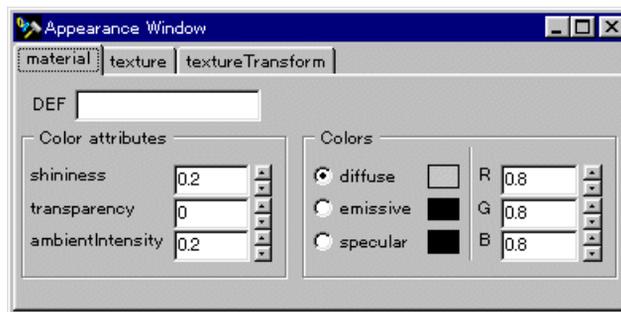


Figure 10: Selecting the material tab

### • Specifying colors with the ColorMap dialog

You can specify a color by using the ColorMap dialog. You can activate the ColorMap dialog by clicking the rectangles on the right of each color (diffuse, emissive, and specular). Dragging the mouse on the color panel on the left will change the Hue (H) and Shade (S). Dragging the slide bar will change the Brightness (V).

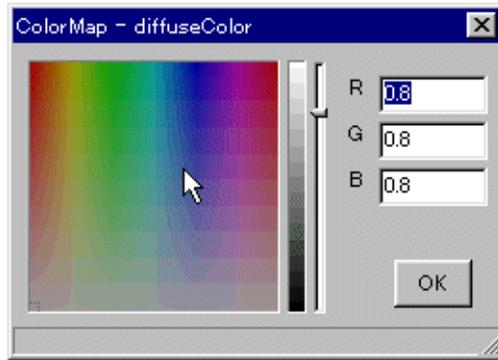


Figure 11: ColorMap dialog

## ■ Editing the Texture node

Click the texture tab to display the Texture page on which you can edit the Image-Texture attributes and the DEF name.

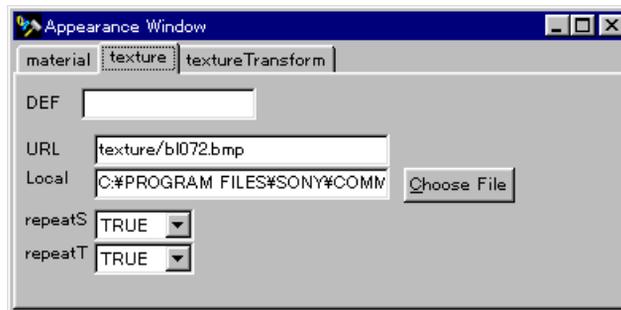


Figure 12: Selecting the texture tab

Note:1) The MovieTexture and PixelTexture nodes are not supported.

Note:2) Press the Enter key after typing the name of a node within the DEF name edit box.

Local indicates the actual location of the texture file. The address in Local is usually the same as that in URL (relative URL must be converted to the absolute URL). Immediately after the file is dropped out of the Library Window, Local indicates the Library directory.

Save or Save As copies the file to the location specified in URL, and thus both URL and Local refer to the same file. (Local can be changed only by using Choose File.)

## ■ Editing the TextureTransform node

Click the textureTransform tab to display the textureTransform page on which you can edit the textureTransform attributes and the DEF name.

On mapping a texture, set the parameters here to adjust the translation, rotation, and the scale.

**Note:** Press the Enter key after typing the name of a node within the DEF name edit box.

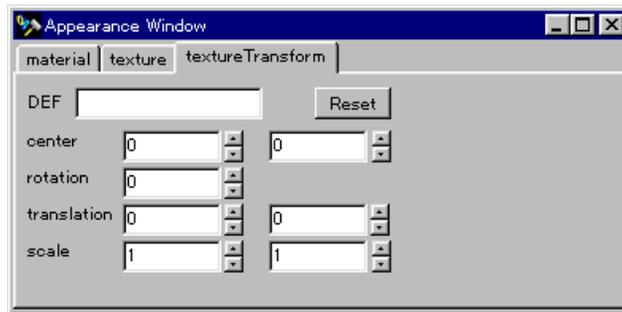


Figure 13: Selecting textureTransform

## Attribute Window

You can specify the attribute and DEF name of the node currently selected in this window. The status bar at the bottom of the window shows the information about the field that the mouse pointer is on.

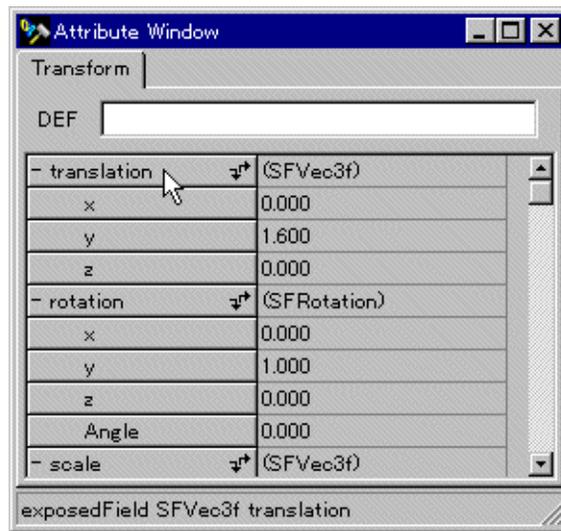


Figure 14: Attribute Window

The icon beside the attribute name indicates the kind of field for the item.

table 10: Fields and Icons

Kind of fields	Icon
field	-
exposedField	↔
eventIn	↵
eventOut	↶

Note: eventIn, eventOut is shown only on Script nodes and Proto instances.

## ■ Editing a DEF name

When routing events or creating a node with USE, use this function. Type the name of a node within the DEF name edit box, then press the Enter key.

## ■ Editing attributes

There are 4 types of attributes, all conforming to the Node field types of VRML 2.0: Value, Character string, Color, and BOOL.

To edit the BOOL type attributes, change the values by double-clicking the area indicated with "TRUE" or "FALSE."

If "(SFCOLOR)" is displayed in the edit area, its attribute is the Color type. Double-click the edit area, and the ColorMap dialog box will appear. Select your desired color and press the OK button.

To apply the newly specified values to the world you are in, press the Enter key or the Up/Down arrow key.

**Note:** If "+" is displayed at the right side of an item, you can display an edit field by double-clicking it. To erase the edit box, click it again.

## ■ Wizard tab

Conductor 2.0 provides you with the Object Customize Wizard which enables you to edit attributes of objects more easily than you do using the former Attribute Window. If the selected object conforms to Wizard, the Wizard tab appears in the Attribute Window. User friendly GUI(Graphical User Interface) will help you edit the attributes. The PROTO library includes various library objects. Dragging and dropping those objects onto the 3D View, the Wizard tab appears. You can customize those objects by adjusting the slide bar.

**Note:** The changes made in the Attribute Window will not be applied to Object Customize Wizard.

**Note:** You cannot use "Undo" on attribute changes made by Object Customize Wizard.

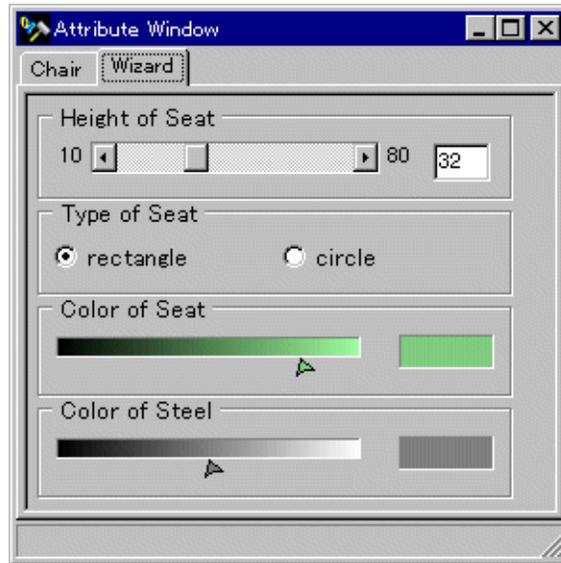


Figure 15: Object Customize Wizard

## ■ Editing nodes within PROTO

If a node within PROTO is selected, all the fields including eventIn and eventOut appear in the Attribute Window. The fields can be mapped by using IS (or can be unmapped).

**Note:** For more information on selecting node inside PROTO, please refer to "Scene Graph Window" section of this manual.

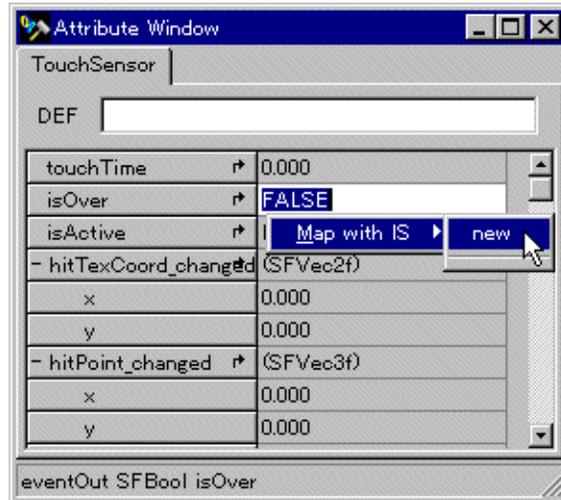


Figure 16: Attribute Window when a node within PROTO is selected

## ■ Pop-up menu

- **map with IS/new**

Newly maps a field using IS. When a field is mapped, the following dialog box appears on the screen in which you can change field names.

When mapping a new field/exposedField, the value before the field is mapped will be used as the default value.



Figure 17: The Dialog Used to Add a New Field

Note: For the details of editing within PROTO, see "Restrictions."

- **unmap from IS**

Releases a mapped field.

- **Editing the Script node**

You can add or delete fields when the Script mode is selected. If a class file exists, however, the source file (.java file) also needs to exist. If URL is not specified, you can add or delete fields.

## Keyframe Editor Window

This window is used to edit the Interpolator node defined in VRML 2.0. In the Keyframe Editor mode, the Interpolator node that automatically reflects the changes on exposed-Field is created. The route also is automatically added in this mode.

To return to the normal edit mode, exit the Keyframe Editor mode.

**Note:** In Keyframe Editor mode, editing an object means editing the Interpolator, and not the object itself. Therefore, the object's attributes will not be changed. To edit the object itself, quit the Keyframe Editor mode.

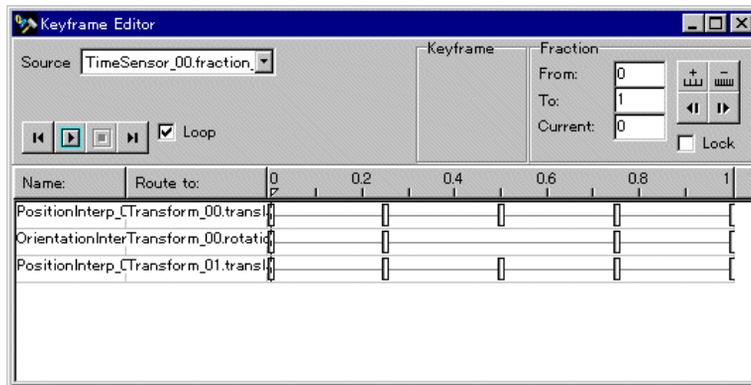


Figure 18: Keyframe Editor Window

**Note:** CoordinateInterpolator and NormalInterpolator are not supported in Keyframe Editor.

### ■ Starting or Quitting the Keyframe Editor

To start Keyframe Editor, click the corresponding icon  in the Main Window, or click Keyframe on the Edit menu.

To quit Keyframe Editor, click again the icon  in the Main Window or click Keyframe on the Edit menu to de-select Keyframe Editor, or click the x (Close) button on the top-right corner of the window.

## ■ Selecting the Source

Select, out of the Source combo box, the field (or the event) that controls the Interpolator. Normally you are supposed to select the fraction event of TimeSensor. To let fields or events appear in this combo box, you need to give DEF names to TimeSensor and so on.

## ■ Enlarging/reducing or scrolling the display

Click the enlarge/reduce  icons to enlarge or reduce the display area.

Click the scroll icon  to scroll the display area.

## ■ Specifying the fraction

- **From field and To field**

Specifies the Keyframe Editor display area. This area varies if the enlarge/reduce or scroll button is clicked.

- **Current field**

Specifies the location when Keyframe is to be added. This value varies if the Fraction panel (the calibrated portion) is clicked.

## ■ Locking the display

When you check the Lock checkbox, the enlarge/reduce buttons, the scroll button, the From and To fields become inoperable.

## ■ Seeing how animation works on the screen

Click the Play button  to see how the linkage between the Interpolator and the current Source works on the display.

**Note:** To see how the entire world works on the screen, quit the Keyframe Editor and then press the Play button in the Main Window.

## ■ Loop checkbox

Specifies whether to loop the animation or not. If TimeSensor is specified in the Source combo box, it will work together with the loop field.

## ■ Selecting multiple Keyframes

More than one keyframe that belong to the same Interpolator can be selected by clicking while holding down the Shift key.

## ■ Pop-up menu

Right clicking in the Keyframe edit area will display the popup menu shown on Figure 16.

- **New Keyframe**

Adds a new Keyframe on the location specified with the current Fraction.

- **Cut**

Cuts the selected Keyframe. The Keyframe that has been Cut is loaded onto the clipboard. Cut is valid only for the Interpolator currently selected.

- **Copy**

Copies the selected Keyframe. The Keyframe that has been Copied is loaded onto the clipboard. Copy is valid only for the Interpolator currently selected.

- **Paste**

Pastes the Keyframe that was Cut or Copied.

- **Select all**

Selects all the Keyframes contained in the Interpolator selected.

- **Manipulate selection**

You can apply the Keyframe operations (explained below) to more than one Keyframes selected.

- **Copy column to**

Copies the Keyframe contained within the current Fraction to the other Fraction. This function is useful when copying more than one Interpolators.

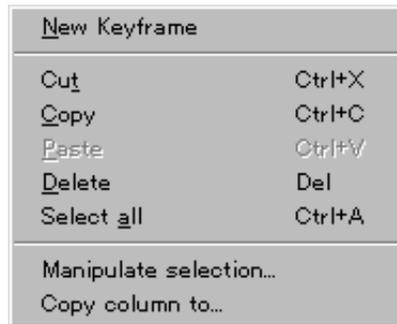


Figure 19: Pop-up menu

## ■ Keyframe operations (Scaling and Smoothing)

You can scale or smooth more than one Keyframe selected. Smoothing, however, is effective only for PositionInterpolator.

While keeping the Shift key caved-in, you can select more than one Keyframe, then right-click to bring up the pop-up menu, and select Manipulate selection.

**Note:** Smoothing can only be done on PositionInterpolator.

### • Scaling

Scale Sheet is displayed when you select the Scale tab shown on Figure 17.

Scaling Factor:

Specifies the value for scaling.

50%: double speed

200%: half speed

-100%: invert

Scaling point:

Specifies the center position for scaling.

Left-hand keyframe: left-most

Center of selection: center position

Right-hand keyframe: right-most position of the keyframe



Figure 20: Keyframe operations dialog (the Scale tab)

- **Smooth**

Creates intermediate keyframes with the Bezier curve. This enables smoother interpolation.

**# of smoothing keyframes:**

Specifies the number of keyframes to newly be inserted to an existing keyframe.

**Tension, Bias:**

Specifies the parameters for the Bezier curve. Normally you are supposed to specify within the range of -1.0 to +1.0. Specifying 1.0 for Tension provides linear interpolation.

**Loop first and last keyframe:**

Use this function when the first and last keyframes agree.

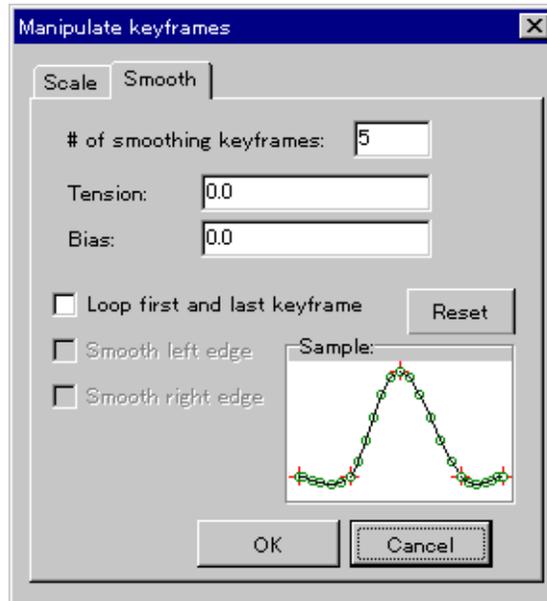


Figure 21: Keyframe operation dialog (the Smooth tab)

## Library Window

### ■ The types of library

This window provides you with various types of libraries, which can be added to your world by dragging and dropping. By clicking one of the tabs, you can switch between these libraries.

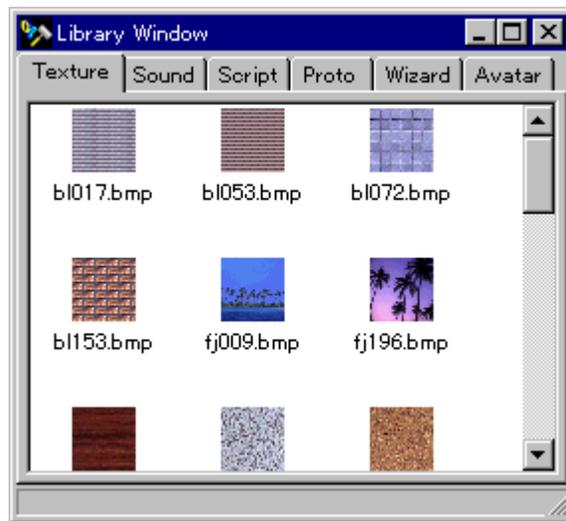


Figure 22: Library Window

Note: The Movie tab is not supported in the current version.

### ■ Dragging and Dropping

#### ● Texture

Select the texture and drag and then drop onto the object on the 2D/3D View Window to paste it as shown on Figure 23 .

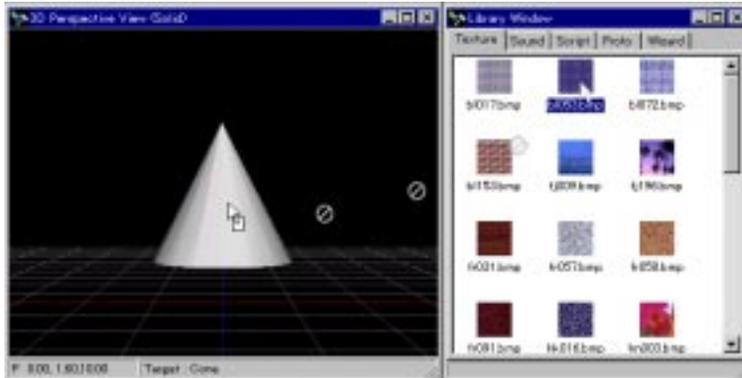


Figure 23: Pasting texture by dragging and dropping

- **Sound, Script, Proto, Wizard:**

Drag and drop onto the 2D/3D View Window.

**Note:** You are not allowed, however, to drag and drop the PROTO/Wizard library onto the PROTO/Wizard definition part.

- **Avatar:**

Drag and drop onto the Multi User Window or the 2D/3D View Window.

**Note:** Avatar page will appear by pushing the Create Multi User world button in the Multi User Window. Refer to "Multi User Window" on page for more information.

## ■ Starting the related application

Double-click the icon of the application to preview with related application.

## ■ Displaying Help in the Script library

You can display the help information in the Script library by clicking the corresponding icon and then pressing the F1 key. The help information will be displayed within the HTML browser.

**Note:** This function works only with Netscape Navigator 3.0 or later or Internet Explorer 3.0 or later.

Note: If Help menu is selected while Netscape Navigator or Microsoft Internet Explorer isn't open, it will open the browser without showing the Help file. To view Help, select the Help menu once again.

## ■ Changing the Library folder

Right-click a folder to display the menu on which you can change the folder being referred to by the library currently displayed.

To change a folder, double-click your desired folder to open it, and then press the OK button.

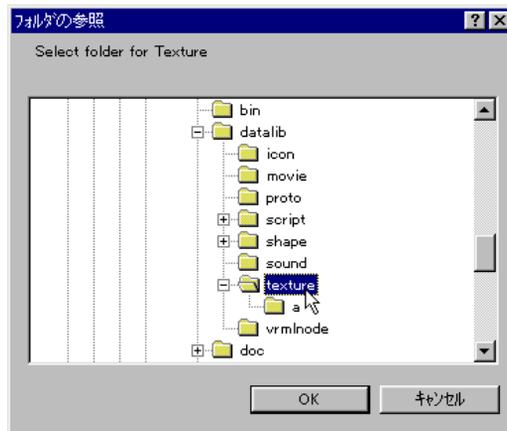


Figure 24: Dialog for selecting a folder

## ■ Customizing an icon

Icons are displayed in the Library Window based on the following rules:

1. Texture files are displayed as icons in the Texture library.
2. In the library windows, a .bmp file in the same directory as a library file is used as an icon. If an .bmp file does not exist, \_default.bmp within the library directory will be used instead. Note that those files are prefixed with "\_" (Underbar).

3. If a `_default.bmp` file does not exist either, the Conductor icon will be used instead. Users can customize icons by providing a `.bmp` file that corresponds to each library file.

**Note:** The size of an icon file must be 32 x 32 dots or larger.

## ■ Formats supported in each library

- **Texture**

bmp, gif, and jpeg

- **Sound**

wav, mod

**Note:** To use mod, DirectX3.0 or higher must have been installed.

- **Script**

VRML file that defines a script and a class file required.

- **Proto**

VRML file that defines a PROTO. PROTO definition and PROTO instance must be written in the VRML file.

**Note:** PROTO instance cannot be surrounded by another node.

- **Wizard**

VRML file with the PROTO definition and instance. Pre-installed library objects conform to Object Customize Wizard . To create objects that conform to Wizard, see Object Customize Wizard Manual .

- **Avatar**

VRML file that defines the Avatar.

## ■ Adding objects to a library

You can add or delete library objects by adding/removing files to/from a library folder. You can add objects to the Texture library by dragging and dropping an image file of a supported format to the Library Window. To add a PROTO to the PROTO library, change the scope of the Scene Graph Window to the PROTO that you want

to register, and choose "Register to the Library" from the pop-up menu. If library objects are added or deleted, the corresponding icons will automatically be updated.

## Message Window

This window shows you messages, such as the warning messages issued during run-time or those issued by the Java compiler.

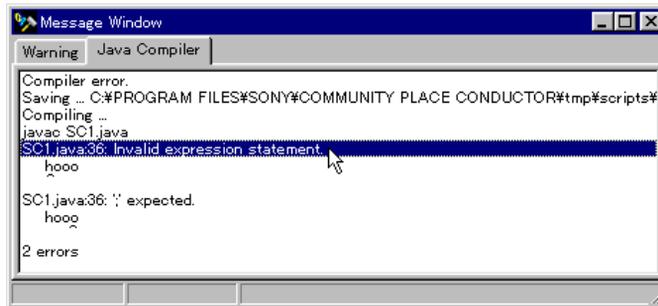


Figure 25: Message Window

### ■ Warning message

Select the Warning tab to display the warning messages.

### ■ Message from Java compiler

Select to display the error messages issued during compilation. If a compile error occurs, click the corresponding error line to jump to the location of the error.

## Multi User Window

The Multi User Window is used during the creation of a multi-user world. It can configure the multi user server (Community Place Bureau (Bureau from now on)) and the avatar that is available in that world. For usage of the Bureau, refer the manual of the Bureau.

### ■ Creating a Multi User World

A button that says "Create Multi User world" will be shown in the Multi User Window (This will not be shown when you add a Sony\_WorldInfo node in the Scene Graph window or when you read a multi user world). To make the currently edited contents into a multi user world, push this button. As a result, Sony\_WorldInfo node is generated and it becomes a multi user world.

After doing so, the following configuration of the multi server world and the avatar to be used can be done.

To change a world back to non-multi user world, delete the Sony\_WorldInfo node in the Scene Graph window.

### ■ Configuring Multi User Server

When the Server tab is clicked, the Server page shown on  will be displayed.

You can set the host name and the port number of the Multi User Server (the Bureau) that the contents will connect to.

The changes made in this page will be reflected to the cpBureau field in Sony\_WorldInfo node. If the "host" section is left blank, it will automatically be set to "localhost." This can be used to run the Bureau and the contents in your local machine for testing the multi user world that you have created.

When the port number section is left blank, it will automatically be set to "5126", which is the default port number of the Bureau.

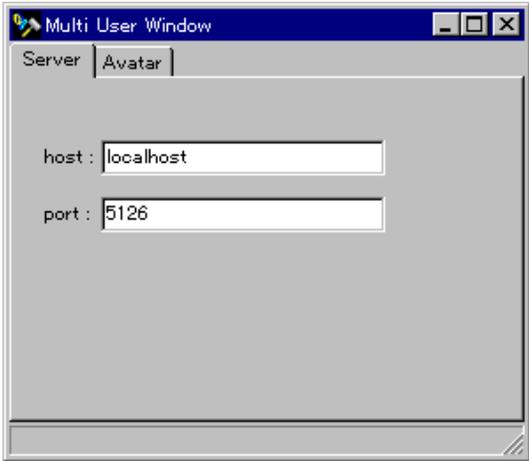


Figure 1: Selecting the Server tab

■ **Specifying the Avatar**

Click on the Avatar tab to display the Avatar page shown on . The Avatar page gives a list of avatars that users can use in the current multi user world.

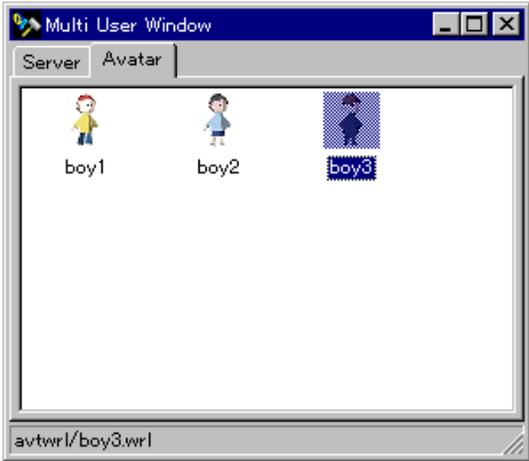


Figure 2: Selecting the Avatar tab

- **Adding to the Avatar List**

Drag and drop the Avatar icon from the Avatar page of the Library Window to the Avatar page of the Multi User Window (of the 3D View Window) to add a new avatar. The Avatar page in the Library Window can be selected only when the contents is a multi user world.



Figure 3: Adding Avatar by dragging and dropping

- **Deleting from the Avatar List**

Choose an Avatar in the Avatar page of the Multi User Window and press Delete to delete an Avatar from the list. Choosing the avatar and selecting "Delete" from the pop-up menu will also delete an Avatar from the list.

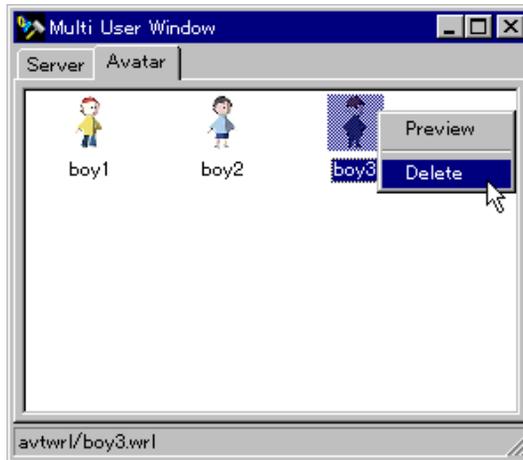


Figure 4: Deleting the Avatar

- **Editing an Avatar**

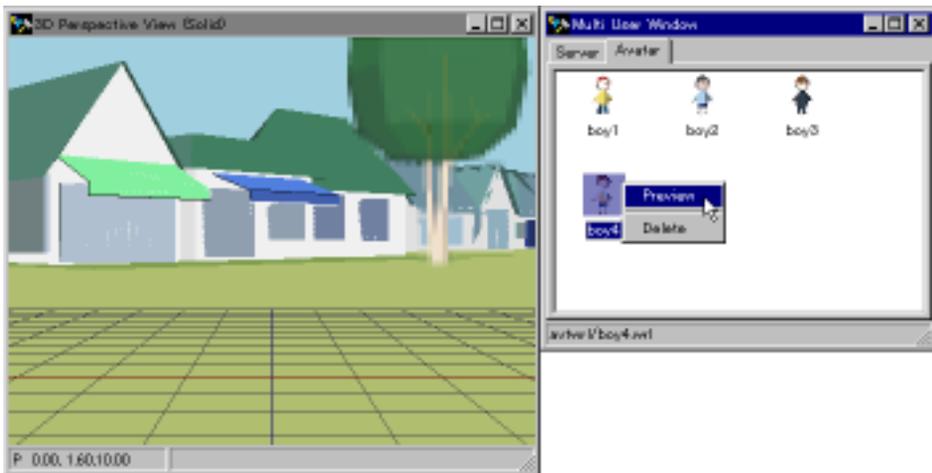
You can verify or edit an available avatar shown in the list. Select the avatar that you wish to edit and select "Preview" from the pop-up menu. (or push the Enter key or double click)

The chosen icon will be surrounded by a red rectangle and the avatar will be displayed in the 3D View Window.

The avatar file will be generated in the Scene Graph Window as an Inline node, and by entering the scope of that Inline node, you can edit the avatar.

Note that the avatar is just temporarily added to the world for editing, and it is not an object included in the world. The Inline node of the avatar will be deleted from the world when the world is saved to a file.

To quit editing the avatar, select that avatar in the Avatar page and select "Preview" from the pop-up menu. (or push Enter key or double click)



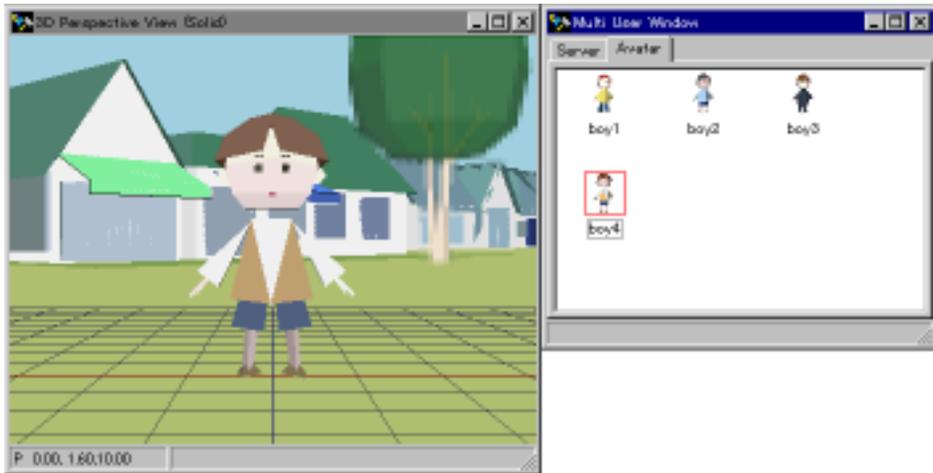


Figure 5: Before generating Avatar (top)  
and generating Avatar (bottom)

- **Checking the operation of the Multi User World**

To do an operation check of a multi user world, run the Bureau according to the host and the port number configured in the Server page.

Then, save the current contents and push the Start button to check its operation in the Conductor.

To confirm the figure and movement of the avatar, run more than one Browser (or Conductor) and read its contents.

## Parallel View Window

This window enables you to display up to 3 angles of Parallel View (the parallel projection view): Top View, Side View, and Front View. All these views will always display objects in the wire frame mode.

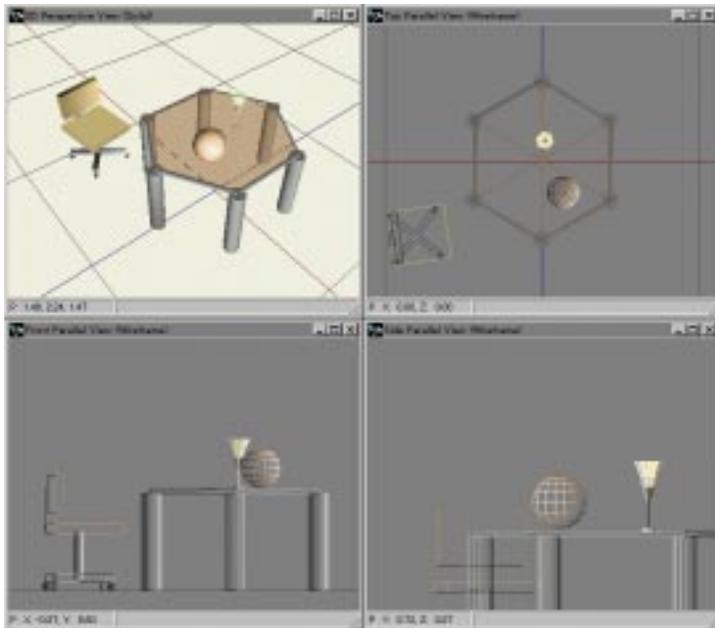


Figure 6: Parallel View window

### ■ Displaying and changing your eye position/direction

The values of your eye position/direction currently set are indicated on the sub-panel at the lower-left corner. You can change the following values by clicking the sub-panel:

table 1: Values shown on the sub-panel

Description	Variable	Example
Eye position in world coordinates	(P)	P Y: -0.47, Z: -2.13
Current zoom ratio	(S)	S 2.222
Eye direction in world coordinates	(V)	V -1.00, -0.00, -0.00

The eye position/direction can be moved only on the surface parallel to each projected plane in world coordinates as shown:

table 2: Eye position/direction of Parallel View

View mode	Eye position	Eye direction
Top View	On the surface parallel to the X-Z plane	Fixed in the direction of -Y
Side View	On the surface parallel to the Y-Z plane	Fixed in the direction of -X
Front View	On the surface parallel to the Y-X plane	Fixed in the direction of -Z

## ■ Pop-up menu

Click the right mouse button to display the pop-up menu. It provides the following sub-menus:

- **Home Position command**

Restores both the eye position and the Zoom ratio to their default values. You will not be shown the process of returning to the home position, but will immediately be brought back to it. This is different from the 3D View. The Home Position values are fixed as follows:

table 3: Values of Home Position

View Mode	Eye position	Eye direction	Zoom ratio
Top View	X:0, Z:0	(0, 0, -1)	1.0
Side View	Y:0, Z:0	(-1, 0, 0)	1.0
Front View	X:0, Y:0	(0, -1, 0)	1.0

### ■ Creating an object

You can create objects in the Parallel View Window as well as in the 3D View Window. The only difference between them is the position where an object is to be created. In Parallel View, an object will be created vertically just below the position you click the mouse. The world coordinate for the object will be created on the grid of the world coordinate at its 0 position for a corresponding direction of eyes.

For example, if you have created an object with Top View, it will be created on the grid position (X, 0, Z).

### ■ Walk through a virtual world

You can move in every direction basically in the same manner as you do with 3D View.

First select the Walk mouse mode in the Conductor window. Then, drag the mouse with its left button within the Parallel View Window. You are not allowed, however, to use the arrow keys on the keyboard.

Each Parallel View allows you only to move on the surface perpendicular to each eye direction.

Moving in your eye direction will not change the scale of the scene viewed. This is because the scenes are obtained by the parallel projection. Instead, Parallel View provides you with additional functions of Zoom In/Out.

You can walk through a world in the Parallel View mode as follows:

table 4: Walk through a world

Movement	Modifier keys	Mouse	Top	Side	Front	Zoom Ratio
Move upward		Drag downward	+Z	-Y	-Y	
Move downward		Drag upward	-Z	+Y	+Y	
Move rightward		Drag leftward	-X	-Z	-X	
Move leftward		Drag rightwards	+X	+Z	+X	
Zoom in	Shift key	Drag upward				Increase
Zoom out	Shift key	Drag downward				Decrease

By pressing down Ctrl key and clicking on the Parallel View window, you can move the viewpoint to that position.

**Note:** The default value for zoom ratio is set to 1. Zoom ratio can range from minimum 0.01 to maximum 16, based on the default value of 1.

## ■ Moving an object

You can move an object in the same way as you do in 3D View. If you move an object with Top View, for example, an object is only allowed to move on the X-Z surface. Its position will never move in the direction of Y-axis.

## ■ Rotating an object

You can rotate an object in the same way as you do in 3D View, except that the rotation is made only on the axis that goes through the local coordinate origin of the selected object and is parallel to the eye direction in the current View.

## ■ **Scaling an object**

You can scale an object in the same way as you do in 3D View. Dragging the mouse toward the center of an object will reduce it. Dragging the mouse away from the object will enlarge it. The center of scaling, however, is the local origin of the object, while in the 3D View the center of scaling is located at the center of the box enclosing the object.

## Route Window

Route window is used to display, add, or delete the routing of events to occur within the current scope. For the details of the current scope, see the explanation about the Scene Graph Window.

You can choose your desired display mode by clicking one of 3 buttons on the left bottom of the window:

-  : Text display mode
-  : Visual display mode
-  : Mixed display mode

### ■ Text display mode:

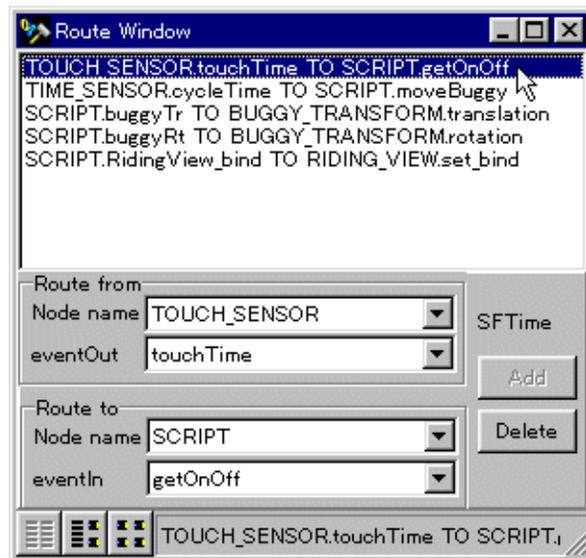


Figure 7: Route Window

Route from

NodeName: Name of the node to output an event

eventOut: Name of eventOut(the field name of the eventOut type)

Route to

NodeName: Name of the node to receive an event

eventIn: Name of eventIn(the field name of the eventIn type)

**Note:** Every node should have a predefined node name on it.

## ■ Adding a route

You can add a route in the following procedure:

1. Select the NodeName of the node to output an event from the list.
2. Select eventOut from the list.
3. Select the NodeName to receive an event from the list.
4. Select eventIn from the list.

**Note:** Only the eventIn names with the same name type as the eventOut will appear on the list.

5. Pressing the Add button, a route is added and displayed on the routing list.

## ■ Deleting a route

Select the routing you want to delete, and then press the Delete button.

## ■ Visual display mode:

In the Visual display mode, you can display routing with icons. Icons are displayed in the units of node, and will be laid out automatically if routes are added or deleted. The node with a DEF name but without a route will be displayed at the bottom of the window.

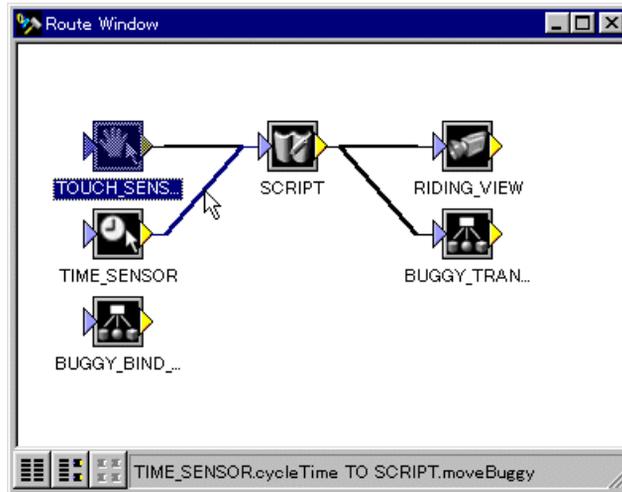


Figure 8: Visual display mode

- **Selecting a route**

Clicking the line connecting nodes, the route appears on the status bar at the lower portion of the Route Window. If more than one route exist, they will be displayed in the pop-up format.

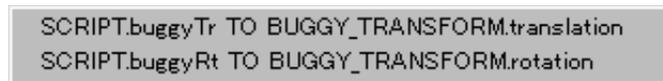


Figure 9: When more than one route exist

- **Adding a route**

Drag and drop icons within the Route Window. You can also add a route by dragging and dropping between the Route and Scene Graph Windows. (Nodes within Scene Graph are not required to be have DEF names.)

For the details on how to add a route, see "Tutorial."

- **Deleting a route**

First select a route, and then right-click to display the pop-up menu. Select the route you want to delete on the menu to delete it.



```
Delete SCRIPT.buggyTr TO BUGGY_TRANSFORM.translation
Delete SCRIPT.buggyRt TO BUGGY_TRANSFORM.rotation
```

Figure 10: The pop-up menu used to delete a route

- **Automatically creating and modifying a DEF name**

Drag and drop a node without a DEF name from the Scene Graph Window to the Route Window. The node will automatically be given a DEF name; the DEF name given here can be modified. The DEF names are created according to the types of node (such as TouchSensor).

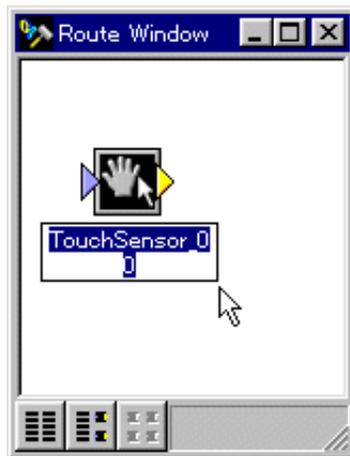


Figure 11: Immediately after a DEF name is automatically created

## ■ Pop-up menu

When an icon is selected:

You can see the routing given to that icon.

When an icon is not selected:

- **2D Layout**

Lay outs the routing in 2 dimensions.

- **1D Topological layout**

Lay outs the routing in a single dimension (in geometrical order).

- **1D Alphabetical layout**

Lay outs the routing in a single dimension (in Alphabetical order).

- **Animate changes**

Displays the rearrangement of the layout with animation.

- **Fade descendants**

Displays the nodes contained in the node selected within the Scene Graph Window with fade-out. This lets you know what routes will be deleted if you delete the node you select.

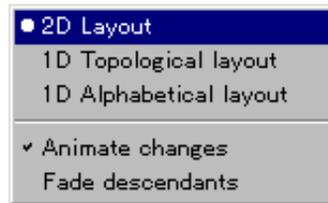


Figure 12: Pop-up menu with no icon selected

## Scene Graph Window

This window displays the tree structure of a world within the current scope. All the VRML 2.0 nodes will be shown in the tree structure. You can select any of the nodes shown and move it with drag & drop.

The nodes which are given DEF names will be displayed with the [ ] mark. The Inline node shows the URL within <>.

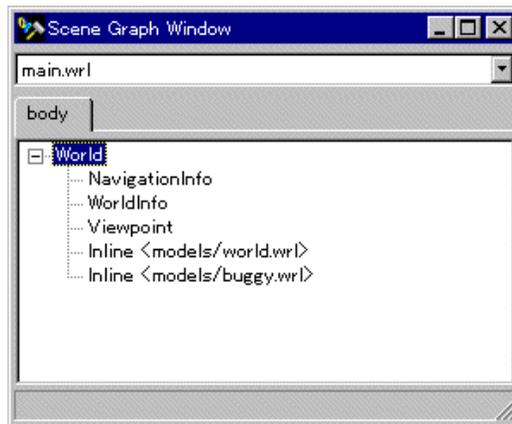


Figure 13: Scene Graph Window

### ■ Switching the current scope

A VRML file containing the Inline node will be given a single scope. You can work only on the current scope of each node. You can switch the current scope to another by using the combo box at the top of the screen.

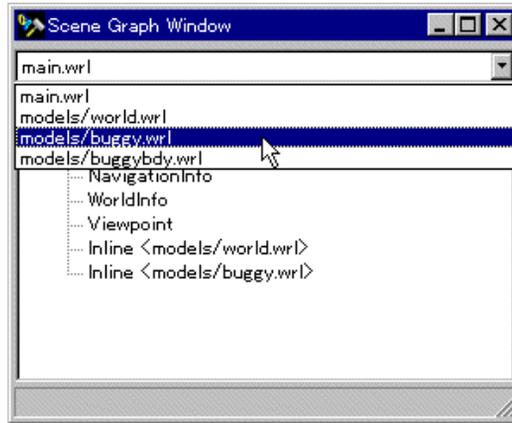


Figure 14: The Combo Box for Switching the Current Scope

## ■ Switching the PROTO scope

If the file currently selected contains PROTO, the tab for switching the PROTO scope will appear on the screen.

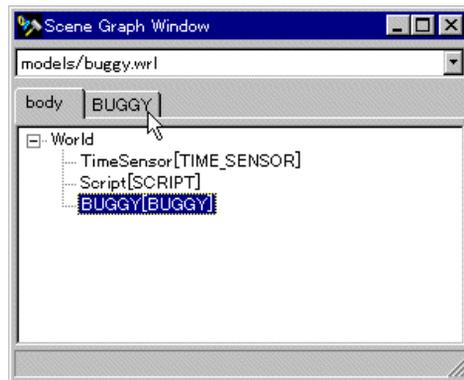


Figure 15: Tab for switching PROTO scope

## ■ Selecting a node

Click the node you want to select. The attributes of the selected node can be modified from within the Attribute Window.

Double-clicking the node will both select the node and display the Attribute Window as the active window.

## ■ Displaying the Help information

By pressing the F1 key, the VRML 2.0 help information will appear on the WWW browser window. If an object is selected, its node will also be explained within the window.

**Note:** This function works only with Netscape Navigator 3.0 or later or Internet Explorer 3.0 or later. To use this function, start the browser and press the F1 key.

## ■ Pop-up menu

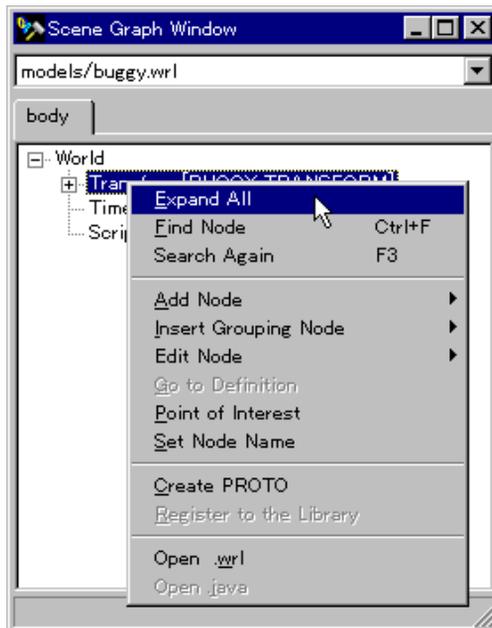


Figure 16: The Pop-up Menu of the Scene Graph Window

- **Expand All**

Expands all the trees under the currently selected node.

- **Find Node**

The following dialog box will appear.

To search from the DEF name of the Node, click on the "By Name" tab. To

search from the type of the Node, click on the "By Type" tab.

Strings to search can be selected from the combo box.

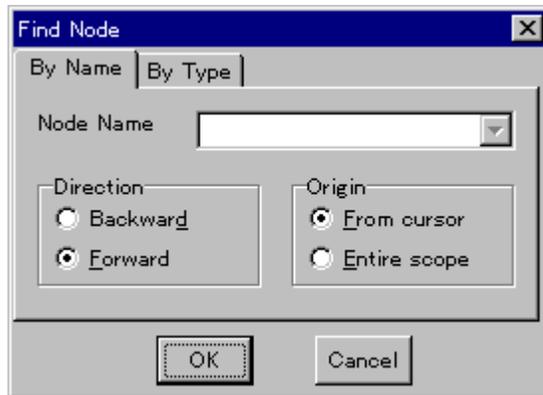


Figure 17: Find Node Dialog Box

Direction

Backward: Searches backward

Forward: Searches forward

Origin

From cursor: Search starts at the cursor point

Entire scope: Searches from the beginning of the file

- **Search Again**

Search again using the same condition selected during "Find Node."

- **Add Node**

Adds the nodes which are prescribed by VRML2.0. The nodes are classified into sub-menus.

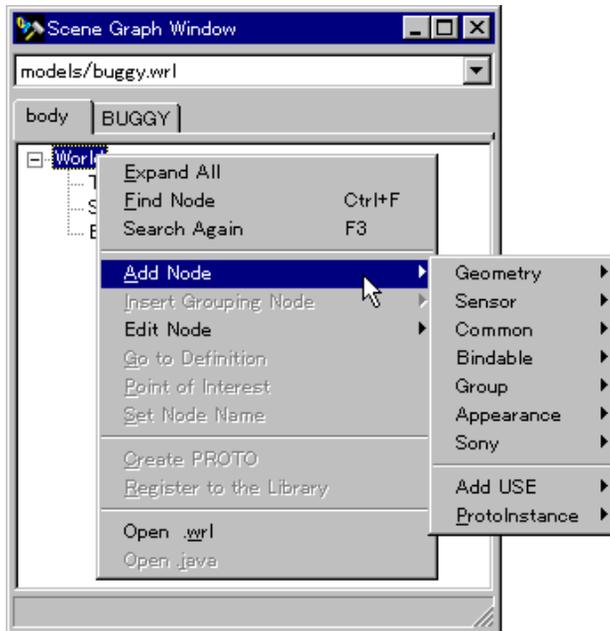


Figure 18: Selecting the Add Node

**Add Use**

DEF names that are valid in the current scope will be shown. When selected, node is created using "USE."

**Add ProtoInstance**

PROTO's that are valid in the current scope will be shown in the sub-menu. When selected, it adds a new instance.

- **Insert Grouping Node**

Adds the grouping node as a parent node to the currently selected node.

- **Edit Node**

Adds the grouping node as a parent node to the currently selected node.

- **Goto definition**

Jumps to the definition part of the PROTO instance or the Inline node selected.

- **Point of Interest**

Moves from the current eye position to the front of the selected object. This can be used only for grouping node, geometry node, light node, and Viewpoint.

- **Set Node Name**

Displays the dialog used to specify the DEF name for the selected node.



Figure 19: DEF name Input Dialog

- **Create PROTO**

Creates PROTO based on the selected node. The selected node will be replaced by the PROTO instance.

Note:1) By selecting this menu, you will be unable to Undo your work.

Note:2) Creating a PROTO using CreatePROTO will automatically generate DEF name in capital letters.

- **Open .wrl**

Displays the VRML file you have loaded.

- **Open .java**

Loads a Java file to the Text Editor. You can use this menu only when a Script node has been selected.

- **Register to the Library**

Registers the file referenced by the selected node to the library.

List of files that can be registered are shown in the sub-menu. When selected, it is registered in the library corresponding to the format of the file.

When a scope of a PROTO is selected, PROTO name will be shown in the sub-menu. When selected, PROTO that is being edited will be registered to the Proto library.

When a file with the same name is already registered, dialog box will appear to confirm overwriting. If you do not want to overwrite, change the registration folder in the next dialog box. Refer to the section on Library Window for file formats.

- **Open .wrl**

Shows the VRML file of the file scope selected in the Text Editor.

*Note: You must save the file before displaying the contents of a VRML being edited.*

- **Open .java**

Shows the Java file in the Text Editor. This menu can only be used when Script node is selected and the source file exists.

## Script Expert Window

Used to create the template (the model) of a Java program conforming to the VRML 2.0 script node.

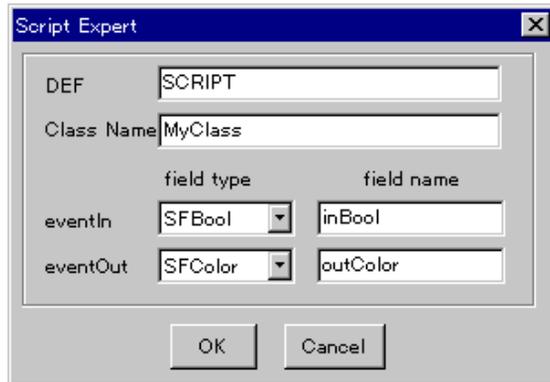


Figure 20: Script Expert dialog

### ■ Using Script Expert

Fill the following four fields, and then press the OK button:

DEF:

Names the script node so that the Java program can be called up from VRML.

Class Name:

Specifies the class name for Java.

This class name will be used as a file name.

**Note:** For a class name, do not include any key word reserved by Java.

Data Type:

Specifies the VRML data type for eventIn or eventOut. Choose a data type out of the VRML Data Types listed.

field name:

Specifies the VRML data type for eventIn or eventOut. Choose a data type out of the VRML Data Types listed.

Press the OK button in the end. This creates the template (the model) of a Java program and compiles it. The Script node is added to the Scene Graph Window at the same time.

**Note:** The Script Expert Windows does not allow you to specify more than one eventIn or eventOut. You can add an event only in the Script Attribute Window. To do this, first choose the Script node in the Scene Graph Window, and then select add field in pop-up menu in the Attribute Window.

## Text Editor Window

The text editor is used to edit VRML files or Java files. When editing scripts, this text editor automatically adds or deletes a necessary handler by working together with the Attribute Window.

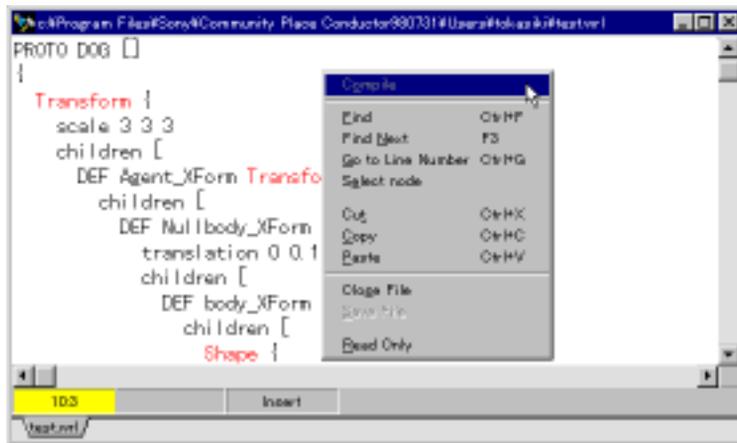


Figure 21: Text Editor window

### ■ Starting the Text Editor

The Text Editor will be started by taking one of the following actions:

Select Open .wrl or Open .java in the pop-up menu has been selected in the Scene Graph Window.

Drag-and-drop a text file has been dragged and dropped out of the Explorer.

### ■ Quitting Text Editor

You can close Text Editor by clicking the x button on the upper right corner of the window. If the file has been modified, a save dialogue will appear.

**Note:** When VRML file is modified after selecting "Open .wrl" from pop-up menu of the Scene Graph window, changes will not be reflected until the file is re-loaded.

## ■ Pop-up menu

### • Compile

Compiles the file being edited. If the compile completes successfully, "Finished" appears in the Message Window. If an error occurs, a corresponding error message will appear.

### • Find

A dialog box will appear to search for the text included in the file being edited.

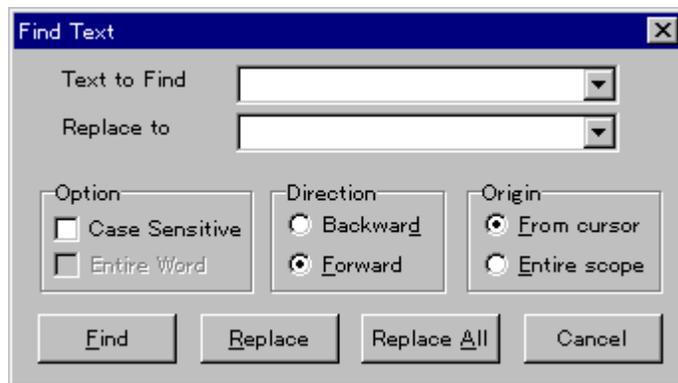


Figure 22: Find Text dialog box

Text to Find: Enter the text to search for

Replace to: Enter text to replace the above text

#### Option

Case Sensitive: When checked, the search will be case sensitive.

Entire Word: When checked, searches for words that include the text.

#### Direction

Backward: Backward search

Forward: Forward search

#### Origin

From cursor: Starts searching from the cursor point.

Entire scope: Searches the entire file.

- **Find Next**  
Searches again using the same condition given in the Find dialog.
- **Go to Line Number**  
Jumps to the specified line.
- **Cut/Copy/Paste/Delete**  
You can Cut/Copy/Paste/Delete texts.
- **Close File**  
Closes a file.
- **Save File**  
Saves a file.
- **Read Only**  
Changes the Read Only attribute of a file.

## Texture Mapping Window

On Texture Mapping window, you can specify the color and texture mapping to the IndexedFaceSet node.

**Note:** This window can only be used when IndexedFaceSet is selected

Each of the sheets are described below.

### ■ Color sheet

By clicking the Color tag, the Color page shown below will be displayed. This page will only be displayed when non-default Color Mode is selected.

The components of the Color sheet are as follows:



Figure 23: Color sheet

(1) Face

When the Color sheet is active, the face screen displays the current face as it would appear if the IndexedFaceSet had no texture.

(2) Color toolbar

The tools are used by clicking or dragging onto the Face area.

table 5: Color toolbar

Button	Function
 Brush tool	Applies the current color to faces or vertices
 Dropper tool	Sets the current color to be the same as the nearest vertex to the mouse position.

(4) Fill all faces

This button fills all faces with the current color.

(5) Current color

The current color is displayed here. Clicking on the color swatch pops up a color choosing dialog. This feature could be replaced by a built-in color chooser on the Color sheet.

(6) Recently used colors

Old colors are moved into this area, so that the user may switch back by clicking on one of the swatches.

(7) Status bar

This contains a different message depending on the color mode and tool. For the brush tool, the following messages shown in Table 16 are used.

table 6: Color mode messages

Color mode:	Message:
1 color per face (indexed)	Click on the face to use this color.
1 color per face	Click on the face to use this color.
1 color per vertex	Click near a vertex to change its color. This will also affect adjacent faces.
1 color per vertex (indexed)	Click near a vertex to change its color.

### ■ Texture Coordinate sheet

By clicking the Texture Coordinate tag, the Texture Coordinate sheet shown below will be displayed. This will only be displayed when a non-default Texture Mode is selected.

The components of the color sheet are as follows:

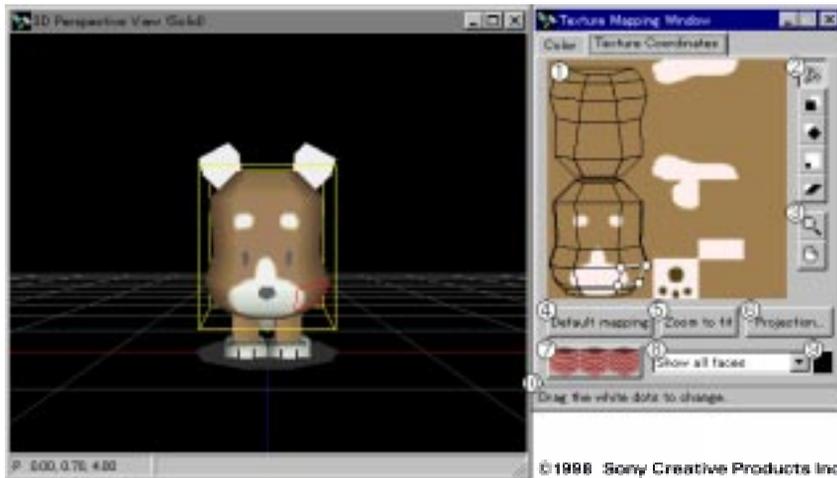


Figure 24: Texture coordinate sheet

#### (1) Texture sample

Shows the current texture and texture coordinates. When the Texture Coordinate sheet is active, the current face is shown in black with its texture applied. If there is

no texture, or the texture url cannot be loaded, a default texture is used so that coordinates may be set properly.

(2) Texture coordinate tools

Contains the tools described below. All of these tools are operated by dragging the mouse within the texture sample.

table 7: Texture coordinate tools

Button	Function
 Point-by-point	texture coordinate points may be individually positioned
 Transform	all texture coordinates may be moved as a group
 Rotation	all texture coordinates may be rotated around a chosen point
 Scale	all texture coordinates may be scaled in the x and y directions by manipulating the corners of the bounding rectangle of the texture coordinates
 Skew	all texture coordinates may be skewed in the x or y direction

(3) Texture sample tools

These have no effect on the texture coordinates. These are used to adjust texture coordinate display on the Texture sample.

table 8: Texture sample tools

Button	Function
 Zoom	The left mouse button zooms in and the right mouse button zooms out, centering the new view at the position that the mouse is clicked.
 Pan	The user may pan the texture sample by dragging the mouse.

**(4) Default mapping**

Sets the texture coordinates such that the bounding box of the texture coordinate fits inside the rectangle [0, 0, 1, 1].

**(5) Zoom to fit**

Adjusts the zoom and pan settings such that the texture coordinates all appear within the texture sample.

**(6) Projection**

Pops up the Projection dialog.

**(7) Status bar**

The status bar is used to present helpful messages regarding the current tool.

**(8) Mesh Type**

Specify the display region of the Mesh, as shown on table9.

table 9: Mesh Type

<b>Mode</b>	<b>Description</b>
Show selected face	Shows only the selected faces
Show neighbor faces	Shows neighbor faces
Show all faces	Shows all faces

**(9) Mesh Color**

When the Mesh type is either "Show neighbor faces" or "Show all faces", specify the color of the Mesh which is not selected.

**(10) Status bar**

The status bar is used to present helpful messages regarding the current tool.

## ■ Color Mode dialog

The Color Mode dialog (Figure 40) allows the user to set the current IndexedFaceSet's "color", "colorIndex", and "colorPerVertex" fields. The default mode is "Use Appearance". The user may choose a color mode from the list of radio buttons.

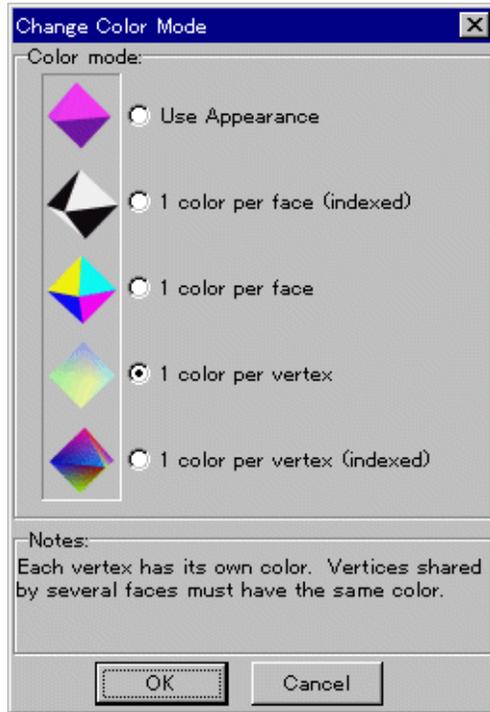


Figure 25: Change color mode dialog

A summary of the VRML structure for each color mode is given in the table below.

table 10: Color modes

Mode	colorPerVertex	color field	colorIndex field
Use Appearance	--	not used	not used
1 color per face (indexed)	FALSE	used	used
1 color per face	FALSE	used	not used
1 color per vertex	FALSE	used	not used
1 color per vertex (indexed)	FALSE	used	used

In the Notes section of the dialog box, a brief description of the currently selected color mode is given. Following is a list of the description texts:

table 11: Description of the color modes

Color mode	Description
Use Appearance	This IndexedFaceSet uses the color information given in its Appearance node.
1 color per face (indexed)	Each face has a solid color, which is chosen from a palette of colors. This will optimize the file size if the number of different face colors is small (but greater than one).
1 color per face	Each face has a solid color. This will optimize the file size if the number of different face colors is large.
1 color per vertex	Each vertex has its own color. Vertices shared by several faces must have the same color.
1 color per vertex (indexed)	Each vertex has its own color. Vertices shared by several faces may have different colors.

**Note:** When "1 color per vertex" or "1 color per vertex (indexed)" is selected in Change Color Mode, appropriate color might not appear in the 3D View Window.

In some cases, changing the color mode will change how the object appears (for instance, when switching to "Use Appearance", all color information is lost). A list of the transitions that may cause the current coloring scheme to change is given in the table below.

table 12: Color mode transitions that can change the object's coloring

From:	To:
1 color per face (indexed)	Use Appearance, 1 color per vertex
1 color per face	Use Appearance, 1 color per vertex
1 color per vertex	Use Appearance, 1 color per face (indexed), 1 color per face
1 color per vertex (indexed)	any

Note: No changes are made to the scene unless OK is clicked.

## ■ Change Texture Mode dialog

Change Texture Mode dialog (Figure 41) allows the user to set the current Indexed-FaceSet's "texCoord" and "texCoordIndex" fields. The default mode is "Default mapping". The user may choose a texture mode from the list of radio buttons.

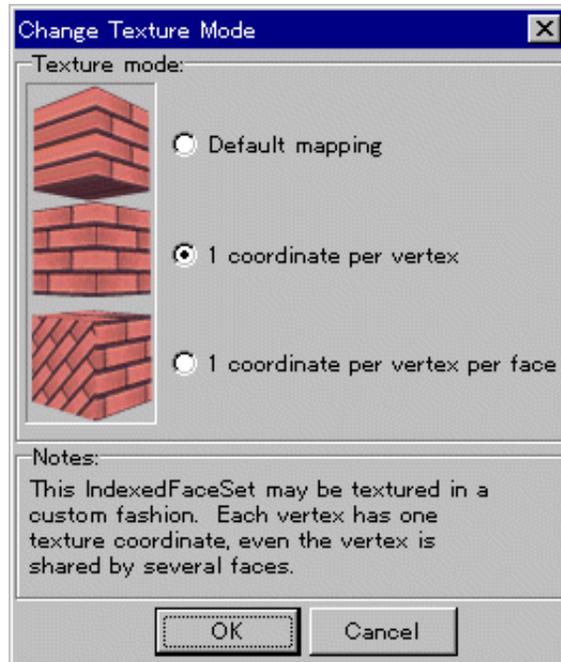


Figure 26: Change texture mode dialog

A summary of the VRML structure for each texture mode is given in the table below.

table 13: Texture modes

Mode	coord field	coordIndex field
Default mapping	not used	not used

table 13: Texture modes

<b>Mode</b>	<b>coord field</b>	<b>coordIndex field</b>
1 coordinate per vertex	used	not used
1 coordinate per vertex per face	used	used

In the "Notes" section of the dialog box, a brief description of the currently selected color mode is given. Following is a list of the description texts:

table 14: Description of the texture mode

<b>Texture Mode</b>	<b>Description</b>
Default mapping	This IndexedFaceSet will be textured automatically using the method given in the VRML specification.
1 coordinate per vertex	This IndexedFaceSet may be textured in a custom fashion. Each vertex has one texture coordinate, even the vertex is shared by several faces.
1 coordinate per vertex per face	This IndexedFaceSet may be textured in a custom fashion. Each vertex has one texture coordinate for each face that uses it.

As with color mode transitions, some texture mode transitions may change the object's appearance. A list of the transitions that may cause the appearance of the object to change is given in Table 24.

table 15: Texture mode transitions that can change the object's texturing

<b>From:</b>	<b>To:</b>
1 coordinate per vertex	Default mapping
1 coordinate per vertex per face	any

**Note:** No changes are made to the scene unless the OK button is clicked.

## ■ Texture Mapping Projection dialog

The projection dialog (Figure 42) allows the user to set all texture coordinates simultaneously using a "projection". The principle behind this technique is that the texture is applied to an imaginary shape (sphere, cylinder, or plane) using the standard VRML texture mapping system for that shape. Then the points of the IndexedFaceSet are mapped to points on the imaginary shape, and the corresponding texture coordinate is used.

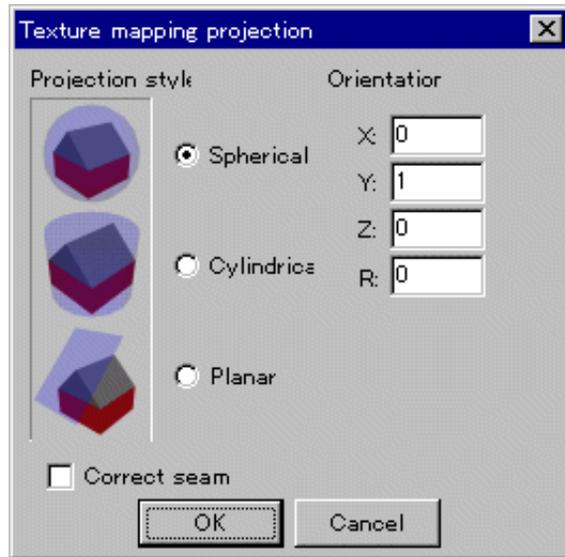


Figure 27: Texture Mapping Projection dialog

The user may choose one of the following projection styles:

table 16: Projection Style Description

Projection Style	Description
Spherical	Let $s$ be an imaginary sphere that surrounds the IndexedFaceSet. For each vertex $v$ , let $l$ be a line passing through $v$ and the center of $s$ . The texture coordinate at $v$ is equal to the texture coordinate at the intersection of $s$ and $l$ .

table 16: Projection Style Description

Projection Style	Description
Cylindrical	Let $c$ be an imaginary cylinder that surrounds the IndexedFaceSet. For each vertex $v$ , let $l$ be a line that passes through $v$ and the central axis of $c$ , and is perpendicular to the central axis. The texture coordinate at $v$ is equal to the texture coordinate at the intersection of $c$ and $l$ .
Planar	Let $p$ be an imaginary plane. For each vertex $v$ , the texture coordinate at $v$ is equal to the texture coordinate on $p$ at the point closest to $v$ .

The user may control the orientation of the imaginary shape that is used for the projection by specifying a VRML-style  $(x, y, z, r)$  rotation to be applied to the projection shape. If the OK button is clicked, the projection is applied to the current shape.

The "correct seam" option is useful for spherical or cylindrical projections. In these cases, it is possible for a face to cross the seam, so that the shape's left side has an  $s$  coordinate that is larger than the right side's  $s$  coordinate. This problem is solved by adding 1.0 to the right side of the shape. However, if "1 coordinate per vertex" mode is being used, this may not work because a single vertex would need to have two different values.

**Note:** In "1 coordinate per vertex" mode it will not function correctly since 1 vertex must contain 2 different values.

## Object Customize Wizard Operating Instructions

Conductor 2.0 offers the Object Customize Wizard functions which enable you to edit object attributes more easily than you do with the usual Attribute Window. If the object you select is supported in Wizard, the Wizard tab appears in the Attribute Window. The Wizard tab helps you with its user-friendly GUI (Graphical User Interface) to edit object attributes to meet your needs.

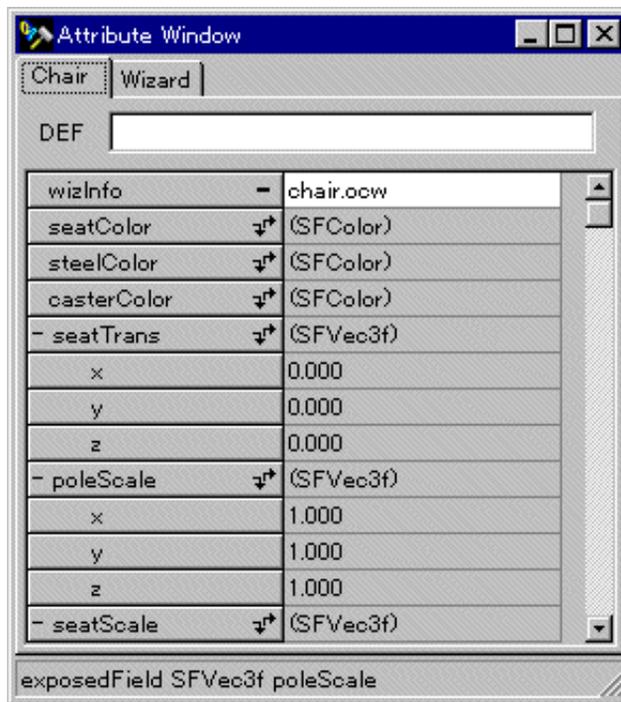


Figure 28: Attribute window

### ■ Modifying VRML objects

To get VRML objects to be supported in Wizard, modify the VRML objects and provide the Wizard description file, as shown in the following procedure. The order of operation will be shown in parentheses.

1. Create a Script node. (Scene Graph Window/Add Node/Common/Script)
2. Add the wizInfo field of the MFString type. (Attribute Window/add field)
3. Type, in that field, the url of the Wizard configuration file that is to be created later. (Attribute Window)
4. Define the Transform node created in step 1) as a PROTO. (Scene Graph Window/Create PROTO)
5. Go to the PROTO definition area (the PROTO scope). Expose the wizInfo field and the fields which will be modified by using Wizard to the interface section. (Attribute Window/map with IS)
6. Return to the body tab which includes PROTO instances. Delete and save all the nodes.

## ■ Entering necessary information in the Wizard configuration file

You can enter necessary information in the file by using the text editor. (See examples of setting files.)

The UI's are displayed sequentially from the top of the Wizard Window. In the Wizard configuration file, you are supposed to type UI settings in the order they appear in the Wizard Window. If you want the slider bar to appear on top of the Wizard Window, type "slider," and the attributes such as its maximum value on the first line of the Wizard configuration file (see Setting the slider bar for details). If you want the radio button to be displayed next, type "radio" and its attributes on the second line (see Setting the radio button for details).

If Wizard includes more than one UIs of the same kind (two slider bars, for example), type [slider.1] or [slider\_height] for example, to distinguish them by adding a string after "slider." If the strings within [ ] begin with "slider," the information within [ ] is identified as the name of a slider.

The UI's and identifiers currently available are as follows:

table 17: UI's and identifiers

UI	Identifier
Slider Bar	[slider]
Radio button	[radio]
Combo box	[combo]
Color picker	[color]

- **Setting the slider**

Example -----

```
[slider.1]
name=height
caption="Height of Seat"
value=32
min=10
max=80
```



table 18:

<b>name</b>	The name of UI within this configuration file. You cannot specify the same name for other UIs within the file. No blank spaces are allowed.
<b>caption</b>	The name of this UI to be displayed in the Wizard window. Specify a name that is easy to understand. If it includes a blank space, specify it in double quotations.
<b>value</b>	Default value for the slider bar. Only integers are currently allowed.
<b>min</b>	The minimum value for the slider bar. Only integers are currently allowed.
<b>max</b>	The maximum value for the slider bar. Only integers are currently allowed.

- **Setting the radio button**

Example -----

```
[radio_2]
name=type
caption="Type of Seat"
value=0
option=rectangle,circle
```



-----

table 19:

<b>name</b>	The name of UI within this configuration file. You cannot specify the same name for other UI's within the file. No blank spaces are allowed.
<b>caption</b>	The name of this UI to be displayed in the Wizard window. Specify the name that is easy to understand. If it includes a blank space, specify it in double quotations.
<b>value</b>	Default value for the radio button. Type "0" to specify the first option, or "1" to specify the second option.
<b>option</b>	Description about each option that can be selected for the ratio button. Insert a comma between strings. Specify a name that is easy to understand. If it includes a blank space, specify it in double quotations.

- **Setting the Combo box**

Example -----

```
[combol]
name=shape
caption="Shape of Seat"
value=0
option="rectangle", "circle"
```



-----

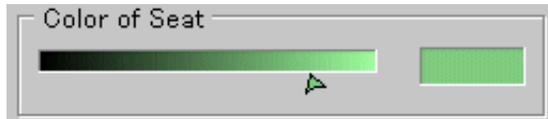
You can set the combo box in the same procedure as you do with the radio button.

For details, see Setting the radio button.

• **Setting the color picker**

Example -----

```
[color]
name=color
caption="Color of Seat"
value=0.5 0.8 0.5
```



-----

table 20: Description of variables in Combo box

<b>name</b>	The name of UI within this configuration file. You cannot specify the same name for other UI's within the file. No blank spaces are allowed.
<b>caption</b>	The name of this UI to be displayed in the Wizard window. Specify a name that is easy to understand. If it includes a blank space, specify it in double quotations.
<b>value</b>	Default value for the color picker. Specified in the same RGB format as VRML.

• **Linking each Wizard UI with a VRML field**

To have the value specified for the Wizard UI reflected in VRML, you have to specify the link between each UI and a VRML field.

Type [route] on the first line, and type the link information on the following lines. Specify the link between each UI and a field in the equation. Type the name of the UI field on the left, and that (specified in "name") of UI on the right. Be sure to specify the link information at the end of the file after all the other UI settings shown above.

Example -----

```
[route]
seatTrans.y=(height - 32) / 100
poleScale.y=height * 2 / 64
seatType=shape
seatColor=color
```

-----  
On the left of an equation, you can specify a field of the SFFloat, SFInt32, SFColor, SFVec3f, or SFRotation type which is specified in the PROTO interface. FVec3f and SFRotation must be resolved into components.

Resolve SFVec3f into x, y, and z. Specifying "seatTrans.y" as shown above, for example, you can set the value on the right of an equation to the y component of the seatTrans field (the SFVec3f type). Resolve SFRotation into x, y, z, and angle, and specify "seatRot.angle" on the right of an equation, for example. You can omit ".angle," however, and specify "seatRot" because "angle" is its default value.

Specify on the right the name of the UI used to set value in the VRML field on the left. If the field on the left is anything other than the SFColor type (namely the SFFloat, SFInt32, SFVec3f, or SFRotation type), you can specify a numerical expression on the right. The numerical expression is allowed to include the name of UI, constants, an operator of the four fundamental rules of arithmetic, and parentheses.

You can also link a single UI to several VRML fields. In the above example, the UI value of height is used to set value to both seatTrans.y and poleScale.y.

- **Others**

When you start Conductor, the VRML files appear under the PROTO tab of the Library Window. By default, the files which appear are only those files located immediately under datalib/proto/ contained in the directory where the Conductor is installed. The VRML files below those contained in datalib/proto/ will not appear in the Library Window.

A bitmap file having the same name as a VRML file and a .bmp extension will be used as an icon in the Library Window.

# Script Reference

## ■ Introduction

This document provides a detailed definition of the scripts that were prepared for Community Place Conductor. The table below lists the topics in this document.

table 21: Table of contents--Script Reference

Basic Scripts		Movement Scripts	Other Scripts
<a href="#"><u>CPSCNot</u></a>	<a href="#"><u>CPSCBoolVec</u></a>	<a href="#"><u>CPSCMoveOn</u></a>	<a href="#"><u>CPSCShowBool</u></a>
<a href="#"><u>CPSCAnd</u></a>	<a href="#"><u>CPSCBoolRot</u></a>	<a href="#"><u>CPSCMoveIt</u></a>	<a href="#"><u>CPSCShowFloat</u></a>
<a href="#"><u>CPSCOr</u></a>	<a href="#"><u>CPSCEdge</u></a>	<a href="#"><u>CPSCMoveAlong</u></a>	<a href="#"><u>CPSCShowInt</u></a>
<a href="#"><u>CPSCXor</u></a>	<a href="#"><u>CPSCStep</u></a>	<a href="#"><u>CPSCMoveTo</u></a>	<a href="#"><u>CPSCShowVec</u></a>
<a href="#"><u>CPSCSineWave</u></a>	<a href="#"><u>CPSCUpper</u></a>	<a href="#"><u>CPSCRotateAlong</u></a>	<a href="#"><u>CPSCReturnBool</u></a>
<a href="#"><u>CPSCToggle</u></a>	<a href="#"><u>CPSCLower</u></a>	<a href="#"><u>CPSCRotateTo</u></a>	
<a href="#"><u>CPSCToggleFrac</u></a>	<a href="#"><u>CPSCMatch</u></a>	<a href="#"><u>CPSCFace</u></a>	
<a href="#"><u>CPSCBoolToTime</u></a>	<a href="#"><u>CPSCRange</u></a>		
<a href="#"><u>CPSCFloatToInt</u></a>	<a href="#"><u>CPSCDelayGen</u></a>		

## CPSCNot

```
DEF CPSCNot Script {
  url "scripts/CPSCNot.class"
  eventIn SFBool inBool
  eventOut SFBool outBool
  field SFString info "this is CPSCNot script."
}
```

This script returns the logical "not" value of the boolean inBool.  
For example, it returns "false" if "true" is given, and returns "true" if "false" is given.

## CPSCAnd

```
DEF CPSCAnd Script {
  url "scripts/CPSCAnd.class"
  eventIn SFBool inOne
  eventIn SFBool inTwo
  eventOut SFBool outBool
  field SFString info "this is CPSCAnd script"
}
```

This script returns the logical "and" value of the 2 booleans (inOne, inTwo)given.

"True" is returned when both of the inputs are "true", and "false" is returned otherwise.

## CPSCOr

```
DEF CPSCOr Script {
  url "scripts/CPSCOr.class"
  eventIn SFBool inOne
  eventIn SFBool inTwo
  eventOut SFBool outBool
  field SFString info "this is CPSCOr script"
}
```

This script returns the logical "or" value of the 2 booleans (inOne, inTwo) given.

"False" is returned when both of the inputs are "false", and "true" is returned otherwise.

## CPSCXor

```
DEF CPSCXor Script {
  url "scripts/CPSCXor.class"
  eventIn SFBool inOne
  eventIn SFBool inTwo
  eventOut SFBool outBool
  field SFString info "this is CPSCXor script"
}
```

This script returns the logical "exclusive or" of the 2 booleans (inOne, inTwo) given.

If the two booleans given to the script are equal to each other, the script returns "false", and if they are different, the script returns "true".

## CPSCSineWave

```
DEF CPSCSineWave Script {
  url "scripts/CPSCSineWave.class"
  eventIn SFFloat inFloat
  eventIn SFFloat phase
  eventIn SFFloat amplitude
  eventIn SFFloat waveLength
  eventOut SFFloat outFloat
  field SFString info "this is CPSCSineWave script"
}
```

This script returns the sine value of the float, with phase(0.0) for phase, amplitude(1.0) for amplitude, and waveLength(1.0) for length of the wave. The values in parentheses are the default values of the variables.

## CPSCToggle

```
Def CPSCToggle Script {
  url "scripts/CPSCToggle.class"
  eventIn SFBool inBool
  eventOut SFBool outBool
  field SFBool defBool TRUE
  field SFString info "this is CPSCToggle script."
}
```

This is the toggle switch. Whenever a "true" event is given to the script, the script returns "true" or "false" alternatively. The first value that will be given to outBool is defined in the defBool field, and its default value is TRUE.

## CPSCToggleFrac

```
Def CPSCToggleFrac Script {
  url "scripts/CPSCToggleFrac.class"
  eventIn  SFFloat  inFloat
  eventOut SFFloat  outFloat
  field SFString info "this is CPSCToggleFrac script."

}
```

This script is useful for toggling interpolators. At the beginning, forward is set to "true". Whenever inFloat decreases, forward is toggled between "true" and "false". If forward is "true", outFloat = inFloat, and otherwise it will be 1 - inFloat.

## CPSCBoolToTime

```
DEF CPSCBoolToTime Script {
  url "scripts/CPSCBoolToTime.class"
  eventIn  SFBool  inBool
  eventOut SFTime  outTime
  field SFString info "this is CPSCBoolToTime script."
}
```

When the script receives a "true" event, it returns the time stamp of that event. At the same time, it prints out the time (as a double) in the Java Console.

## CPSCFloatToInt

```
DEF CPSCFloatToInt Script {
  url "scripts/CPSCFloatToInt.class"
  eventIn  SFFloat  inFloat
  eventOut SFInt32  outInt
  field SFString info "this is CPSCFloatToInt script."
}
```

This script rounds the given float to the nearest integer value.

## CPSCBoolVec

```
DEF CPSCBoolVec Script {
  url "scripts/CPSCBoolVec.class"
  eventIn SFBool    inBool
  eventOut SFVec3f  outVec
  field SFVec3f     vector 0 0 0
  field SFString    info "this is CPSCBoolVec script."
}
```

When the script receives a "true" event, it returns a vector value specified in the vector field of the script. The default values are 0, 0, 0.

## CPSCBoolRot

```
DEF CPSCBoolRot Script {
  url "scripts/CPSCBoolRot.class"
  eventIn SFBool    inBool
  eventOut SFRotation outRot
  field SFRotation  rotate 0 0 1 0
  field SFString    info "this is CPSCBoolRot script."
}
```

When the script receives a "true" event, it returns a rotation specified in the rotate field of the script. The default values are 0, 0, 1, 0.

## CPSCEdge

```
DEF CPSCEdge Script {
  url "scripts/CPSCEdge.class"
  eventIn SFBool inBool
  eventOut SFBool outBool
  field SFString info "this is CPSCEdge script"
}
```

This script returns "true" or "false" alternatively whenever the input changes from "true" to "false" or vice versa. If the input value does not change (for example, keeps on giving "true"), the script does not return any value.

## CPSCStep

```
DEF CPSCStep Script {
  url "scripts/CPSCStep.class"
  eventIn SFBool inBool
  eventOut SFFloat outFloat
  field SFFloat lower 0.0
  field SFFloat upper 10.0
  field SFFloat step 1.0
  field SFString info "this is CPSCStep script"
}
```

When "true" is given to this script, it returns a float, starting from the value at field lower up to the value at field upper. Every time the value "true" is given to the script, it adds the amount specified in the field step and returns that value. It will stop adding step when the value exceeds the value specified at upper.

Take the default value for example. The first time it receives "true" the script returns 0.0. The next time it receives "true" it will return 1.0, then 2.0, and so on. After a while it will return 10.0, but if you add to this value, it would exceed the value specified at field upper. Hence, the calculation will stop here and the script will keep on returning 10.0 every time after that.

The value at fields upper, lower, and step, and the value that is being returned will also be printed out on the Java Console.

## CPSCUpper

```
DEF CPSCUpper Script {
    url "scripts/CPSCUpper.class"
    eventIn SFFloat inFloat
    eventOut SFBool outBool
    field SFFloat limit 5.0
    field SFString info "this is CPSCUpper script"
}
```

This script will return "true" if the value received exceeds the value specified in the field limit. If the value is less than or equal to limit, it will not return any values.

The default value given to the field limit is 5.0, which means that whenever a value bigger than 5.0 (for example 6.0) is given, the script will return "true".

## CPSCLower

```
DEF CPSCLower Script {
    url "scripts/CPSCLower.class"
    eventIn SFFloat inFloat
    eventOut SFBool outBool
    field SFFloat limit 5.0
    field SFString info "this is CPSCLower script"
}
```

This script will return "true" if the value received is less than the value specified at field limit. If the value is more than or equal to limit, the script will not return any values.

The default value given to the field limit is 5.0, which means that whenever a value smaller than 5.0 (for example 4.0) is given, the script will return "true".

## CPSCMatch

```
DEF CPSCMatch Script {
  url "scripts/CPSCMatch.class"
  eventIn SFFloat inFloat
  eventOut SFBool outBool
  field SFFloat value 5.0
  field SFString info "this is CPSCMatch script"
}
```

This script will return "true" if the value received matches the value specified at field value. If the received value does not equal to value, the script will not return any values.

The default value given to the field value is 5.0, which means that whenever a value 5.0 is given, the script will return "true".

## CPSCRange

```
DEF CPSCRange Script {
  url "scripts/CPSCRange.class"
  eventIn SFFloat inFloat
  eventOut SFBool outBool
  field SFFloat upperLimit 6.0
  field SFFloat lowerLimit 4.0
  field SFString info "this is CPSCRange script"
}
```

If the value given to the script is in the range of the values specified by the fields upperLimit and lowerLimit, the script returns "true". If the value does not exist between lowerLimit and upperLimit, the script will not return any values.

The default value given to the field lowerLimit and upperLimit is 4.0 and 6.0, which means that whenever a value between 4.0 and 6.0 (including 4.0 and 6.0) is given, the script will return "true".

## CPSCDelayGen

```
DEF CPSCDelayGen Script {
  url "scripts/CPSCDelayGen.class"
  eventIn  SFTIME    inTime
  eventOut SFTIME    outTime
  field    SFFloat   delay 5
  field    SFString  info   "this is CPSCDelayGen script"
}
```

When a SFTIME event is received, the script adds the float value given at field delay to the time received and returns that value. The default value set to the field delay is 5.0.

## CPSCShowBool

```
DEF CPSCShowBool Script {
  url "scripts/CPSCShowBool.class"
  eventIn  SFBool    inBool
  eventOut SFBool    outBool
  field    SFString  info   "this is CPSCShowBool script"
}
```

This script prints out the boolean value received on the Java Console, and returns that same value. This script is used to test if the correct value is being given to the next script.

## CPSCShowFloat

```
DEF CPSCShowFloat Script {
  url "scripts/CPSCShowFloat.class"
  eventIn  SFFloat   inFloat
  eventOut SFFloat   outFloat
  field    SFString  info   "this is CPSCShowFloat script"
}
```

This script prints out the float received on the Java Console, and returns that same value. This script is used to test if the correct value is being given to the next script.

## CPSCShowInt

```
DEF CPSCShowInt Script {
  url "scripts/CPSCShowInt.class"
  eventIn SFInt32 inInt
  eventOut SFInt32 outInt
  field SFString info "this is CPSCShowInt script"
}
```

This script prints out the integer received on the Java Console, and returns that same value. This script is used to test if the correct value is being given to the next script.

## CPSCShowVec

```
DEF CPSCShowVec Script {
  url "scripts/CPSCShowVec.class"
  eventIn SFVec3f inVec
  eventOut SFVec3f outVec
  field SFString info "this is CPSCShowVec script"
}
```

This script prints out the vector (SFVec3f) received on the Java Console, and returns that same value. This script is used to test if the correct value is being given to the next script.

## CPSCReturnBool

```
DEF CPSCReturnBool Script {
  url "scripts/CPSCReturnBool.class"
  eventIn SFBool inBool
  eventOut SFBool outBool
  field SFBool retBool TRUE
  field SFString info "this is CPSCReturnTrue script"
}
```

Whenever this script receives a boolean value, it returns the boolean defined in the field retBool.

## CPSCMoveOn

```
DEF CPSCMoveOn Script {
  url "scripts/CPSCMoveOn.class"
  eventIn SFFloat      inFloat
  eventOut SFVec3f     posChange
  eventOut SFRotation  oriChange
  field SFString info "this is CPSCMoveOn script"
}
```

This script defines the movement and rotation of an object. User can modify the java source to change the movements. The default movement is a sine wave on the x-y plane.

## CPSCMoveIt

```
DEF CPSCMoveIt Script {
  url "scripts/CPSCMoveIt.class"
  directOutput TRUE
  eventIn SFBool clicked
  field SFNode node NULL
  field SFString info "this is CPSCMoveIt script"
}
```

This script moves an object to a certain point (which will be defined in the script) every time it receives a TRUE value.

## CPSCMoveAlong

```
DEF CPSCMoveAlong Script {
  url "scripts/CPSCMoveAlong.class"
  directOutput TRUE
  eventIn SFVec3f position
  field SFNode node NULL
  field SFString info "this is CPSCMoveAlong script"
}
```

This script moves the node to a new position relative to the original position of the node. This script can be used with the Position Interpolator to move a node along the Interpolator.

## CPSCMoveTo

```
DEF CPSCMoveTo Script {
  url "scripts/CPSCMoveTo.class"
  directOutput TRUE
  eventIn SFFloat fraction
  eventIn SFVec3f position
  field SFNode node NULL
  field SFString info "this is CPSCMoveTo script"
}
```

When the script receives a new fraction, it moves the object defined in the field node, towards the specified position (which means that if no new value is given to position, it will not move).

Pay attention to the "Translation" field of the node. There might be several "Translation" fields in the node referred to by the script, and it might not move to the expected position if the correct "Translation" hasn't been selected. (Especially after moving the node by dragging onto the mouse. It creates a new "Translation" by doing so, and makes it difficult to determine what the combined transformation would be.)

## CPSCRotateAlong

```
DEF CPSCRotateAlong Script {
  url "scripts/CPSCRotateAlong.class"
  directOutput TRUE
  eventIn SFRotation    orientation
  field   SFNode        node NULL
  field SFString    info "this is CPSCRotateAlong script"
}
```

This script rotates the node to a new orientation relative to the original orientation of the node. This script can be used with the Orientation Interpolator to rotate a node along the Interpolator.

## CPSCRotateTo

```
DEF CPSCRotateTo Script {
  url "scripts/CPSCRotateTo.class"
  directOutput TRUE
  eventIn SFFloat      fraction
  eventIn SFRotation   orientation
  field   SFNode        node NULL
  field SFString    info "this is CPSCRotateTo script"
}
```

When the script receives a new fraction, it rotates the object defined in the field node, according to the specified orientation (which means that if no new value is given to orientation, it will not rotate).

## CPSCFace

```
DEF CPSCFace Script {
    url "scripts/CPSCFace.class"
    directOutput TRUE
    eventIn SFVec3f      inVec
    eventIn SFRotation  outRot
    field SFNode        node NULL
    field SFString      info  "this is CPSCFace script"
}
```

When the script receives a new vector in `inVec`, it rotates the object defined in the field `node`, so that it will face towards (or in case of a cone, it will point to) the point defined in the `inVec`.

You must be careful of the transform node of the object to rotate. You might have to form another transform node under the object in the hierarchy and fix its orientation in order for the script to rotate the object correctly.