



VrmlPad v. 1.3

User's Guide

Copyright © 1999-2001 ParallelGraphics

Contents

OVERVIEW.....	1
About VrmIPad	1
Product Features.....	1
New Features in VrmIPad 1.3.....	2
New Features in VrmIPad 1.2.....	3
Available Add-ins.....	4
MATERIAL EDITOR	4
Extrusion Editor	4
About VRML	5
Design Criteria.....	5
Characteristics of VRML.....	5
Scope	6
Introduction to the VrmIPad Automation	7
TEXT EDITOR	9
Overview: Text Editor	9
Syntax Coloring	9
AutoComplete.....	11
Error Processing.....	11
Node Folding	12
Finding Text.....	13
Regular Expressions	15
File Navigation.....	16
Using Drag-and-Drop Editing.....	17
Editor Commands and Keystrokes	17
SCENE TREE	20
Overview: Scene Tree	20
Scene Tree Navigation	21
Moving, Copying, and Cloning Nodes within the Scene Tree.....	22
Naming Nodes in the Scene Tree.....	22
ROUTING MAP.....	24
Overview: Routing Map	24
Managing Routes	24
RESOURCE VIEW.....	26
Overview: Resource View.....	26
Managing Resources.....	26
Using Drag-and-Drop in the Resource View	27
FILE LIST.....	29
Overview: File List	29
Using Drag-and-Drop in the File List.....	29
PUBLISHING WIZARD.....	31

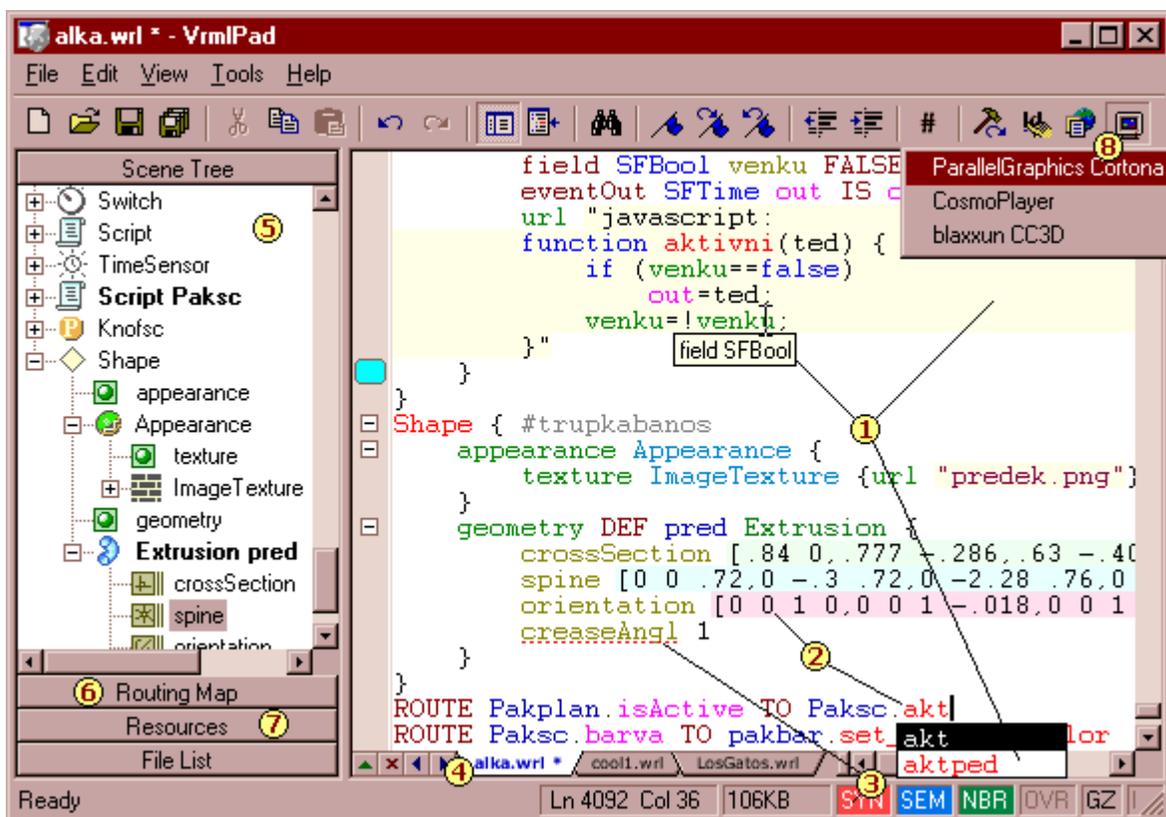
The Basics of Publishing	31
Setting Up the Destination	31
Choosing Additional Resource Directories.....	31
Excluding Files from Publishing	32
VRML Optimization Options	32
Specifying Directory Organization.....	33
Reviewing the Directory Structure	33
Previewing Published Documents	33
HINTS AND TIPS.....	34
How to compress a VRML file.....	34
How to download a VRML file from the Net.	34
How to upload a VRML file to a remote server.....	35
How to quickly find the definition (reference) of an identifier.	35
How to insert a pair of node or PROTO braces.....	35
How to insert default field value.	36
How to indent a block of lines.	36
How to comment a block of lines or an entire node.....	36
How to locate a syntax or semantic error.	36
How to make a VRML file smaller.	36
How to expose a field to a PROTO interface.	37
How to register a VRML extension.....	37
How to change a set of the standard nodes.....	38
How to change a source of the Node Help.	38
What are the ways I can access the VrmIPad object model?	39
How to automate VrmIPad from another application.	39

Overview

About VrmIPad

VrmIPad is a powerful and flexible authoring tool from ParallelGraphics that allows you to design and develop professional VRML content. Use VrmIPad to create VRML worlds for publishing on the World Wide Web. VrmIPad fully supports the VRML 97 specification.

Product Features



1 Editing – Access to local and remote files, multiple undo/redo, set bookmarks, advanced find and replace, syntax tips, smart AutoComplete, autoindent, dynamic node folding, customizable keyboard shortcuts.

2 Enhanced color-coding – Customizable by VRML keywords, field and node categories.

3 Dynamic errors detecting – View and enumerate syntax and wide range of semantic errors and warnings – undefined identifiers, nonmatched fields, duplicated node definitions and more.

4 Document Tabs – Open and edit multiple documents. Find and replace in all opened documents. Document list includes also file dependencies of the active document.

5 Scene Tree – View, edit and navigate hierarchical scene structure. Synchronize selection in both directions. Delete, comment and rename nodes, PROTOs and field declarations.

6 Routing Map – View and edit scene routes. Synchronize selection in both directions. Delete, comment and add new routes using **Start Route** and **Add Route** commands.

7 Resource View – View, edit and navigate document dependencies. Rename and browse all references to a resource. Convert, when possible, absolute references to relative and vice versa.

8 Preview – Preview scenes in installed browsers and VRML viewers.

9 Publishing Wizard (not shown on the figure) – Organize and optimize your scene with dependencies for publishing on the Net and put the files on your Web server or send it by e-mail.

10 Automation and Scripting (not shown on the figure) – Manipulate the program from external applications using the C, C++, Visual Basic®, Borland Delphi®. Create add-ins, VBScript and JavaScript macros and execute them from within the VrmIPad environment, automating custom tasks.

New Features in VrmIPad 1.3

- **Multiple documents**

Now you can open multiple files in the same instance of VrmIPad, select between them by using the tabbed Document Bar and Ctrl+Tab hotkey, find and replace text in all opened files. Using the pull-down Document List you can browse opened files and VRML-dependencies (inlines, anchors and EXTERNPROTOS) of the active document.

- **Routing Map**

The Routing Map displays routes of your scene and lets you browse and edit connections between events and exposed fields in your scene. Also you can more quickly add new routes to a scene using new **Add Route** dialog.

- **VRML extension node libraries**

Now you can use the latest VRML extension nodes from Web3D consortium, ParallelGraphics, Blaxxun, GeoVRML, etc. VrmIPad supports semantic and publishing rules for the extension nodes and optionally insert appropriate EXTERNPROTO declarations for these nodes. The list of node libraries can be updated online. Also you can use your own custom set of extension nodes. To turn on some node extension, select appropriate checkmark from the **Node libraries** list at the **General** tab of the **Options** dialog box and restart VrmIPad.

- **New automation objects**

Additional automation objects have been included: **Documents** and **VrmIMatrix**. The Documents collection represents all open workspace documents. The VrmIMatrix object provides many useful methods for performing manipulations on 4x4 matrices.

- **AutoComplete feature in inline scripts**

In addition to list interface fields and events of a Script node, the **Complete** command (Ctrl+Space) now shows subfields of *SFNode* and *MFNode* fields, if possible, and lists names of standard VRML-script objects, properties and methods.

- **Line numbering**

Additional margin at the left side of Text editor displays line number for every line of a document. To turn on line numbering, select **Show line numbers** checkmark at the **Editor** tab of the **Options** dialog box.

- **More Unused Identifiers**

Now the **Unused Identifiers** dialog box shows also redundant (duplicated) field definitions and duplicated routes.

- **Improved performance**

VrmlPad now parses, formats and indents/unindents text faster than previous version does.

See also: "New Features in VrmlPad 1.2".

New Features in VrmlPad 1.2

- **Automation and Scripting**

From this version VrmlPad provides **VrmlPad Object Model** that enables authors and developers to manipulate the program from external applications using the C, C++, Visual Basic®, Borland Delphi®. In addition, power users can now create add-ins, VBScript and JavaScript macros and execute them from within the VrmlPad environment, automating custom tasks and adding visual support. VrmlPad macros are procedures you write in the VBScript or JavaScript language, and add-ins are in-process ActiveX components (DLLs) you write in C, C++, Delphi or Visual Basic.

- **Node Folding**

Node folding lets you hide a body of huge nodes and prototype declarations. Also you can define a set of uninteresting nodes that collapses automatically on scene load.

- **Preview using multiple VRML viewers**

In addition to preview a scene using default Web browser with a default VRML viewer, now you can choose a viewer from a list of installed in the system VRML viewers. To do this, From the Tools menu, choose Options and select the **Preview** tab. Then select **Show a choice...** option.

- **Keyboard shortcuts customization**

On the **Shortcut Keys** page of the **Options** dialog you can assign custom shortcut key combinations to all VrmlPad commands (including macro and add-in commands).

- **Inlining resources**

Now publishing wizard can include resources into VRML file body as base64 encoded data, excluding them from the list of publishing files. VRML files with in-lined resources are currently supported only by ParallelGraphics Cortona VRML viewer.

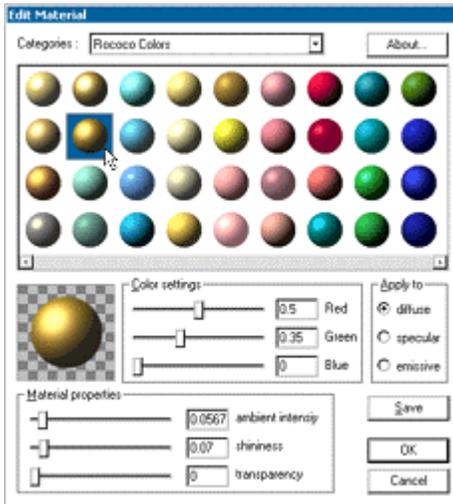
- **Inserting Anchors and External prototypes**

In addition to inserting Inline nodes as a result of drag-and-drop VRML files from File List or Resource View, now it's also possible to insert Anchor nodes and EXTERNPROTO declarations.

See also: "Using Drag-and-Drop in the File List".

Available Add-ins

Material Editor

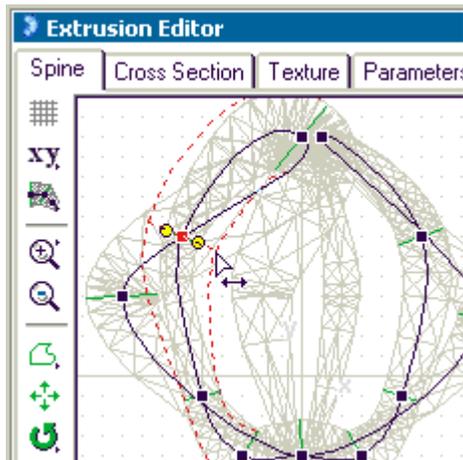


Two optional components have been included in the VrmIPad installation pack, as examples of VrmIPad automation and scripting, Material Editor add-in and sample VBScript macros.

The Material Editor lets you create and edit Appearance and Material nodes. You also can choose existing materials from a pre-built library and add your own materials to it.

Sample macros include useful commands for enumerating faces in IndexedFaceSet nodes, completing selected nodes with all fields, and converting the Box, Cone and Cylinder nodes to an IndexedFaceSet etc.

Extrusion Editor



Extrusion Editor, a visual plug-in for VrmIPad, offers you an effective method for creating and editing existing extrusion models.

Extrusion Editor is not included in the VrmIPad installation pack. The trial version is available from the Web at the following location:

<http://www.parallelgraphics.com/products/vrmipad/extrusioneditor/download/>

Key Features

- Drawing a series of connected vertices one by one;
- Drawing Bezier curves (spline curves);
- The use of pre-defined forms (rectangle, diamond, octagon, triangle, oval, etc.);
- Move, scale, flip and rotate the spine and cross-section;
- Browse for, move, scale, and rotate the texture;
- Assemble extrusions in complex models and perform editing operations on the group of shapes;
- Manipulate extrusion parameters, edit the material, and control the smoothness of the surface;

- Immediate preview changes in the 3D window;
- Add new extrusions to the library of extrusions and insert their code into VrmIPad's text editor;
- Full VRML97-compliance.

Limitations in the trial version:

- Thirty (30) days evaluation period;
- You cannot save an extrusion (or a group of extrusions) if the number of segments in the spine is over 15.

See also: <http://www.parallelgraphics.com/products/vrmlpad/extrusioneditor/>

About VRML

The Virtual Reality Modeling Language (VRML) is a file format for describing interactive 3D objects and worlds. VRML is designed to be used on the Internet, intranets, and local client systems. VRML is also intended to be a universal interchange format for integrated 3D graphics and multimedia. VRML may be used in a variety of application areas such as engineering and scientific visualization, multimedia presentations, entertainment and educational titles, web pages, and shared virtual worlds.

Design Criteria

VRML has been designed to fulfill the following requirements:

- **Authorability**
Enable the development of computer programs capable of creating, editing, and maintaining VRML files, as well as automatic translation programs for converting other commonly used 3D file formats into VRML files.
- **Composability**
Provide the ability to use and combine dynamic 3D objects within a VRML world and thus allow re-usability.
- **Extensibility**
Provide the ability to add new object types not explicitly defined in VRML.
- **Implementability**
Capable of implementation on a wide range of systems.
- **Performance**
Emphasize scalable, interactive performance on a wide variety of computing platforms.
- **Scalability**
Enable arbitrarily large dynamic 3D worlds.

Characteristics of VRML

VRML is capable of representing static and animated dynamic 3D and multimedia objects with hyperlinks to other media such as text, sounds, movies, and images. VRML browsers, as well as authoring tools for the creation of VRML files, are widely available for many different platforms.

VRML supports an extensibility model that allows new dynamic 3D objects to be defined and a registration process that allows application communities to develop interoperable extensions to the base standard. There are mappings between VRML objects and commonly used 3D application programmer interface (API) features.

Scope

The VRML specification defines a file format that integrates 3D graphics and multimedia. Conceptually, each VRML file is a 3D time-based space that contains graphic and aural objects that can be dynamically modified through a variety of mechanisms. VRML defines a primary set of objects and mechanisms that encourage composition, encapsulation, and extension.

The semantics of VRML describe an abstract functional behaviour of time-based, interactive 3D, multimedia worlds. VRML does not define physical devices or any other implementation-dependent concepts (e.g., screen resolution and input devices). VRML is intended for a wide variety of devices and applications, and provides wide latitude in interpretation and implementation of the functionality. For example, VRML does not assume the existence of a mouse or 2D display device.

Each VRML file:

1. implicitly establishes a world coordinate space for all objects defined in the file, as well as all objects recursively included by the file;
2. explicitly defines and composes a set of 3D and multimedia objects;
3. can specify hyperlinks to other files and applications;
4. can define object behaviors.

An important characteristic of VRML files is the ability to compose files together through inclusion and to relate files together through hyperlinking. For example, consider the file `earth.wrl` which specifies a world that contains a sphere representing the earth. This file may also contain references to a variety of other VRML files representing cities on the earth (e.g., file `paris.wrl`). The enclosing file, `earth.wrl`, defines the coordinate system that all the cities reside in. Each city file defines the world coordinate system that the city resides in but that becomes a local coordinate system when contained by the earth file.

Hierarchical file inclusion enables the creation of arbitrarily large, dynamic worlds. Therefore, VRML ensures that each file is completely described by the objects and files contained within it and that the effects of each file are strictly scoped by the file and the spatial limits of the objects defined in the file. Otherwise, the accumulation of files into larger worlds would produce unscalable results (as each added world produces global effects on all other worlds). For example, light sources have the potential of global effect since light energy theoretically does not dissipate to zero. And, if the earth file contains 100 city files each containing 100 lights each affecting all objects in the world, the lighting calculations would quickly become intractable. Therefore, in order to prevent global effects, light source objects are scoped by either a maximum radius or by location within the file.

Another essential characteristic of VRML is that it is intended to be used in a distributed environment such as the World Wide Web. There are various objects and mechanisms built into the language that support multiple distributed files, including:

5. in-lining of other VRML files;
6. hyperlinking to other files;
7. using established Internet standards for other file formats;
8. defining a compact syntax.

See also: <http://www.vrml.org/Specifications/VRML97>

Introduction to the VrmlPad Automation

The VrmlPad Automation provides application programming interface (API) that enables authors and developers to manipulate the program from external applications using the C, C++, Visual Basic®, Delphi, and scripts within HTML pages.

In addition, power users can now create add-ins, VBScript and JavaScript macros and execute them from within the VrmlPad environment, automating custom tasks. VrmlPad macros are procedures you write in the VBScript or JavaScript language, and add-ins are in-process ActiveX components (DLLs) you write in C, C++, Pascal or Visual Basic. Each add-in can add one or more commands, and each command can perform one or more tasks. The number of commands and tasks is up to you.

If your tasks are routine or repetitive, you can benefit from automating them. Through Automation (formerly OLE Automation), you can reduce the time spent on these tasks, and you can prevent errors that often result from performing the tasks manually. Also, you can add visual support for editing some VRML nodes in addition to text editing provided by VrmlPad.

With Automation, you perform tasks by manipulating VrmlPad and its components as objects. For example, you open, edit, and close a document by manipulating it as an object. Similarly, you get or edit a VRML node's properties by manipulating it as an object. Each VrmlPad object implements a dual interface through which you can manipulate the object.

You manipulate objects by using methods, properties, and events associated with the objects. Methods represent actions you take against the objects. Properties represent characteristics of the objects, such as their type or size. And, events represent conditions under which actions are taken against the objects.

Suppose you want to create a VRML file, containing WorldInfo node, quickly — by simply choosing a menu item. Start by writing a VBScript macro that will create the file. The macro looks like this:

```
Sub CreateNew
Document.NewFile
Document.Selection = "#Created by the Author" + vbCrLf
Set node = Document.RootNodes.Add("WorldInfo")
node.Fields("info").Add "Powered by VrmlPad"
End Sub
```

This macro does the following:

- The **Sub** statement begins the macro, which has the name **CreateNew**. The name is arbitrary, you can use whatever name you choose.
- The first line of the macro clears the file, using the **New File** method of the **Document** object.
- The second line of the macro adds a comment string and moves the caret to the next line, using the **Selection** property of the **Document** object.
- The next line of the macro adds an empty *WorldInfo* node, using the **RootNodes** collection of the **Document** object.
- The next line of the macro adds the *info* field to the node, using the **Fields** collection of the **Node** object, and sets field's value, using the **Add** method of the **Field** object.

The **End Sub** statement ends the macro.

After writing the macro, copy the macro file to the AddIns directory. Then, select the menu item **CreateNew** from the **Tools** menu to run the macro.

For more information about VrmIPad objects and writing add-ins and macros, see <http://www.parallelgraphics.com/developer/products/vrmlpad>

Text Editor

Overview: Text Editor

The VrmlPad environment includes an integrated text editor to manage, edit, and print source files. Most of the procedures for using the editor should seem familiar if you have used other Windows-based text editors. With the Text editor, you can:

- Set and customize syntax coloring for source VRML files.
- Use AutoComplete for quick entering a VRML keyword, node type, node name, field name, default field value or another syntax element.
- Perform advanced find and replace operations in a file, including using regular expressions.
- Use virtual spaces for advanced cursor positioning.
- Navigate through sections of code using the **Go To** dialog box.
- Use Bookmarks to mark frequently accessed lines in your source file.
- Customize the Text editor with save preferences, tabs, and indents.
- Modify the font style, size, and color.
- Select lines or multiple lines, copy and cut selection into clipboard.
- Split the Text editor window into two panes.
- Hide a body of huge or uninteresting nodes and prototype declarations.
- Use drag-and-drop editing within editor window, and between the Text editor and other applications.

Tip: While using the Text editor, in many instances you can click the right mouse button to display a shortcut menu of frequently used commands. The commands available depend on what the mouse pointer is pointing to.

Syntax Coloring

Syntax coloring uses different colors for various code elements, such as keywords, comments, strings, URLs, nodes, fields and field values. This coloring gives you visual cues about the structure and state of your code.

```
PROTO Message [
  field SFNode console NULL
  field MFString text []
  eventIn SFTIME input
]{
  Script {
    field MFString text IS text
    field SFNode console IS console
    eventIn SFTIME input IS input
    url "javascript:
function input (val) {
  console.text = text;
  console.startTime = val;
}"
  }
}
```

► **To change colors used for syntax coloring**

1. From the Tools menu, choose Options.
2. Select the **Format** tab.
3. In the **Font** box, select the font you want.
4. The **Font** box displays the different fixed-pitch fonts installed on your system. The text sample in the **Sample** box changes to the font you select.
5. In the **Size** box, select the size to be used with the font you selected.
The **Size** box displays the sizes available for the selected font. The text sample in the **Sample** box changes to the size you select.
6. In the **Colors** box, select the type of text you want to color.
7. In the **Background** box, select a background color.
8. In the **Foreground** box, select a foreground color.

The **Background** and **Foreground** boxes display the standard colors, the **Custom** setting and the **Automatic** setting. The text sample displayed in the **Sample** box changes to the color you select.

The behavior of the **Automatic** setting depends on the element selected. For colors that map to standard system elements (such as **Text** or **Selected Text**), the **Automatic** setting sets the element to the appropriate system color. For syntax coloring elements and other non-system defined colors, the **Automatic** setting indicates that the foreground color or background color from the parent category is to be used.

9. Click **OK**.

► **To disable syntax coloring**

1. From the Tools menu, choose Options.
2. Select the **Editor** tab.
3. In the **Options** box, unselect **Highlight language syntax** checkmark.
4. Click **OK**.

AutoComplete

The **Complete** command (Ctrl+Space) opens a dropdown list box in the Text editor that contains the VRML identifiers (keywords, PROTO, node and field names, fields types) appropriate to the grammatical context at the current caret position. To have the list box automatically open as you type your code, select **Auto list identifiers** on the **Editor** tab in the **Options** dialog box.

You can find the identifier you want in the list box by:

- Typing the name.
As you type, the identifier that matches the characters you type is selected and moves to the top of the list.
- Using the up and down arrow keys to move up and down in the list.
- Scrolling through the list and selecting the identifier you want.

You can insert the identifier into your code by:

- Clicking the identifier.
- Selecting the identifier and pressing SPACE or ENTER to insert the selection.

```

    DEF TOUCH TouchSensor {
        enabled IS soundEnabled
    }
    color
1  ColorInterpolator
ROU CoordinateInterpolator 0.startTim
# Tran CylinderSensor
DEF
EF PIC DirectionalLight
ever Fog
ever Group
field Inline
field ISBPicture
field LOD
field SFInt32 nActiveFrame -1
itchTime
intDummy
eCount
ayOrder

```

You can also type Ctrl+Space after a node or PROTO identifier. The command adds braces after identifier, inserts empty line and sets there the caret.

After a field declaration or definition the command inserts default field value.

Error Processing

As you type, VrmIPad can automatically check your document and underline syntax and semantic errors or possible warnings. There are three types of errors indicated on the status bar at the bottom of the VrmIPad window:

- Syntax errors.
Wrong construction underlines by red wavy line and indicates on the status bar by a red pane labeled **SYN**.
- Semantic errors and warnings. Missing, duplicated or type-mismatched identifiers.
Identifier underlines by red dash line and indicates on the status bar by a blue pane labeled **SEM**.

- Nonmatching braces, square brackets and double quotes.
Status bar indicates this error by a green pane labeled **NBR**.

```

    ]
    ROUTE PS.position_changed TO TR.set_rotation
    ROUTE PS.orientation_changed TO TR.
    ROUTE TM.isActive TO SC.hit
Expected routed field name Active TO PS.enabled
    ROUTE SC.choice TO SW.whichChoice

```

Type of routed values does not match | Ln 93 Col 37 | 1.56MB | SYN SEM NBR

► To get explanation for an error:

- Rest the mouse pointer over the wrong construction for a moment and read a ScreenTip for the error.
- Double-click the highlighted pane on the status bar. The caret will move to the first error of that type.

To quickly move to the next error, choose **Next Error** from the **View** menu, or type F4 (to go to the previous error, type Shift+F4).

Node Folding

Node folding lets you hide a body of huge or uninteresting nodes and prototype declarations. The lines of code inside these nodes can be expanded or collapsed using a [+]/[-] icon on the selection margin at the left side of Text editor.

```

[-] Shape {
    appearance Appearance {
        texture ImageTexture {
            url "e3u3/metal14_1.gif"
        }
    }
    [+ | geometry IndexedFaceSet { # 98 faces
    }
}

```

► To expand or collapse a node fold

- Click the + icon to expand the fold or – icon to collapse.
- Or set the caret at the folded lines and from the **Edit / Fold** submenu, choose **Expand** or **Collapse** or press **Alt+Down Arrow** or **Alt+Up Arrow**.
- Or right-click the fold sign and from the pull-down menu choose **Toggle**.

► To expand or collapse all node folds

- From the **Edit / Fold** submenu, choose **Expand All** or **Collapse All**.
- Or right-click any fold sign and from the pull-down menu choose **Expand All** or **Collapse All**.

By default, a number of lines that can be folded should be at least 20 lines.

► To change the minimum number of lines to fold

1. From the **Tools** menu, choose **Options**.
2. Select the **Node Folds** tab.
3. In the **Minimum length of node to fold** box, type the number of lines.
Tip: Type **0** lines to fold all nodes and prototype declarations.
4. Click **OK**.

You can define a set of node types that should be hidden normally. All nodes of such uninteresting types can be collapsed manually or automatically (on scene load or text insertion).

► **To define a set of uninteresting nodes**

1. From the **Tools** menu, choose **Options**.
2. Select the **Node Folds** tab.
3. Using **Add >>** and **<< Remove** buttons, modify a list of nodes in the **Uninteresting nodes** box.
4. To automatic collapse uninteresting nodes, select **Automatically collapse uninteresting nodes on load and inserts** checkbox.

► **To disable node folding**

1. From the **Tools** menu, choose **Options**.
2. Select the **Editor** tab.
3. In the **Options** box, unselect **Allow node folding** checkmark.
4. Click **OK**.

Finding Text

With the advanced find and replace capabilities of the Text editor, you can search for literal text strings or use regular expressions to find words or characters. A regular expression is a search string that uses special characters to match a text pattern in a file. You can use regular expressions, including tagged regular expressions, with both the **Find** and **Replace** commands.

With the **Find** and **Replace** commands, you can:

- Find text in a document.
- Replace text in a whole document.
- Replace text in a part of a document.
- Use regular expressions.

► **To find a text string**

1. Move the insertion point to where you want to begin your search.
The editor uses the location of the insertion point to select a default search string.
2. From the **Edit** menu, choose **Find** or type Ctrl+F.
3. In the **Find what** box, type the search text or a regular expression.

Tip: Select the menu button to the right of the box to display a list of regular search expressions. When you select an expression from this list, the expression is substituted as the search text in the **Find what** box. If you do use regular expressions, be sure the **Regular expression** check box is selected. You can also use the drop-down list to select from a list of up to 16 previous search strings.

4. Select any of the **Find** options.
5. Start the search by clicking the **Find Next** or **Mark All** buttons.

▶ **To start a find without using the Find dialog box**

- Type Ctrl+F3 to search a word under the caret position.
- To continue your search, type the **Find Next** (F3) or **Find Previous** (Shift+F3) buttons.

▶ **To replace text**

1. Move the insertion point to where you want to begin your search.
2. The editor uses the location of the insertion point to select a default search string.
3. From the **Edit** menu, choose **Replace**.
4. In the **Find what** text box, type the search text or a regular expression.

Tip: Click the button to the right of the box to display a list of regular search expressions. When you select an expression from this list, the expression is substituted as the search text in the **Find what** text box. If you do use regular expressions, be sure the **Regular expression** check box is selected. You can also use the drop-down list to select from up to 16 previous search strings.

5. In the **Replace with text** box, type the replacement text.

Tip: Click the button to the right of the box to display a list of tagged expressions you can use as replacement text.

6. Select any of the remaining **Find** options.
7. Start the search by clicking the **Find Next**, **Replace**, or **Replace All** buttons.

Regular Expressions

Regular Expression	Description
.	Any single character
[]	Any one of the characters contained in the brackets, or any of an ASCII range of characters separated by a hyphen (-). For example, b[aeiou]d matches bad, bed, bid, bod, and bud, and r[eo]+d matches red, rod, reed, and rood, but not reod or roed. x[0-9] matches x0, x1, x2, and so on. If the first character in the brackets is a caret (^), then the regular expression matches any characters except those in the brackets.
^	The beginning of a line.
\$	The end of a line.
()	Indicates a tagged expression to retain for replacement purposes. If the expression in the Find what text box is (PROTO)OldName, and the expression in the Replace With box is \1NewName, all selected occurrences of PROTO OldName are replaced with PROTO NewName. Each occurrence of a tagged expression is numbered according to its order in the Find what text box, and its replacement expression is \n, where 1 corresponds to the first tagged expression, 2 to the second, and so on. You can have up to nine tagged expressions.
	Any one of the characters separated by the alternation symbol (). For example, (j u)+fruit finds jfruit, jfruit, ufruit, ufruit, and so on.
*	None or more of the preceding characters or expressions. For example, ba*c matches bc, bac, baac, baaac, and so on.
+	At least one or more of the preceding characters or expressions. For example, ba+c matches bac, baac, baaac, but not bc.
?	None or one of the preceding characters or expressions. For example, ba?c matches bc, and bac.
{ }	Any sequence of characters between the escaped braces. For example, (ju)+fruit finds jfruit, jjufruit, jujufruit, and so on. Note that it will not find jfruit, ufruit, or ujfruit, because the sequence ju is not in any of those strings.
[^]	Any character except those following the caret (^) character in the brackets, or any of an ASCII range of characters separated by a hyphen (-). For example, x[^0-9] matches xa, xb, xc, and so on, but not x0, x1, x2, and so on.
\a	Any single alphanumeric character [a – zA – Z0 – 9].
\w+	Any white-space characters. The \w+ finds tabs and spaces.
\c	Any single alphabetic character [a – zA – Z].
\d	Any decimal digit [0 – 9].
\v	Any VRML identifier.
\s	Any quoted string ("^[^"]*").
\	Removes the pattern match characteristic in the Find what text box from the special characters listed above. For example, 100\$ matches 100 at the end of a line, but 100\\$ matches the character string 100\$ anywhere on a line.

File Navigation

The Text editor provides a variety of methods to move around in a source file. In addition to the standard Windows navigation mechanisms, the Text editor includes an assortment of commands that enable you to move to almost any location in a file. Further, in addition to comprehensive **GoTo** and **Bookmarks** commands, the Text editor includes several advanced navigation features, such as virtual space.

About Virtual Space

All editors support moving the cursor by one character position. The most common difference among text editors is whether you can move the cursor into a location that does not currently contain text. For example, if your cursor is on column 20 and there is no text on the line below the current line, moving the cursor down can do one of two things: Either the cursor moves to column 1 — because there is no text on the line below — or the cursor moves to column 20 of the next line. This behavior is called virtual space.

With the Text editor, you can treat text selection and space insertion in two ways. When you select the **Virtual Spaces** option, spaces are inserted between the end of the line and the insertion point before new characters are added to the line. When you clear the **Virtual Spaces** option, the Text editor behaves like Microsoft Word for Windows — the insertion point is set to the end of the line.

▶ To enable virtual space

1. From the **Tools** menu, choose **Options**.
2. Select the **Editor** tab.
3. In the **Options** box, select **Enable virtual space** checkmark.
4. Click **OK**.

About Go To

The **Go To** dialog box allows you to jump quickly to several different items in a file, including:

- Lines (type the line number)
- Lines in inline scripts (type the line number from the beginning of the script and scripts node name)
- PROTO, node and field declarations (type or select the name of a PROTO, node, or field to go to where it is defined)
- PROTO, node and field references (type or select the name of a PROTO, node, or field to go to where it is referenced)
- ROUTEs (type or select the known node and field names to go to where it is routed)

▶ To use the Go To dialog box

1. From the **Edit** menu, choose **Go To**.
2. In the **Go To what** box, select the type of item you want.
3. Enter any additional information required.
4. Click one of the navigation buttons: **Go To**, **Previous**, or **Next**.

Tip: To quickly jump to definition of an identifier (PROTO, node, or field name) under the caret position, type Ctrl+F11. To jump to an identifier reference, type F11.

About Bookmarks

You can set bookmarks to mark frequently accessed lines in your source file. Once a bookmark is set (Ctrl+F2), you can use menu or keyboard commands (F2 and Shift+F2) to move to it. You can remove a bookmark when you no longer need it (Ctrl+F2 again).

Using Drag-and-Drop Editing

Drag-and-drop editing is the easiest way to move or copy a selection of text within a file, between files, or between applications. The text you drop remains selected, which makes it easy to copy a chunk of text into several places.

► To move text using drag-and-drop editing

1. Select the text you want to move.
2. Drag the selected text to the new location.

Note: You can also use the right mouse button for drag-and-drop editing. Select the text you want, and then use the right mouse button to drag the text to a new location. A shortcut menu appears, asking if you want to move or copy the selected text.

Tip: At any time during a drag-and-drop procedure, you can click the other mouse button to cancel the operation.

► To copy text using drag-and-drop editing

1. Select the text you want to copy.
2. While holding down the CTRL key, drag the selected text to the new location.

► To move or copy text to another document

1. Select the text you want to move or copy.
2. Drag the selected text to the document bar and hold it over the document tab.
3. After the target document selected drag the text to the new location.

Editor Commands and Keystrokes

You can access Text editor commands in many ways: from the menus, from the toolbar, and from the shortcut menu. There are many Text editor commands, but not all of them appear on the menus and toolbars. The **Selection** menu (available from the **Edit** menu) contains some of the most useful edit commands, including **Format**, **Indent**, **Unindent**, **Tabify**, **Untabify**, **Make Uppercase**, **Make Lowercase**, and **Comment**.

Use these key combinations in the Text editor window:

Press	To
F1	Get context-sensitive Help on node type under the caret position.
F2	Moves to the line containing the next bookmark.
Shift+F2	Moves to the line containing the previous bookmark.
Ctrl+F2	Toggles a bookmark for the current line on and off.
F3	Finds the next occurrence of the specified text.
Shift+F3	Finds the previous occurrence of the specified text.
Ctrl+F3	Finds the next occurrence of the selected text.
Ctrl+Shift+F3	Finds the previous occurrence of the selected text.
Alt+F3	Finds the specified text.
F4	Moves to the line containing the next error or warning.
Shift+F4	Moves to the line containing the previous error or warning.
Ctrl+F4	Closes the active document.
F5	Preview the document, using default Web browser and VRML viewer.
F11	Displays an identifier definition.
Shift+F11	Displays an identifier reference.
Ctrl+A	Selects the entire document.
Ctrl+C	Copies the selection to the Clipboard.
Ctrl+Shift+C	Comment or uncomment the selected lines.
Ctrl+E	Marks a node or a field at the caret position as a route source.
Ctrl+F	Activates the Find tool.
Ctrl+Shift+F	Formats the selection using the smart indent settings.
Ctrl+G	Moves to a specified location.
Ctrl+L	Deletes the selected lines and puts them on the Clipboard.
Ctrl+R	Marks a node or a field at the caret position as a route target and opens Add Route dialog box.
Alt+S	Synchronize the caret position with a selection in the Scene Tree.
Ctrl+Shift+T	Replaces spaces with tabs in the selection.
Ctrl+Alt+T	Shows or hides tab characters.
Ctrl+U	Makes the selection all lowercase
Ctrl+Shift+U	Makes the selection all uppercase.
Ctrl+V	Inserts the Clipboard contents at the insertion point.
Ctrl+X	Cuts the selection and moves it to the Clipboard.
Ctrl+Y	Redoes the previously undone action.
Ctrl+Z	Undoes the last action.
Ctrl+]	Finds the matching brace.
Ctrl+Shift+]	Extends the selection to the matching brace.
Ctrl+Space	Opens a dropdown list box that contains the available identifiers.
Ctrl+Shift+Space	Replaces tabs with spaces in the selection.
Tab	Indents the selection.
Shift+Tab	Unindents the selection.
Ctrl+Tab	Activates next document.

Ctrl+Shift+Tab	Activates previous document.
Backspace	Deletes the selection or, if there is no selection, the character to the left of the cursor.
Ctrl+Backspace	Deletes a word to the left.
Del	Deletes the selection.
Ctrl+Del	Deletes a word to the right.
Left Arrow	Moves the cursor one character to the left.
Shift+Left Arrow	Extends the selection one character to the left.
Ctrl+Left Arrow	Moves back one word.
Ctrl+Shift+Left Arrow	Extends the selection back one word.
Right Arrow	Moves the cursor one character to the right.
Shift+Right Arrow	Extends the selection one character to the right.
Ctrl+Right Arrow	Moves forward one word.
Ctrl+Shift+Right Arrow	Extends the selection forward one word.
Up Arrow	Moves the cursor up one line.
Shift+Up Arrow	Extends the selection up one line.
Ctrl+Up Arrow	Scrolls the file contents down one line.
Alt+Up Arrow	Collapse the node fold at the cursor
Down Arrow	Moves the cursor down one line.
Shift+Down Arrow	Extends the selection down one line.
Ctrl+Down Arrow	Scrolls the file contents up one line.
Alt+Down Arrow	Expand the node fold at the cursor
Home	Moves to either the start of the current line or the start of the text on that line.
Shift+Home	Extends the selection to either the start of the current line or the start of the text on that line.
Ctrl+Home	Moves to the beginning of the file.
Ctrl+Shift+Home	Extends the selection to the beginning of the file.
End	Moves to the end of the current line.
Shift+End	Extends the selection to the end of the current line.
Ctrl+End	Moves to the end of the document.
Ctrl+Shift+End	Extends the selection to the end of the document.
Page Down	Moves the cursor down one page.
Page Up	Moves the cursor up one page.
Shift+Page Down	Extends the selection down one page.
Shift+Page Up	Extends the selection up one page.
Ins	Toggles between inserting and replacing text.

Scene Tree

Overview: Scene Tree

The Scene Tree displays VRML hierarchy of your scene and lets you:

- Browse and edit the VRML hierarchy of your scene.
- Edit node, PROTO, and field names.
- Make precise selections within the Text Editor.

For each node, the Scene Tree displays:

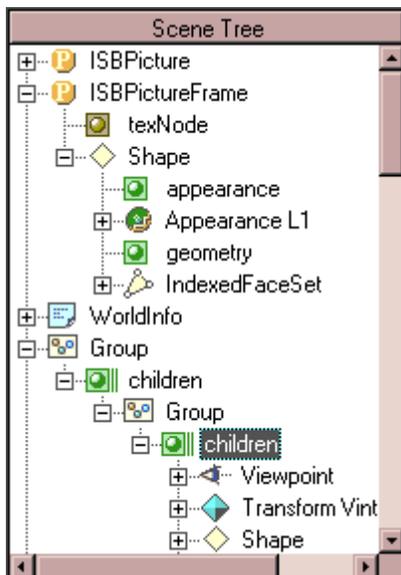
- An icon indicating the node type (for Transform , for Shape , and so on).
Each node type has a particular icon that is associated with it. Clones (multiple instances) of a node are indicated with the link mark  above the node icon.
- The name of the node type (Transform for instance).
- The name (if any) that you've given to the node.

For each PROTO interface declaration, the Scene Tree displays:

- An PROTO icon .
- The name of the PROTO.

For each field declaration or definition, the Scene Tree displays:

- An icon indicating the field type and category (for exposedField SFNode , for eventIn SFFloat , and so on).
ISed fields are indicated with the link mark  above the field icon.
- The name of the field (children for instance).



To show only nodes and PROTOs in the Scene Tree, select **Show nodes only** on the **Tree** tab in the **Options** dialog box.

Scene Tree Navigation

You can see various levels of the hierarchy contained under nodes and PROTOs by clicking the node icon. Opening a node refers to expanding the hierarchy beneath it. Closing a node refers to collapsing the hierarchy beneath it. You can also click a MFNode field icon to toggle it open and see its values.

▶ **To open or close a node, PROTO, or field:**

- Click the + icon to open the item.
- Click the – icon to close the item.
- Type the * button to open all subitems of the selected item.

▶ **To jump to the definition of the selected item (PROTO-instance, nodes clone, or field):**

- Choose **Go To Definition** from the pull-down menu (right-click the item to see the menu).
- Or press Ctrl+F11.

▶ **To jump to the reference of the selected item (PROTO-declaration, node, or field declaration):**

- Choose **Go To Reference** from the pull-down menu.
- Or press F11.

You can synchronize current selection in the Scene Tree with the caret position in the Text Editor in both directions.

▶ **To set the caret position in the Text editor at a node, PROTO or field:**

- Double-click the item.
- Or select the item and press ENTER.
- Or choose **Go To** from the pull-down menu.

▶ **To set the selection in the Scene Tree to the node, PROTO or field near the caret position:**

1. Activate Text editor window.
2. Choose **Synchronize** from the **View** menu (or type Alt+S).

To have the Scene Tree automatically synchronize selection as you type your code, select **Automatic synchronize context** on the **Tree** tab in the **Options** dialog box.

Moving, Copying, and Cloning Nodes within the Scene Tree

The top-level nodes in a VRML file are in no particular order; so you can reorder them to suit your tastes without changing what your world looks like. You can also move a node (other than a top-level node) or field declaration/definition to reorganize the structure of your world.

▶ To move a node (field) in the scene hierarchy:

1. Select the node (field) you want to move.
2. Click and drag the node (field) icon. When the pointer is over a place where the item can legally be inserted, two red arrows appear; if you release the mouse button at that spot, the node (field) is inserted there.

Note: When the pointer is over a closed grouping node or a MFNode field, rest the mouse pointer over the item for a second and the item will open.

▶ To copy a node in the scene hierarchy:

1. Select the node you want to copy.
2. While holding down the CTRL key, drag the selected node to the new location.

▶ To clone a node in the scene hierarchy:

1. Select the node you want to clone.
2. While holding down the ALT key, drag the selected node to the new location below current selection.

Note: If the dragged node is unnamed, Scene Tree prompts you for the node name.

If you use your right mouse button to drag, a menu appears with the available options.

Naming Nodes in the Scene Tree

Using the Scene Tree, you can rename a node, PROTO declaration or field declaration with appropriate renaming of all its instances (clones).

▶ To edit the node name using the Scene Tree:

1. Select the node you want to rename.
2. Press F2 or right-click over the node and choose **Rename**.
3. Type a name in the text box and press Enter or press Esc if you want to leave the text box unchanged.

Note: VRML node names are not allowed to contain spaces, single or double quotation marks, pound signs (#), commas, periods, square brackets, backslashes, or curly braces. Also, they can't begin with a numerical digit or the plus (+) or minus (-) characters.

▶ To clear the node name:

1. Select the node you want to unname.
2. Press F2 or right-click over the node and choose **Rename**.
3. Clear the text box and press Enter.

Note: If the node name is in use, the Scene Tree warns you about it.

▶ **To rename a PROTO (with all PROTO-instances of this PROTO):**

1. Select the PROTO declaration you want to rename.
2. Press F2 or right-click over the PROTO and choose **Rename**.
3. Type a name in the text box and press Enter or press Esc if you want to leave the text box unchanged.

▶ **To rename a field declaration (with all references to this field):**

1. Select the field declaration you want to rename.
2. Press F2 or right-click over the field and choose **Rename**.
3. Type a name in the text box and press Enter or press Esc if you want to leave the text box unchanged.

Note: To prevent renaming field references inside the inlined vrml-script body, select **Disable renaming in a script body** check box on the **Tree** tab in the **Options** dialog box.

▶ **To remove a node, PROTO or field declaration**

1. Select the item you want to remove.
2. Press Del or right-click over the item and choose **Delete**.

Note: To remove also all references to this item (node and PROTO instances, routes, PROTO field implementations) and items within this item, press Ctrl+Del key or choose **Delete** from the pull-down menu while holding down the CTRL key.

Routing Map

Overview: Routing Map

The Routing Map displays routes of your scene and lets you browse and edit connections between events and exposed fields in your scene.

For each route, the Routing Map displays:

- An icon  and a label indicating the PROTO-context (namespace) of the route, if available.
- Icons and labels indicating the source and target nodes. Some icons may be shared between two or more routes.

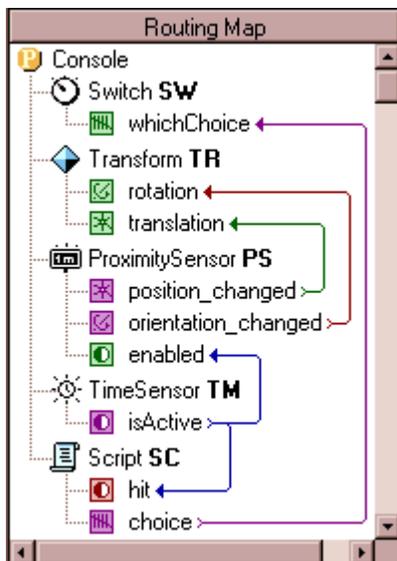
By default, Routing Map displays only node DEF-names. To show also typenamees of nodes, select **Show node typenamees** on the **Tree** tab in the **Options** dialog box.

- Icons and labels indicating the source and target fields.

To show also icons indicating actual field owners for **ISed** prototype interface fields, select **Show actual field owners** on the **Tree** tab in the **Options** dialog box.

- A colored line indicating the route itself.

To change a color corresponding to some route type, select the route type and desired color in the **Route colors** box on the **Tree** tab in the **Options** dialog box.



Managing Routes

► To set the caret position in the Text editor at a route:

- Double-click the route line. To move to the route source or target node or field declaration, double-click the appropriate icon or label.
- Choose **Go To** from the pull-down menu (right-click the route line to see the menu). To move to the route source or target node (or scripts field declaration), choose **Go To Source** or **Go To Target**.

▶ **To set the selection in the Routing Map to the route near the caret position:**

1. Activate Text editor window.
2. Choose **Synchronize** from the **View** menu (or type Alt+S).

If the caret position is near a node, field definition or event declaration, the **Synchronize** command enumerates all routes connecting this node or field with some other fields.

▶ **To select next or previous route:**

- Press Down or Up. Left and Right buttons scrolls Routing Map to show both ends of a long selected route. Typing Ctrl+Page Down and Ctrl+Page Up scrolls Routing Map one page down or up without changing the selection.

▶ **To remove a route:**

1. Select the route you want to remove.
2. Press Del or right-click over the route line and choose **Delete Route**.

▶ **To redirect a route:**

1. Select the route you want to change.
2. Right-click over the route line and choose **Add Route** (or type Ctrl+R).
3. Select a new source or/and target node or event name and click OK.

▶ **To add a new route:**

1. Activate Text editor or Scene Tree window.
2. Select in Scene Tree or move the caret at the source node, scripts eventOut or exposedField.
3. Choose **Start Route** from the **Edit** menu (or type Ctrl+E) to mark the node or the field as a source for the new route.
4. Select in Scene Tree or move the caret at the target node, scripts eventIn or exposedField.
5. Choose **Add Route** from the **Edit** menu (or type Ctrl+R) to open **Add Route** dialog box.
6. Enter, if necessary, unique names for the source or/and target nodes and fill out missing field names.
7. Check **Insert the route at the caret position** if you want to place newly created route near the caret position in Text editor. Otherwise, the new route will be inserted at the end of the file or the prototype declaration.
8. Click **OK**.

Resource View

Overview: Resource View

The Resource View displays dependencies (resources) of your scene and lets you browse and edit links to the resources.

For each resource, the Resource View displays:

- An icon indicating the resource type (for ImageTexture , for AudioClips , for custom resources , and so on).

To determine type of the resource, the Resource View finds the final declaration (unwinding all IS statements) of the field or exposed field of type MFString, containing the URL to the resource as a first string. The type of standard node or PROTO, that contain this declaration, becoming a type of the resource.

Remote resources (beginning with 'file:' protocol or with no explicit protocol) are indicated with the link mark  above the resource icon.

Missing local resources are indicated with the cross mark  above the resource icon.

- The URL of the resource.
- The name of the resource type (ImageTexture for instance).
- The reference count to the resource (number of links to this resource).
- The size of the local resource.

Resources			
URL	Type	Ref.	Size
 chasemusic3.wav	AudioClip	1	66KB
 vespa_all.jpg	ImageTexture	4	10KB
 lotus_all.jpg	ImageTexture	4	10KB
 jo_all.jpg	ImageTexture	1	14KB
 Os_All.jpg	ImageTexture	5	12KB
 ../Robot.wrl	Avatar	1	31KB
 Movie2a.wrl	Inline	1	164KB

Managing Resources

You can easily enumerate all links (URLs) to the resource.

► To set the caret position in the Text editor at a resource URL:

- Double-click the item. To move to the next link, double-click the item again.
- Choose **Select** from the pull-down menu (right-click the item to see the menu).

The list of items in the Resource View may be sorted by URL, by type, by reference count or by file size.

► To sort list of resources:

- Right-click an empty space or the header of the Resource View. The **Sort by** entries in the pull-down menu let you specify the sort order.

- Or click on the appropriate portion of the list box header. For example, click on the word **Size** in the header to sort by size. To reverse the sort order, click it again.

Using the Resource View, you can change all links to some resource to the another resource.

▶ **To redirect links to a resource:**

1. Select the resource you want to relink.
2. Press F2 or right-click over the resource and choose **Rename**.
3. Type a name in the text box and press Enter or press Esc if you want to leave the text box unchanged.

▶ **To browse for a local resource:**

1. Right-click over the resource and choose **Browse**.
2. Select a resource file you want to link to.
3. Check **Store absolute path** if you want to store absolute URL to the resource. Otherwise, relative path to the resource will be inserted, if possible.
4. Click **OK**.

▶ **To convert absolute path to a local resource to relative and vice versa:**

1. Right-click over the resource and choose **Browse**.
2. Check or uncheck **Store absolute path**.
3. Click **OK**.

If the resource has file associations, you can open it for previewing or editing using the Resource View.

▶ **To refresh list of resources after external files modification:**

- Right-click an empty space or the header of the Resource View. Choose **Refresh** from the pull-down menu.

▶ **To open a resource using associated application:**

- Right-click over the resource and choose **Open** or **Edit**.

Using Drag-and-Drop in the Resource View

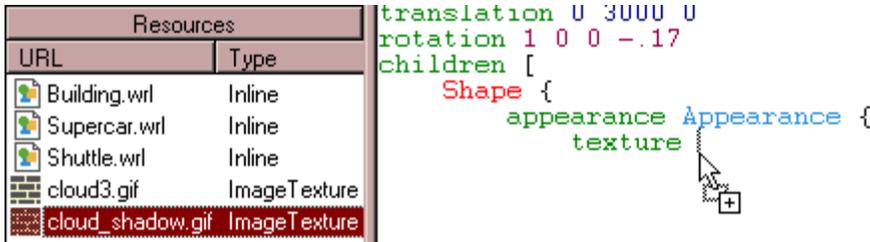
You can easily create a new link to resource, using drag-and-drop in the Resource View. You can also move or copy a resource file to an another application (Windows Explorer, for instance).

▶ **To create a new link to a resource:**

1. Select the resource you want to link to.
2. Drag the selected resource to the location in the Text editor.

When the pointer is over a place where the resource link can legally be inserted, dimmed caret appear; if you release the mouse button at that spot, a new link to the resource is inserted there.

Depending on the place you dropping the resource, only URL string may be inserted, or also a node of the resource type with the url field defined.



► **To copy resource file to an another application:**

1. Select the resource you want to copy.
2. Drag the selected resource to the application.

If you use your right mouse button to drag, a menu appears with the available options.

► **To open VRML resource in the VrmIPad:**

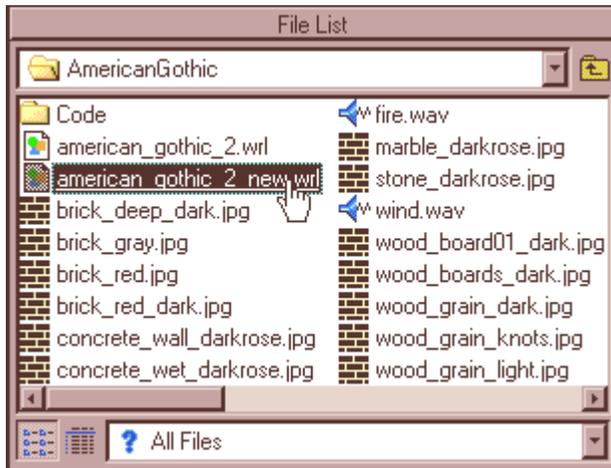
1. Select the VRML resource you want to open.
2. Drag the selected resource to the title bar of the VrmIPad.

If you drop a remote resource, VrmIPad automatically downloads it from the Net.

File List

Overview: File List

The File List displays drives, folders, and files accessible from your PC. The right-click menu contains file handling commands and you can open Windows Explorer directly to create folders, run searches, etc.



► To open VRML file in the VrmIPad:

1. Navigate to the drive and directory you want to access.
2. Double-click on a folder to display its files.
3. Double-click on a file to open it in the editor.

Tip: To have the File List use single-click to open a file, select **Single click to open a file (point to select)** on the **File List** tab in the **Options** dialog box.

Tip: To open a file in a new instance of VrmIPad, hold Shift key while opening the file.

You can set the file list to show specific file types. The filter is a global setting for all directories.

► To filter the file list:

1. Open combo box at the bottom of the File List.
2. Select a file type from the list.

The file list refreshes to apply the filter.

Using Drag-and-Drop in the File List

Drag-and-drop in the File List works like in Windows Explorer - you can drag files from and to the File List using left and right mouse buttons and CTRL or ALT keyboard modifiers.

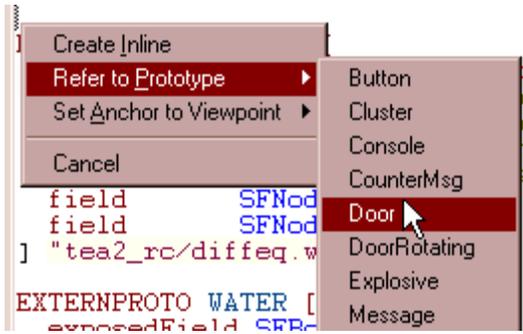
In addition, you can insert links to files in the Text editor, using drag-and-drop like in the **Resource View**.

► To create a link to a file in the Text editor:

1. Select the file you want to link to.
2. Drag the selected file to the location in the Text editor.

When the pointer is over a place where the resource link can legally be inserted, dimmed caret appear; if you release the mouse button at that spot, a new link to the resource is inserted there.

Depending on the place you dropping the resource, only URL string may be inserted, or also a node of the resource type with the url field defined. In the case of VRML files, a pull-down menu appears; you can choose between an Inline, Anchor or EXTERNPROTO declaration.



Publishing Wizard

The Basics of Publishing

Publishing refers to the process of locating all the files necessary to create a document, scene, or world and organizing the files for publication on a server. The Publishing Wizard is also able to put the files on your Web server, using Microsoft Web Publishing engine, or send it by e-mail, using MAPI transports.

Publishing is a one-way process. Publishing leaves your original files alone, but as it makes copies of those files it changes and optimizes those copies. Any time you want to change your world, reopen the original source files for the world, then make changes and republish. Don't modify the published copies directly.

Setting Up the Destination

The first step of the publishing process is to choose a destination for your published files.

If you have already published your scene, choose an existing destination directory. Later in the publishing process, the Publishing Wizard may remove the contents of the directory before creating the new version of the published world.

If you haven't previously published your scene, enter a name for a new destination directory (which the Publishing Wizard then creates for you), or click the Browse button and choose a directory from the file browser.

If you have selected **Publish files to the net** option for the first time, get ready to download and install the latest version of Microsoft Web Publishing Wizard. It automates the process of copying files from a temporary destination directory to the Web or FTP server.

Click **Next** when you're done.

If you change your mind, you can come back to this dialog by clicking a **Back** button. At any point in the publishing process you can click **Cancel** to stop the process without publishing.

If you already published some scenes, you can click **Finish** to complete the process now using recently used settings for the rest of the dialogs.

Choosing Additional Resource Directories

The next step of the publishing process is to choose the additional resource directories. Some scenes may contain invalid links to the resources, that that are possibly kept somewhere. To help the Wizard find these resources, you can specify a list of directories. Starting from these directories, the Publishing Wizard will search missing resources.

► To add a directory to the list:

1. Click the **Add** button.
2. Choose a directory from the file browser.
3. Click **OK**.

Click **Next** when you're done, or **Back** to return to the previous Wizard step.

Excluding Files from Publishing

This step lets you list files that you don't want the Publishing Wizard to include. You might want to exclude files that have been previously published, files that have been manually checked, files you know will already be on the Web server, or test files that you don't want to include in the final world.

Exclude filenames from the publish by selecting them in the list, then click **Exclude**. Revert this operation by selecting filenames in the list, then click **Include**. Excluded filename are marked by a cross mark .

URLs in your document which refer to these files are left unchanged in your documents.

Click **Next** when you're done, or **Back** to return to the previous Wizard step.

VRML Optimization Options

This step of the publishing process lets you choose options to optimize your world.

- **Pack VRML files using maximum compression**

Compressed files download faster and use less disk space, so use of this feature is recommended. Most browsers automatically decompress files that have been compressed. To disable this feature, click the box to remove the check.

- **Remove extra formatting from VRML files**

Check the box if you want the Publishing Wizard to reformat the VRML files by unindenting lines, removing unnecessary whitespaces and empty lines. Reformatted files use less disk space, but usually lost in human readability.

- **Remove comments**

Selecting this option removes all comments from a file during the reformat.

- **Remove default field values**

Selecting this option removes field values which are explicitly declared with a value identical to the default field value.

- **Simplify floating point numbers**

Check the box if you want the Publishing Wizard when possible to reduce float numbers to integers without lost of quality, remove trailing zeros and remove unnecessary zero values to the left of decimal point. For example, number 10.0 reduce to 10, 0.10 – to .1 and so on.

- **Adjust numeric resolution**

This option also removes unnecessary text, but in some cases such removal is lossy, and in other cases there may be external code, such as an EAI application, relying on the text. So you should be considered carefully before use the option.

This option adjusts numeric resolution of floating point numbers. In many cases, exporters output these numbers with a much greater resolution than necessary. The sub-options allow resolution to be indicated by type of the number. For example, coordinate resolution relates to all world coordinates and sizes: the **point** field in the **Coordinate** node, the **size** field in the **Box** node and so on. Orientation resolution relates to all SFRotation and MFRotation fields and also to the fields that defines angles.

Click **Next** when you're done, or **Back** to return to the previous Wizard step.

Specifying Directory Organization

In this step, you decide whether you want to use a flat directory structure, or whether you want to use a hierarchical structure.

- **Put all source files in one directory**

This is the best choice if your world uses only a few files.

- **Arrange source files in subfolders, depending of its type**

Use the resource type based directory structure. This is the best choice if you have a complex multi-file world.

- **Retain the same directory structure used by the source files**

Use the same directory structure that your source files use. This is the best choice if you have a complex multi-file world stored in a set of nested directories.

- **In-line resources as base64 encoded data**

Include resources of specified types into VRML files body, excluding it from the list of publishing files. VRML files with in-lined resources are currently supported only by Cortona VRML viewer.

Click **Next** when you're done, or **Back** to return to the previous Wizard step.

Reviewing the Directory Structure

In this step, you review the directory structure. This dialog lists every file that will be placed in the destination directory. If the files don't have the names and locations you expect, click **Back** to go back to the mappings dialog.

Click **Next** when you're done, or **Back** to return to the previous Wizard step. Your files are set up for final publishing when you click **Next**, so you can't turn back to previous steps from this point.

Previewing Published Documents

The final step of the Publishing lets you preview the published world immediately. Click **Preview** to preview the world in your default browser. Click **Open Folder** to open the destination directory with your files, using Windows Explorer window.

If you have chosen publishing to the net or through e-mail in the first step, click **Next** to launch Microsoft Web Publishing Wizard or mail sender. Otherwise, click **Close** to finish publishing.

Hints and Tips

How to compress a VRML file

Compressed files download faster and use less disk space, so use of this feature is recommended. Most browsers automatically decompress files that have been compressed.

▶ **To compress an existing VRML file:**

1. From the **File** menu, choose **Open**.
2. Select your file.
3. Click **OK**.
4. Change the compression level, if you like, as written below.
5. From the **File** menu, choose **Save As**.
6. Select **Save compressed** checkbox.
7. Enter or select a destination file name.
8. Click **OK**.

▶ **To change default compression level:**

1. From the Tools menu, choose Options.
2. Select the **General** tab.
3. Use the **Default compression** drop-down list to select the compression level:
 - Maximum – best compression, but lowest speed of compression;
 - Normal – balance between speed and compression quality;
 - Fast – fastest speed, but poor compression quality.
4. Click **OK**.

▶ **To compress a file without using the Save As dialog box:**

1. Double-click the pane **GZ** in the status bar at the bottom of VrmIPad window.
2. From the **File** menu, choose **Save**, or type Ctrl+S.

How to download a VRML file from the Net.

▶ **If you are using download dialog for the first time:**

1. From the Tools menu, choose Options.
2. Select the **General** tab.
3. Enter or browse destination directory in the **Location for downloaded files** box.

This directory will keep the downloaded files and its resources – images, audio files, scripts, etc.
4. Specify additional download options.
5. Click **OK**.

▶ **To download a file:**

1. From the **File** menu, choose **Download**, or double-click the pane **WEB** in the status bar at the bottom of VrmIPad window.
2. Type valid Internet address (URL) of your file to the **Open** box, or use the drop-down list to select previous URLs. When you select an URL in this list, the URL is added in the **Open** box.
3. Specify additional download options.
4. Click **OK**.

▶ **To download a file without using the Download dialog box:**

- Drag-and-drop an Internet shortcut from your Web-browser to the caption of VrmIPad.

▶ **To cancel downloading:**

- Double-click the animated globe in the status bar at the bottom of VrmIPad window.

How to upload a VRML file to a remote server.

First of all you should download and install the latest version of Microsoft Web Publishing Wizard from <http://www.microsoft.com/windows/software/webpost/>. It automates the process of copying files from your PC to the Web or FTP server.

▶ **To upload a VRML file using Microsoft Web Publishing Wizard:**

1. Open the file into VrmIPad.
2. From the **File** menu, choose **Publish**.
3. Select **Publish files to the net** options in the first page.
4. Proceed through the rest of the Publishing Wizard.
5. After the last page, follow instructions of the Microsoft Web Publishing Wizard. For the first time, specify path to your FTP server, user name and login.

How to quickly find the definition (reference) of an identifier.

▶ **To jump to the definition of an identifier (PROTO, node, or field name):**

1. Position the caret at the identifier in the Text editor, or select an identifier in the Scene Tree.
2. Select **Go To Definition** from the pull-down menu or type Ctrl+F11.

▶ **To jump to the reference of an identifier (PROTO, node, or field name):**

1. Position the caret at the identifier in the Text editor, or select an identifier in the Scene Tree.
2. Select **Go To Reference** from the pull-down menu or type F11.
3. Select **Next Reference** from the pull-down menu or type F11 to jump to the next reference of the identifier.

How to insert a pair of node or PROTO braces.

▶ **To insert a pair of braces after node or PROTO name:**

1. Position the caret to the space after node or PROTO name.
2. Type Ctrl+Space.

How to insert default field value.

▶ **To insert a default value after field declaration or definition:**

1. Position the caret to the space after the field declaration or definition or inside multiple field value.
2. Type Ctrl+Space.

How to indent a block of lines.

▶ **To indent a block of lines:**

- Select the lines and then press Tab.

▶ **To unindent a block of lines:**

- Select the lines and press Shift+Tab.

▶ **To format a block of lines using the smart indent settings:**

- Select the lines and press Ctrl+Shift+F.

How to comment a block of lines or an entire node.

▶ **To comment a block of lines:**

- Select the lines and press Ctrl+Shift+C.

▶ **To uncomment a block of lines:**

- Select the commented lines and press Ctrl+Shift+C.

▶ **To comment a node, PROTO declaration or field:**

- Select the identifier of a node, PROTO declaration or field in the Scene Tree and choose the **Comment** from the pull-down menu or press Ctrl+Shift+C.

How to locate a syntax or semantic error.

▶ **To jump to the first error:**

- Double-click on the highlighted error indicator in the status bar at the bottom of the VrmIPad window.

▶ **To jump to the next error:**

- From the **View** menu, choose **Next Error** or type F4.

How to make a VRML file smaller.

Reducing file size and improving reliability and performance go hand in hand. VrmIPad can automatically remove code that is unnecessary, redundant or unusable. The size reduction is often dramatic, and the resulting file is more likely to run consistently across browsers.

▶ **To reduce file size using the Unused Identifiers dialog box:**

1. From the **View** menu, choose **Unused Identifiers**.

2. In the list below, select identifiers you want to remove.
3. Click **Delete** or type Del.

Note: You can safely remove default field values. Remove identifiers of the other types with caution – some node names or field declarations may be used by outer scripts, EAI or ActiveX components. Also, some VRML authoring tools may use unnecessary for rendering node names and field declarations for holding additional user interface information, such as object names and properties.

► **To reduce file size using the Publishing Wizard:**

1. From the **File** menu, choose **Publish**.
2. In the first page, choose copying document to a folder.
3. Proceed through the following pages up to the **Specify VRML preferences** page.
4. Select the options you want to.
5. Proceed through the rest of the Wizard.

How to expose a field to a PROTO interface.

► **To automatically expose a field:**

1. Position the caret to the space after the field definition or **Script** node's field declaration (without a field value).
2. Type the **IS** keyword into your code and press SPACE.
3. Type Ctrl+Space and select one of the choices.

There are four possible choices in the case of the exposedField:

- Expose the field as an exposeField declaration.
- Expose the field as a field declaration.
- Expose the field as an eventIn declaration.
- Expose the field as an eventOut declaration.

```
PROTO Timer [  
]{  
    TimeSensor {  
        stopTime IS |  
        set_stopTime  
        stopTime  
        stopTime  
        stopTime changed
```

4. Press SPACE or ENTER.

```
PROTO Timer [  
    exposedField SFTIME stopTime 0  
]{  
    TimeSensor {  
        stopTime IS stopTime
```

How to register a VRML extension.

VRML extension is a custom node type in addition to a set of the standard VRML97 nodes.

► **To register a VRML extension:**

1. Create a VRML file (in VrmIPad, for instance), containing PROTO declarations with empty bodies. These PROTOs should describe your VRML extension – names of the additional node types, field names, categories and types.

Note: Actually, a set of the standard VRML97 nodes implemented internally in VrmIPad as a hidden VRML file. To open it in the editor, type Ctrl+Shift+N.

2. Save the file somewhere.
3. From the **Tools** menu, choose **Options**.
4. Select the **General** tab.
5. Select **Enable VRML extensions** checkbox.
6. Type in the **Source** field path to the saved file or browse it using a button at the right of the field.
7. Click **OK**.
8. Restart VrmIPad.

How to change a set of the standard nodes.

► **To change a standard set of node types, using VrmIPad:**

1. Press Ctrl+Shift+N to load the standard nodes as PROTO declarations.
2. Make all changes you want to.
3. Save the file somewhere.
4. From the **Tools** menu, choose **Options**.
5. Select the **General** tab.
6. Select **Enable VRML extensions** checkbox.
7. Type in the **Source** field path to the saved file or browse it using a button at the right of the field.
8. Select **Ignore all standard node types** checkbox.
9. Click **OK**.
10. Restart VrmIPad.

How to change a source of the Node Help.

Help for the standard VRML97 nodes is available through the **Help for Current Node** from the **Help** menu. By default, VrmIPad uses remote HTML manual <http://www.vrml.org/technicalinfo/specifications/vrml97/part1/nodesRef.html>, but you can download the manual into your PC and redirect VrmIPad on it.

► **To change a Node Help source:**

1. From the Windows **Start** menu, choose **Run**.
2. In the **Open** box, type 'regedit', and click **OK**.
3. Open registry key HKEY_CURRENT_USER/Software/ParallelGraphics/VrmIPad/Settings.
4. Create a new string value, named 'NodesRef'.
5. Specify in this value full path to your downloaded manual, for example – 'C:\VRML Spec\part1\nodesRef.html'.

What are the ways I can access the VrmlPad object model?

You can access the VrmlPad object model in three ways:

- Write VBScript or JavaScript macros that script VrmlPad through its object model.
- Create add-ins.
- In ActiveX-enabled applications, write an Automation controller that accesses VrmlPad by creating an instance of it. For example, in Visual Basic, use the **CreateObject** function to create an instance of VrmlPad. Or in C++, call the Win32 **CoCreateInstance** function. For more information, see the application's documentation.

How to automate VrmlPad from another application.

With Automation, you can control VrmlPad from other applications such as Word, Excel, or Visual Basic. You can start VrmlPad, automate VrmlPad tasks, and then stop VrmlPad when the tasks are finished.

To start VrmlPad from another application, create an instance of the VrmlPad **Document** object. For example, to start VrmlPad from Visual Basic or Visual Basic for Applications, use the following code:

```
Dim doc As Object
Set doc = CreateObject("VrmlPad.Document")
```

In Visual Basic you can use the VrmlPad type library to access the VrmlPad object model. To do this, from the **Project** menu choose **References** and select the *VrmlPad Application Library* checkbox.

```
Dim doc As VrmlPad.Document
Set doc = New VrmlPad.Document
```

When VrmlPad starts, it will be invisible — unless you make it visible by using the **Visible** property of the **Window** object:

```
Dim doc As VrmlPad.Document
Set doc = New VrmlPad.Document
doc.Window.Visible = True
```

After starting VrmlPad, you can automate tasks in it by using the appropriate code. For example, in the following code, a *WorldInfo* node is added to the created document:

```
Dim doc As VrmlPad.Document
Set doc = New VrmlPad.Document
doc.Window.Visible = True
doc.RootNodes.Add("WorldInfo")("info") = "Created by me"
```

After completing your VrmlPad tasks, you can stop VrmlPad releasing all references to the **Document** object. If you use Visual Basic to automate VrmlPad, Visual Basic automatically releases all references to the **Document** object.

Note: If the user tries to stop VrmlPad while your program is controlling it, VrmlPad will remain in memory but the user interface will become invisible. If this happens, release all references to the **Document** object so that VrmlPad can quit.