Virtual Reality Toolbox Release Notes

The "Virtual Reality Toolbox 4.0 Release Notes" on page 1-1 describe the changes introduced in the latest version of Virtual Reality Toolbox. The following topics are discussed in these Release Notes.

The Virtual Reality Toolbox Release Notes also provide information about recent versions of the product, in case you are upgrading from a version that was released prior to Release 14.

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

1 Virtual Reality Toolbox 4.0 Release Notes

2 Virtual Reality Toolbox 3.0 Release Notes

3 Virtual Reality Toolbox 2.0 Release Notes

1

Virtual Reality Toolbox 4.0 Release Notes

New Features

This section summarizes the new features added to the Virtual Reality Toolbox for Release 14.

Improved Overall Performance

Infrastructure changes have improved the overall performance of Virtual Reality Toolbox.

Tracing and Offline Animation Files

Virtual Reality Toolbox enables you to record animations of virtual scenes that are controlled by Simulink or MATLAB. You can then later play back these animations offline (in other words, independent of MATLAB, Simulink, or Virtual Reality Toolbox). You can save animation data in the following formats:

- 3-D VRML — Virtual Reality Toolbox traces object movements and saves that data into a VRML file using VRML97 timers and position/orientation interpolators. During the simulation, Virtual Reality Toolbox creates interpolators and saves key and interpolation data values for all the controlled movable objects in the scene. After the simulation is stopped, Virtual Reality Toolbox saves the entire original VRML scene with the added timers/interpolators into a user-specified VRML file.

  You can then view these files with the Virtual Reality Toolbox viewer. 3-D VRML files typically use much less disk space than Audio Video Interleave (AVI) files. If you make any navigation movements in the Virtual Reality Toolbox viewer while recording the animation, Virtual Reality Toolbox does not save any of these movements.
- 2-D AVI — Virtual Reality Toolbox traces object movements and writes animation data into an Audio Video Interleave (AVI) file using the Virtual Reality Toolbox `vrfigure/capture` method and the MATLAB `avifile` function. Because avifile can record the exact 2-D snapshot of each figure window, the recorded 2-D animation reflects exactly what you see in the viewer window. It includes any navigation movements you make during the recording.

This functionality has been integrated into the Virtual Reality Toolbox Viewer. See Viewing Virtual Worlds of the Virtual Reality Toolbox User's Guide documentation for complete details.

You can also save animation data through the MATLAB interface. This functionality has been integrated into the `vrworld` and `vrfigure` methods. See MATLAB Interface of the Virtual Reality Toolbox User's Guide documentation for procedures on how to save animation data. For updates to the `vrworld` and `vrfigure` methods, see "vrworld Object Property Updates" and "vrfigure Object Method Updates" respectively.

Notes when working with animation data:

- If you distribute the resulting VRML animation file, be sure to also distribute all the inlined object and texture files referenced in the original VRML world file.
- While recording 2-D AVI animation data, always ensure that the Virtual Reality Toolbox viewer is the topmost window and fully visible. Graphics acceleration limitations might prevent the proper recording of 2-D animation otherwise.
- For the creation of animation files, timing is an important issue. When you create animation files that are controlled from Simulink, the timing properties that control when the recording is started and stopped, and when each data snapshot (3-D) or animation frame (2-D) is saved, are related to the Simulink time. To better capture the scene dynamics, you might want to change the **Sample time** parameter of the VR Sink block parameter.

  When you create animation files that are controlled from MATLAB, you need to programmatically advance the time in the virtual scene. See "Improved Timing Control" for further details.

Improved Virtual Reality Toolbox Viewer

The Virtual Reality Toolbox Viewer has been completely updated and improved. In addition to the original Navigation Panel at the bottom of the window, the Virtual Reality Toolbox Viewer now has a menu bar and associated toolbar. These additional areas allow you to perform a number of operations on the loaded virtual world, including

- Start the V-Realm Builder editor
- Perform navigation and rendering operations
- Work with, add, and remove viewpoints

- Start and stop model simulation
- Edit VR block parameters
- Record animations of the model simulation. You can later play back the animation files without Virtual Reality Toolbox or other MATLAB product.

In addition, Virtual Reality Toolbox has been enhanced for better VRML 97 compatibility, including:

- Proper handling of PROTO nodes
- Improved VRMLScript processing
- Improved texture rendering, allowing for resolution to the hardware limit of your video card
- Improved rendering of transparent images

In addition, the Virtual Reality Toolbox Viewer keyboard navigation has changed. See Viewing Virtual Worlds of the Virtual Reality Toolbox User's Guide documentation for further details.

Double-Clicking VR Sink Blocks Behavior Changed

The behavior of a VR Sink block when you double-click it has changed as follows:

1. The first time you add a VR Sink block to a model, double-clicking that block displays the block parameter dialog for that block.

2. After configuration is complete and you make an association with a virtual world, double-clicking the VR Sink block displays the Virtual Reality Toolbox Viewer for that model.

   To display the block parameter dialog for the VR Sink block in the Virtual Reality Toolbox viewer, select the **Simulation** menu, then **Block Parameters**.

Extended Support of Data Types

Virtual Reality Toolbox now supports a number of MATLAB data types. In earlier releases, Virtual Reality Toolbox only supported a data type of `double`. Virtual Reality Toolbox provides an interface between the MATLAB and

Simulink environment and VRML scenes. With this interface, you can set and get the VRML scene node field values. The `setfield` and `getfield` vrnode methods have been updated to work with these new data types. See VRML Data Types in the Virtual Reality Toolbox User's Guide documentation for further details about data types.

Improved Timing Control

The time in virtual scenes advances independently of MATLAB and Simulink interfaces. As a result, if you define dynamic actions directly in the VRML world, these actions cannot interact appropriately with object actions controlled by Virtual Reality Toolbox through MATLAB or Simulink. To compensate for this, Virtual Reality Toolbox now provides two `vrworld` properties, `Time` and `TimeSource`. The `Time` property contains the current time in the virtual world. The `TimeSource` property defines the source of the time for the virtual world. The `TimeSource` property can take the following values:

- `'external'` — This is the default value. This property value specifies that an external source controls the time in a virtual world (in other words, all Virtual Reality Toolbox viewer windows). This external source can be either from Simulink, using the simulation time, or from MATLAB, using the `vrworld` method `set(w,'Time',time_value)`. The `'external'` property value eliminates the unpredictable interferences between Simulink and internal virtual world time and ambiguities in interpreting the time in the virtual world.
- `'freerun'` — This property value specifies that the time in virtual worlds advances independently based on the system timer. You can still set the time from which the virtual world clock starts advancing with the `vrworld` method `set(w,'Time',time_value)`.

HP-UX Platform Support

The Virtual Reality Toolbox is now supported on Hewlett-Packard UNIX (HP-UX) 11.00.

New vrfigure Object Functions

The `vrfigure` object has two new methods, `vrgcf` and `vrgcbf`.

- `vrgcf` — Gets the handle for a currently active virtual reality figure. It is most useful to query and set virtual reality figure properties.
- `vrgcbf` — Gets the current callback virtual reality figure.

Joystick Input and Magellan Space Mouse Block Updates

When you place the Joystick Input and Magellan Space Mouse blocks in disabled subsystems, these blocks no longer require that actual hardware be connected to the computer for the model to run. This allows you to create models with alternative user interfaces, switchable according to the actual hardware configuration.

New and Updated Demos

New and updated demos in Release 14 are listed in the table below. You can open the demos by entering the corresponding model names in the MATLAB Command Window.

Title	Model Name
Portal Crane with Joystick Control	`vrcrane_joystick`
Portal Crane with Predefined Trajectory	`vrcrane_traj`
Heat Transfer Visualization with 2-D Animation	`vrheat_anim`
Vehicle Dynamics Visualization	`vr_octavia`

Virtual Reality Toolbox Simulink Blocks Updates

The Simulink block library for the Virtual Reality Toolbox has the following block updates:

- VR Sink blocks now accepts all meaningful data types as input. It converts these data types to natural VRML types as necessary. These data types include logical values, many types of signed and unsigned integers, singles, and doubles. See VRML Data Types in the Virtual Reality Toolbox User's Guide documentation for further details about data types.

- VR Source blocks now output signals of data types corresponding to the natural VRML data types of the associated fields. These data types include logical values, many types of signed and unsigned integers, singles, and doubles. See VRML Data Types in the Virtual Reality Toolbox User's Guide documentation for further details about data types.
- Space Mouse Input blocks now support USB devices (such as the SpaceBall input device), and the SpaceTraveler motion controller.
- Joystick Input blocks now support force-feedback devices such as force-feedback joysticks, steering wheels, and haptic devices. To use this functionality, you must install Microsoft DirectX Version 8.0 or higher.

VR Sink and VR Source Block Source File

The behavior of the VR Sink and VR Source block Source file text field has changed as follows:

- Browsing to a file enters that filename in the text field, with a path relative to the Simulink model location. For example, if the VRML file resides in `<Simulink model location>\vrml\vrbounce.wrl`, the filename that appears in the text field is `vrml\vrbounce.wrl`.
- If a model has a VRML file associated with it, Virtual Reality Toolbox searches the path associated with that file. If Virtual Reality Toolbox does not find the file on this path, it then searches for the filename using the MATLAB path value.
- If you enter an absolute and fully qualified path for the VRML file, Virtual Reality Toolbox searches just that path for the file.

MATLAB Interface Updates

Associated with the support of extended data types (see "Extended Support of Data Types"), the MATLAB interface `vrsetpref` and `vrgetpref` functions now set and get the following new preferences in addition to earlier preferences:

Preferences Changes

Virtual Reality Toolbox lets you specify preference settings so that you can define the default behavior of Virtual Reality Toolbox. The preferences functionality has the following changes:

- The ability to set Virtual Reality Toolbox preferences has moved from the VR Sink/Source blocks to the standard MATLAB preference system accessible through MATLAB main window **File -> Preferences** menu item.

  Note that for this release, you can set only the preferences for **HttpPort**, **VrPort**, **Editor**, and **TransportBuffer** preferences using the MATLAB **Preferences** menu. To get or set the full set of preferences, use the functions `vrgetpref` and `vrsetpref`.
- The following preferences have been added to the Virtual Reality Toolbox. For preferences that begin with the string `DefaultFigure` or `DefaultWorld`, these values are the default values for the corresponding `vrfigure` or `vrworld` property.

Preference	Description
`DataTypeBool`	Specifies the handling of the VRML `Bool` data type for `vrnode/setfield` and `vrnode/getfield`. Valid values are `'logical'` and `'char'`. If set to `'logical'`, the VRML `Bool` data type is returned as a logical value. If set to `'char'`, the `Bool` data type is returned `'on'` or `'off'`. Default is `'logical'`.
`DataTypeInt32`	Specifies handling of the VRML `Int32` data type for `vrnode/setfield` and `vrnode/getfield`. Valid values are `'int32'` and `'double'`. If set to `'int32'`, the VRML `Int32` data type is returned as `int32`. If set to `'double'`, the `Int32` is returned as `'double'`. Default is `'double'`.
`DataTypeFloat`	Specifies the handling of the VRML `float` data type for `vrnode/setfield` and `vrnode/getfield`. Valid values are `'single'` and `'double'`. If set to `'single'`, the VRML `Float` and `Color` data types are returned as `'single'`. If set to `'double'`, the `Float` and `Color` data types are returned as `'double'`. Default is `'double'`.

Preference	Description
`DefaultFigureAnti Aliasing`	Determines whether antialiasing is used by default for new `vrfigure` objects. Valid values are `'off'` and `'on'`.
`DefaultFigureDeleteFcn`	Specifies the default callback invoked when closing a `vrfigure` object.
`DefaultFigureLighting`	Specifies whether the lights are rendered by default for new `vrfigure` objects. Valid values are `'off'` and `'on'`.
`DefaultFigureMax TextureSize`	Specifies the default maximum pixel size of a texture used in rendering new `vrfigure` objects. Valid values are `'auto'` and `32 <= x <=` video card limit, where `x` is a power of 2.
`DefaultFigureRecord2D CompressMethod`	Specifies the default compression method for creating 2-D animation files for new `vrfigure` objects. Valid values are `''`, `'auto'`, `'lossless'`, and `'codec_code'`.
`DefaultFigureRecord2D CompressQuality`	Specifies the default quality of 2-D animation file compression for new `vrfigure` objects. Valid values are 0–100.
`DefaultFigureRecord2D FileName`	Specifies the default 2-D offline animation file name for new `vrfigure` objects.
`DefaultFigureStatusBar`	Specifies whether the status bar appears by default at the bottom of the Virtual Reality Toolbox viewer for new `vrfigure` objects. Valid values are `'off'` and `'on'`.
`DefaultFigure Transparency`	Specifies whether or not transparency information is taken into account when rendering for new `vrfigure` objects. Valid values are `'off'` and `'on'`.

Preference	Description
`DefaultFigureWireframe`	Specifies whether objects are drawn as solids or wireframes by default for new `vrfigure` objects. Valid values are `'off'` and `'on'`.
`DefaultWorldRecord3D FileName`	Specifies the default 3-D animation file name for new `vrworld` objects.
`DefaultWorldRecordMode`	Specifies the default animation recording mode for new `vrworld` objects. Valid values are `'manual'` and `'scheduled'`.
`DefaultWorldRecord Interval`	Specifies the default start and stop times for scheduled animation recording for new `vrworld` objects. Valid value is a vector of two doubles.
`DefaultWorldRemoteView`	Specifies whether the virtual world is enabled by default for remote viewing for new `vrworld` objects. Valid values are `'off'` and `'on'`.
`DefaultWorldTimeSource`	Specifies the default source of the time for new `vrworld` objects. Valid values are `'external'` and `'freerun'`.
`TransportTimeout`	Amount of time VR server waits for a reply from the client. If there is no response from the client, VR server disconnects from the client.

vrworld Object Property Updates

Associated with the support of animation files, the `vrworld` object now supports the following new properties:

Property	Description
`Record3D`	Enables 3D animation recording. Read/write.
`Record3DFileName`	3D animation file name. The string can contain tokens that are replaced by the corresponding information when the animation recording takes place. Read/write.
`Recording`	Animation recording toggle. This property acts as the master recording switch. Read/write.
`RecordMode`	Animation recording mode. Read/write.
`RecordInterval`	Start and stop times for scheduled animation recording. Corresponds to the virtual world object `Time` property. Read/write.
`Time`	Current time in the virtual world. Read/write.
`TimeSource`	The source of the time for the virtual world. If set to `'external'`, time in the scene is controlled from MATLAB (by setting the `Time` property) or Simulink (simulation time), if set to `'freerun'`, time in the scene advances independently based on the system timer. Read/write.

vrfigure Object Method Updates

The `vrfigure` object now supports the following new properties:

Property	Description
`MaxTextureSize`	Sets the maximum pixel size of a texture used in rendering `vrfigure` objects. The smaller the size, the faster the texture can render. Increasing this value improves image quality but decreases performance. A value of `'auto'` sets the maximum possible pixel size. If the value you enter is unsuitable, a warning might trigger. Virtual Reality Toolbox then automatically adjusts the property to the next smaller suitable value. Read/write.
`NavMode`	Specifies navigation mode. Read/write.
`NavSpeed`	Specifies navigation speed. Read/write.
`NavZones`	Toggles navigation zones on/off. Read/write.
Record2D	Enables 2-D offline animation file recording. Read/write.
Record2DCompressMethod	Specifies the compression method for creating 2-D animation files. The codec_code must be registered in the system. See the MATLAB function documentation for `avifile`. Read/write.
Record2DCompressQuality	Specifies the quality of 2-D animation file compression. Read/write.
Record2DFileName	Specifies the 2-D offline animation file name. The string can contain tokens that are replaced by the corresponding information when the animation recording takes place. Read/write.

Property	Description
StatusBar	Toggles the status bar at the bottom of the Virtual Reality Toolbox viewer. Read/write.
Toolbar	Toggles toolbar on the Virtual Reality Toolbox viewer. Read/write.

Virtual Reality Toolbox and Graphics Cards Drivers

For optimal performance, use Virtual Reality Toolbox on a computer with a graphics card that supports 3-D hardware acceleration and the OpenGL environment.

If you encounter issues with the Virtual Reality Toolbox rendering or the Virtual Reality Toolbox viewer display, ensure that you are using the latest version of your graphics card driver. You can obtain the latest version from the graphics card manufacturer.

To list general OpenGL information, such as the OpenGL version and vendor information, type

```
opengl info
```

If you are running Virtual Reality Toolbox on a Windows systems, and you are experiencing rendering problems, experiment with the graphics card hardware acceleration settings to reduce and turn off the hardware acceleration. There are a number of ways that you can access the hardware acceleration setting. One way is the following:

1. From the **Control Panel**, select **Display**.

   The **Display Properties** window is displayed.

2. Select the **Settings** tab.

3. Click the **Advanced** button.

   A configuration window for your graphics card is displayed.

4. Select the **Troubleshooting** tab.

5. Iteratively adjust the hardware acceleration setting lower and off. Try the results after each adjustment.

If lowering or disabling the hardware acceleration corrects the rendering issues, your graphics card does not properly support OpenGL. If you already have the latest version of the graphics card driver, you might need a new graphics card.

Known Limitations

This section describes known limitations for Virtual Reality Toolbox on Linux desktop or window management environments for Virtual Reality Toolbox Release 4.0:

- For some Linux window managers/desktop environments, the Virtual Reality Toolbox viewer does not save its position. For example, this limitation occurs in the following:
  - K Desktop Environment (KDE)
  - Debian GNU/Linux ICE Window Manager
  - F Virtual Window Manager (FVWM) Version 2.0 and higher
- Due to a limitation in the internal viewer, scenes that have large scaled worlds might not be displayed correctly.

  The workaround is to edit your virtual world through the V-Realm Builder or a text editor. Increase the `visibilityLimit` field of the `NavigationInfo` node to a number that is greater than the longest distance in the virtual world. For example

```
NavigationInfo
{
  ...
  visibilityLimit 1e7
  ...
}
```

- You might not be able to access an Virtual Reality Toolbox object method reference page from the MATLAB Command Window. To access a page, use the MATLAB Help window.

Major Bug Fixes

Virtual Reality Toolbox 4.0 includes several bug fixes made since Version 3.1. This section describes the particularly important Version 4.0 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

2

Virtual Reality Toolbox 3.0 Release Notes

New Features

This section summarizes the new features and enhancements introduced in the Virtual Reality Toolbox 3.0.

For an overview of the Virtual Reality Toolbox, see “Introduction to the Virtual Reality Toolbox” on page 3-2.

Cross-Platform Support

The Virtual Reality Toolbox 3.0 is supported on both PC and UNIX platforms. See Supported Computer Platforms in the Virtual Reality Toolbox documentation for more information.

Virtual Reality Toolbox Viewer

The Virtual Reality Toolbox 3.0 now contains a VRML viewer. You can use this viewer on all supported operating systems. With the Virtual Reality Toolbox viewer, you can move between predefined viewpoints in a virtual scene. You can also control the navigation method, speed, and rendering of the virtual world. From the MATLAB interface, you can control the Virtual Reality Toolbox viewer better than other VRML-enabled Web browsers.

Improved Performance

The communication between the Virtual Reality Toolbox and VRML-enabled Web browsers is improved. You are now able to perform such tasks as controlling multiple objects in a virtual scene or setting multiple field values faster than before.

Improved MATLAB Interface

It is now easier to access and manipulate virtual world objects from the MATLAB command line. You can use dot notation to change object properties. Also, multiple field values are transferred between MATLAB and the Virtual Reality server in a form consistent with their VRML representation.

Platform Limitations for HP and IBM

The Virtual Reality Toolbox is not supported on HP and IBM platforms.

Upgrading from an Earlier Release

This section describes an upgrade issue involved in moving from the Virtual Reality Toolbox 2.0 to Version 3.0.

Customized V-Realm Object Libraries

If you are on a PC platform and you created your own object libraries in V-Realm while using the Virtual Reality Toolbox 2.0, deleting MATLAB deletes these custom libraries. You need to save these libraries before uninstalling the older version of MATLAB:

1. Save the contents of the `<MATLAB root>\toolbox\vr\vrealm` directory to another location on your system.
2. Uninstall the older version of MATLAB.
3. Install MATLAB 6.5 (Release 13).
4. Install V-Realm using the command

   ```
   vrinstall -install editor
   ```

5. Save the files from your old `vrealm` directory into the new `vrealm` directory.

MATLAB 6.5 should have the same directory structure as your previous version of MATLAB. If the directory structure is not identical, search your system for `vrbuild2.ini`. Edit the relative paths within this file to reflect the directory structure of MATLAB 6.5.

After you have moved your files into the new `vrealm` directory and customized your `vrbuild2.ini` file, do not reinstall the V-Realm editor. The command

```
vrinstall -install editor
```

replaces the existing `vrbuild2.ini` file with the default template file.

Known Software and Documentation Problems

This section includes a link to a description of known software and documentation problems in Version 3.0.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

3

Virtual Reality Toolbox 2.0 Release Notes

Introduction to the Virtual Reality Toolbox

The Virtual Reality Toolbox extends the capabilities of MATLAB and Simulink into the world of virtual reality graphics. Using standard Virtual Reality Modeling Language (VRML) technology, you can create animated three-dimensional scenes that are driven from the MATLAB and Simulink environment.

Simulink Interface

You can observe a simulation of your dynamic system over time in a visually realistic three-dimensional model.

Most of the Virtual Reality Toolbox features can be implemented with Simulink blocks. Once you include these blocks in a Simulink diagram, they allow you to select the virtual world, which you connect to Simulink signals. The Virtual Reality Toolbox automatically scans the virtual world for available VRML nodes that can be driven by Simulink.

MATLAB Interface

The Virtual Reality Toolbox provides a flexible MATLAB interface to a virtual reality world.

Virtual Reality Modeling Language (VRML)

The Virtual Reality Modeling Language (VRML) is an ISO standard that is open, text-based, and uses a Web-oriented format. You use VRML to define a virtual world that you can display in a Web browser and connect to a Simulink model.

The Virtual Reality Toolbox uses many of the advanced features defined in the current VRML97 specification. The term VRML, in this chapter, always refers to VRML as defined in the VRML97 standard ISO/IEC 14772-1:1997. This format includes a description of three-dimensional scenes, sound, internal actions, and Web anchors.

VRML Viewing

If you install a VRML plug-in, you can view a virtual world in your preferred Web browser. For PC platforms, the Virtual Reality Toolbox includes the popular VRML plug-in, blaxxun Contact.

The Virtual Reality Toolbox connects MATLAB and Simulink with a VRML enabled browser to display a simulated process using the TCP/IP protocol. This allows you to watch a simulated virtual world not only on the computer where MATLAB and Simulink are running, but also on other computers connected through the Internet.

VRML Authoring

For PC platforms, the Virtual Reality Toolbox includes the leading VRML authoring tool, V-Realm Builder, by Ligos Corp. With the addition of this VRML authoring tool, the Virtual Reality Toolbox provides a complete authoring, development, and working environment for carrying out three-dimensional visual simulations.