



MAK RTI

Users Guide



VT MAK
A company of VT Systems



MAK RTI

Users Guide

Version 4.5

Copyright © 2018 VT MAK
All rights Reserved. Printed in the United States.

Under copyright laws, no part of this document may be copied or reproduced in any form without prior written consent of VT MAK.

VR-Exchange™, VR-TheWorld™, VR-Vantage™, DI-Guy™, and DI-Guy Scenario™ are trademarks of VT MAK. MAK Technologies®, VR-Forces®, RTIspy®, B-HAVE®, and VR-Link® are registered trademarks of VT MAK.

GL Studio® is a registered trademark of The DiSTI® Corporation.

Portions of this software utilize SpeedTree® technology (©2008 Interactive Data Visualization, Inc.). SpeedTree is a registered trademark of Interactive Data Visualization, Inc. All rights reserved.

SilverLining™ is a trademark of Sundog Software.

Terrain Profiles are based in part on the work of the Qwt project (<http://qwt.sourceforge.net>).

3ds Max® is a registered trademark of Autodesk, Inc.

All other trademarks are owned by their respective companies.

For third-party license information, please see “Third Party Licenses,” on page xiv.

VT MAK
150 Cambridge Park Drive, 3rd Floor
Cambridge, MA 02140 USA

Voice: 617-876-8085
Fax: 617-876-9208

info@mak.com
www.mak.com

Revision RTI-4.5-3-180312



Contents

Preface

How the Manual is Organized	vii
Documentation	viii
MAK Products	ix
How to Contact Us	xii
Document Conventions	xiii
The Windows Start Menu	xiv
Mouse Button Naming Conventions	xiv
Third Party Licenses	xiv
Boost License	xiv
libXML and libICONV	xv

Chapter 1. Introduction to the MAK RTI

1.1. The MAK RTI	1-2
1.1.1. MAK RTI Components	1-3
1.2. The rtiexec is not the RTI	1-4
1.3. Lightweight Mode	1-4
1.4. Compatibility with Other RTI Implementations	1-5
1.4.1. Obtaining HLA RTI Specification Documents	1-5
1.5. The RTI Assistant	1-6
1.5.1. The RTIspy Diagnostic GUI	1-6
1.6. The Unlicensed Version of the MAK RTI	1-7

Chapter 2. Installing the MAK RTI

2.1. Installing the MAK RTI on Windows	2-2
2.2. Installing the MAK RTI on a Linux System	2-2
2.3. Installing the MAK RTI Java Bindings	2-2
2.3.1. Linux Installation	2-3
2.3.2. Windows Installation	2-3
2.3.3. Installing the HLA 1516 Java Fedtime Library	2-3
2.4. Managing Licenses for the MAK RTI	2-5

2.5. Installing and Setting Up the MAK License Manager	2-8
2.5.1. Specifying the License Server	2-9
2.6. Localizing the Graphical User Interface	2-12
2.6.1. Translating Other Interface Files	2-15
2.6.2. Applying the Language Files	2-15

Chapter 3. HLA Concepts

3.1. The High Level Architecture	3-2
3.1.1. The RTI	3-2
3.1.2. Federates and Federations	3-2
3.2. RTI Services	3-2
3.2.1. Federation Management	3-3
3.2.2. Declaration Management	3-3
3.2.3. Object Management	3-3
3.2.4. Ownership Management	3-3
3.2.5. Time Management	3-3
3.2.6. Data Distribution Management	3-4
3.3. FOMs and FED Files	3-4
3.4. RTI Settings and the RTI Initialization Data (RID) File	3-5

Chapter 4. Connecting Federates to the MAK RTI

4.1. Connecting a Federate to the MAK RTI	4-2
4.1.1. Troubleshooting Federate Connection Problems	4-6
4.2. Starting the rtiexec	4-7
4.2.1. Starting the rtiexec when Starting a Federate	4-11
4.2.2. Starting the rtiexec from the Federations View	4-13
4.2.3. Shutting Down the rtiexec	4-14
4.3. Starting an RTI Forwarder	4-14
4.3.1. Shutting Down an RTI Forwarder	4-14
4.4. Resigning a Federate from a Federation	4-15

Chapter 5. Managing Federations

5.1. Opening the Federations View	5-2
5.2. Destroying Federations	5-2
5.2.1. Shutting Down all Local RTI Components	5-3
5.3. Removing Disconnected Federates from the Display	5-4

Chapter 6. The RTI Assistant

6.1. Starting the RTI Assistant	6-2
6.2. The RTI Assistant Windows	6-2
6.2.1. Federations View	6-3
6.2.2. RTI Preferences Dialog Box	6-6
6.2.3. Local Component Notification History Window	6-7
6.3. Configuring the RTI Assistant's Startup Behavior	6-7
6.4. Specifying the RTI Assistant's Exit Policy	6-8
6.5. Configuring the RTI's Crash Behavior	6-8
6.6. Connecting to Remote Assistants	6-9
6.6.1. Adding an Assistant Connection	6-10
6.6.2. Removing an Assistant Connection	6-11

6.6.3. Connecting to or Disconnecting from an Assistant Connection	6-11
6.7. Shutting Down the RTI Assistant	6-11
Chapter 7. Configuring the MAK RTI Startup Environment	
7.1. Configuring Your System for the MAK RTI	7-2
7.1.1. MAK RTI Environment Variables	7-3
7.1.2. Specifying RTI Settings Programmatically in HLA 1516 ..	7-3
7.2. Configuring the RTI Version and Settings	7-4
7.2.1. Adding an RTI Configuration	7-6
7.2.2. Editing an RTI Configuration	7-7
7.3. Managing Connection Configurations	7-8
7.3.1. Adding rtiexec Connection Configurations	7-10
7.3.2. Adding RTI Connection Configurations	7-12
7.3.3. Editing Connections	7-14
7.3.4. Viewing Connection Status	7-15
7.3.5. Removing Connection Configurations	7-15
7.4. Default Port Values	7-16
Chapter 8. Licensing Issues	
8.1. Configuring a Federate to Let the rtiexec Check Out a License	8-2
8.2. Unlicensed Federations	8-3
8.3. Troubleshooting License Problems	8-4
8.3.1. Specifying a Default License Server	8-5
8.3.2. Displaying Default License Server Settings	8-5
8.3.3. Clearing Default License Server Settings	8-6

MAK Products Glossary

Index



Preface

This manual is for persons who want to use the MAK RTI to run HLA federates. It is oriented towards customers who simply want to run their applications with a minimum of attention to configuration and network topology issues. Developers who need to design network topologies for HLA exercises, develop HLA applications, or use the RTIspy API should read *MAK RTI Reference Manual*.

This manual assumes that you are familiar with the command-line and graphical windowing environments for your operating system.

For information about changes to the MAK RTI since this manual went to press, please see *MAK RTI Release Notes*.

How the Manual is Organized

This manual is organized as follows:

Chapter 1, *Introduction to the MAK RTI* describes the MAK RTI and its features.

Chapter 2, *Installing the MAK RTI* describes installation and configuration tasks, including setting up the license server.

Chapter 3, *HLA Concepts*, provides conceptual information about the HLA so that you can understand the terminology used elsewhere in the manual.

Chapter 4, *Connecting Federates to the MAK RTI*, describes how to connect a federate to the MAK RTI, how to start the rtiexec, and how to remove federates from a federation.

Chapter 5, *Managing Federations* describes some basic federation management procedures, primarily how to remove inactive federates and how to destroy a federation.

Chapter 6, *The RTI Assistant* provides a high-level introduction to the RTI Assistant.

Chapter 7, *Configuring the MAK RTI Startup Environment*, explains how to configure your system to use a particular MAK RTI version and RTI settings. It also explains how to configure connections that the rtiexec and federates can use when they start.

Chapter 8, *Licensing Issues* explains how to let the MAK RTI check out licenses and how to troubleshoot license issues.

The *MAK Products Glossary* defines terms common to HLA and real-time simulations.

Documentation

The MAK RTI documentation set is as follows:

- ♦ *MAK RTI Users Guide* is a brief introduction to HLA and the MAK RTI and explains how to configure the MAK RTI startup environment and run a federate.
- ♦ *MAK RTI Reference Manual* provides detailed information about configuring the MAK RTI and designing network topologies for HLA federations.
- ♦ *MAK RTI Release Notes* provides system information, new features and updates, a list of fixed bugs, and a list of known problems.
- ♦ MAK RTIspy API class documentation and MAK RTI API developer documentation in HTML format. You can access the class documentation from the Start menu in Windows or by opening `./classdoc/index.html`.
- ♦ Online help. The RTI Assistant has online help.
- ♦ *MAK Products Interoperability Guide* explains how MAK products work together and provides troubleshooting information.

MAK Products

The MAK RTI is a member of the VT MAK line of software products designed to streamline the process of developing and using networked simulated environments. The VT MAK product line includes the following:

- ♦ **VR-Link® Network Toolkit.** VR-Link is an object-oriented library of C++ functions and definitions that implement the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS) protocol. VR-Link has built-in support for the RPR FOM and allows you to map to other FOMs. This library minimizes the time and effort required to build and maintain new HLA or DIS-compliant applications, and to integrate such compliance into existing applications.

VR-Link includes a set of sample debugging applications and their source code. The source code serves as an example of how to use the VR-Link Toolkit to write applications. The executables provide valuable debugging services such as generating a predictable stream of HLA or DIS messages, and displaying the contents of messages transmitted on the network.

- ♦ **MAK RTI.** An RTI (Run-Time Infrastructure) is required to run applications using the High Level Architecture (HLA). The MAK RTI is optimized for high performance. It has an API, RTIsPy®, that allows you to extend the RTI using plug-in modules. It also has a graphical user interface (the RTI Assistant) that helps users with configuration tasks and managing federates and federations.
- ♦ **VR-Forces®.** VR-Forces is a computer generated forces application and toolkit. It provides an application with a GUI, that gives you a 2D and 3D views of a simulated environment.

You can create and view entities, aggregate them into hierarchical units, assign tasks, set state parameters, and create plans that have tasks, set statements, and conditional statements. You can simulate using entity-level modeling, which focuses on the actions of individual people and vehicles, and aggregate-level modeling, which focuses on the interaction of large hierarchical units.

VR-Forces also functions as a plan view display for viewing remote simulation objects taking part in an exercise. Using the toolkit, you can extend the VR-Forces application or create your own application for use with another user interface.

- ♦ **VR-Vantage™.** VR-Vantage is a line of products designed to meet your simulation visualization needs. It includes three end-user applications (VR-Vantage Stealth, VR-Vantage PVD, and VR-Vantage IG) and the VR-Vantage Toolkit.
 - VR-Vantage Stealth displays a realistic, 3D view of your virtual world, a 2D plan view, and an exaggerated reality (XR) view. Together these views provide both situational awareness and the big picture of the simulated world. You can move your viewpoint to any location in the 3D world and can attach it to simulation objects so that it moves as they do.

- VR-Vantage IG is a configurable desktop image generator (IG) for out the window (OTW) scenes and remote camera views. It has most of the features of the Stealth, but is optimized for its IG function.
- VR-Vantage PVD provides a 2D plan view display. It gives you the big picture of the simulated world.
- SensorFX. SensorFX is an enhanced version of VR-Vantage that uses physics based sensors to view terrain and simulation object models that have been materially classified. It is built in partnership with JRM Technologies.
- The VR-Vantage Toolkit is a 3D visual application development toolkit. Use it to customize or extend MAK's VR-Vantage applications, or to integrate VR-Vantage capabilities into your custom applications. VR-Vantage is built on top of OpenSceneGraph (OSG). The toolkit includes the OSG version used to build VR-Vantage.
- ♦ MAK Data Logger. The Data Logger, also called the Logger, can record HLA and DIS exercises and play them back for after-action review. You can play a recorded file at speeds above or below normal and can quickly jump to areas of interest. The Logger has a GUI and a text interface. The Logger API allows you to extend the Logger using plug-in modules or embed the Logger into your own application. The Logger editing features let you merge, trim, and offset Logger recordings.
- ♦ VR-Exchange™. VR-Exchange allows simulations that use incompatible communications protocols to interoperate. For example, within the HLA world, using VR-Exchange, federations using the HLA RPR FOM 1.0 can interoperate with simulations using RPR FOM 2.0, or federations using different RTIs can interoperate. VR-Exchange supports HLA, TENA, and DIS translation.
- ♦ VR-TheWorld™ Server. VR-TheWorld Server is a simple, yet powerful, web-based streaming terrain server, developed in conjunction with Pelican Mapping. Delivered with a global base map, you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private, classified networks to provide streaming terrain data to a variety of simulation and visualization applications behind your firewall.
- ♦ DI-Guy™. The DI-Guy product line is a set of software tools for real-time human visualization, simulation, and artificial intelligence. Every DI-Guy software offering comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy enables the easy creation of crowds and individuals who are terrain aware, autonomous, and react intelligently to ongoing events. Save time, money and create outstanding simulations with DI-Guy. The DI-Guy product line includes the following products:
 - The DI-Guy SDK. Embed the DI-Guy library in your real-time application and populate your world with lifelike human characters.
 - DI-Guy Scenario™. Author and visualize human performances in a rich, user-friendly graphical environment. Use DI-Guy Scenario as an end visualization application or save scenarios and load them into your DI-Guy SDK enabled application.

- ECOSim. Enhanced Company Operations Simulation (ECOSim) is a company-level training simulation that teaches leaders how best to deploy troops, UAVs, convoys, and other assets. ECOSim focuses on ease-of-use, rapid scenario generation, runtime operator control, and realistic and reactive human simulation.
- DI-Guy AI. Generate crowds of autonomous characters to quickly populate your worlds with hundreds and thousands of terrain-aware, collision avoiding DI-Guys. Used as a module on top of DI-Guy Scenario and DI-Guy SDK.
- Expressive Faces Module. Enable DI-Guy characters to have faces that display emotion, eyes that look in directions and blink, and lips that sync to sound files.
- DI-Guy Motion Editor. Create or customize motions to your particular needs in an easy-to-use graphical application.
- ♦ RadarFX. RadarFX is a client-server application that simulates synthetic-aperture radar (SAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests SAR images from the server. RadarFX includes a sample client application.
- ♦ VR-Engage. VR-Engage is a multi-role virtual simulator that lets users play the role of a first person human character, a ground vehicle driver, gunner or commander, or the pilot of a fixed wing aircraft or helicopter. It incorporates the VR-Force simulation engine and the VR-Vantage graphics rendering capabilities.
- ♦ WebLVC Server. WebLVC Server implements the server side of the WebLVC protocol so that web-based simulation federates can participate in a distributed simulation. It is part of the WebLVC Suite, which includes the server and several sample applications that work with VR-Forces and VR-Vantage.

How to Contact Us

For RTI technical support, information about upgrades, and information about other MAK products, you can contact us in the following ways:

Telephone

Call or fax us at:	Voice:	617-876-8085 (extension 3 for support)
	Fax:	617-876-9208

E-mail

Sales and upgrade information:	info@mak.com
Technical support:	support@mak.com

Internet

MAK web site home page:	www.mak.com
License key requests:	www.mak.com/support/get-licenses
Product version and platform information:	www.mak.com/support/product-versions
For the free, unlicensed MAK RTI:	www.mak.com/support/bonus-material

Post

Send postal correspondence to:	VT MAK 150 Cambridge Park Drive, 3rd Floor Cambridge, MA, USA 02140
--------------------------------	---

When requesting support, please tell us the product you are using, the version, and the platform on which you are running.

Document Conventions

This manual uses the following typographic conventions:

Monospaced	Indicates commands or values you enter.
Monospaced Bold	Indicates a key on the keyboard.
Monospaced Italic	Indicates command variables that you replace with appropriate values.
Blue text	A hypertext link to another location in this manual or another manual in the documentation set.
{ }	Indicates required arguments.
[]	Indicates optional arguments.
	Separates options in a command where only one option may be chosen at a time.
()	In command syntax, indicates equivalent alternatives for a command-line option, for example, <code>(-h --help)</code> .
/	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames. If you are running on a PC, substitute a \ (backslash) when you type pathnames.
<i>Italic</i>	Indicates a file name, pathname, or a class name.
sans Serif	Indicates a parameter or argument.
➤	Indicates a one-step procedure.
Menu → Option	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as: Choose File → Save .
	Click the icon to run a tutorial video in the default browser.
i	Indicates supplemental or clarifying information.
!	Indicates additional information that you must observe to ensure the success of a procedure or other task.

Directory names are preceded with dot and slash characters that show their position with respect to the RTI home directory. For example, the directory *makRtix.x4.5/doc* appears in the text as *./doc*.

The Windows Start Menu

Windows 7 and 8 support submenus on the Windows Start menu. MAK applications use these submenus to organize executables, documentation, and tools. However, Windows 10 has a flat organization that does not support submenus. All executables and documentation are simply listed under MAK Technologies.

In this manual, directions for accessing applications through the Start menu use the full menu path supported by Windows 7 and 8. If you are using Windows 10, look for the application under the MAK Technologies menu. If you would like to take advantage of submenus, you can create a toolbar on the Windows Taskbar. For details, see <https://www.mak.com/learn/blog/entry/tech-tip-from-fred-starting-mak-applications-on-windows-10>.

Mouse Button Naming Conventions

An instruction to click the mouse button, refers to clicking the primary mouse button, usually the left button for right-handed mice and the right button for left-handed mice. The context-sensitive menu, also called a popup menu or right-click menu, refers to the menu displayed when you click the secondary mouse button, usually the right button on right-handed mice and the left button on left-handed mice.

Third Party Licenses

MAK software products may use code from third parties. This section contains the license documentation required by these third parties.

Boost License

VR-Link, and all MAK software that uses VR-Link uses some code that is distributed under the Boost License. All header files that contain Boost code are properly attributed. The Boost web site is: www.boost.org.

Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the “Software”) to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libXML and libICONV

VR-Link and all MAK software that uses VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MAK Products. MAK has made no modifications to these libraries. For more information about these libraries please see the following web sites:

- ♦ The LGPL license is available at: <http://www.gnu.org/licenses/lgpl.html>.
- ♦ Information about IconV is at: <http://www.gnu.org/software/libiconv/>.
- ♦ Information about LibXML is at: <http://xmlsoft.org/>.



1. Introduction to the MAK RTI

An RTI is an implementation of the Run-Time Infrastructure (RTI) portion of the High-Level Architecture (HLA).

The MAK RTI	1-2
MAK RTI Components.....	1-3
The rtiexec is not the RTI	1-4
Lightweight Mode.....	1-4
Compatibility with Other RTI Implementations.....	1-5
Obtaining HLA RTI Specification Documents	1-5
The RTI Assistant	1-6
The RTIsby Diagnostic GUI	1-6
The Unlicensed Version of the MAK RTI.....	1-7

1.1. The MAK RTI

The MAK RTI is a software library (and supporting executables) that implements the High Level Architecture (HLA) 1.3, 1516, and 1516-2010 (HLA Evolved) interface specifications. The HLA was developed by the U. S. Department of Defense (DoD) to provide a means for the many different types of simulations in use by the DoD to interoperate. In HLA, applications exchange data through RTI calls, which means that all HLA applications must use an RTI.

Since the function of the RTI is to manage exchange of data between applications (called federates) in a simulation exercise (called a federation), a person running a federate typically does not need to (or want to) know how the RTI works or is configured. They just want to start their federate and participate in the federation.

A useful analogy would be that of calling someone on a telephone. If you pick up a telephone, you enter a phone number (specify a connection) and the phone system does the rest. You do not need to know how the system sends its signals or connects to other phones.

This Users Guide is oriented towards the person who needs to use the MAK RTI the way they would use a telephone. It tells you how to start a federate and choose a connection. It also describes some of the user-oriented features of the MAK RTI. If you need to configure the network topology for your simulation, optimize the MAK RTI configuration, or use the RTIspy API to customize the MAK RTI, please refer to *MAK RTI Reference Manual*.

i

The MAK RTI has been verified by the Modeling and Simulation Coordination Office (MSCO, formerly DMSO) as fully compliant with the HLA Interface Specification, version 1.3 and with the IEEE 1516 version of the HLA Interface Specification (SISO DLC HLA API 1516 (SISO-STD-004.1-2004)). All services are implemented.

The US DoD RTI Verification activity was suspended due to budget constraints in January, 2015 before declaring any HLA Evolved (IEEE 1516.1-2010) RTIs officially verified. However, at the time that the RTI verification activity was suspended, the MAK RTI had successfully passed 100% of the available tests

1.1.1. MAK RTI Components

The MAK RTI consists of:

- ♦ The RTI library. A DLL or SO that implements the Local RTI Component (LRC).
- ♦ The rtiexec (short for RTI Executive). A server executable that provides services for the entire federation execution.
- ♦ The RTI Forwarder. An application that supports centralized TCP forwarding, distributed TCP forwarding, and distributed UDP forwarding. (Section 5.3, [“The RTI Forwarder Application”](#), in *MAK RTI Reference Manual*)
- ♦ The RTI Assistant. The RTI Assistant monitors federate and rtiexec activity and provides helpful messages when needed. It includes a network map and it provides access to error logs and configuration options. It is accessed through an icon in the system tray. (Chapter 6, [The RTI Assistant](#))
- ♦ The RTI Chooser (Windows only). The RTI Chooser helps you configure your MAK RTI environment. ([“Configuring the RTI Version and Settings,”](#) on page 7-4)
- ♦ The RTI Command Line Tool. This tool lets you send a limited number of commands to the RTI from a console window. (Section 5.4, [“The RTI Command Line Tool”](#), in *MAK RTI Reference Manual*)

All of these components are optional for running a federate except use of the RTI library.

1.2. The rtiexec is not the RTI

MAK RTI users are sometimes confused about the difference between the RTI executive (rtiexec) and local RTI components (LRCs) and what it means to install and run an RTI.

An LRC is the RTI library (DLL or .so) linked with a federate. The LRC provides an implementation of the HLA interface through which a federate interacts with the RTI and thus communicates with other federates in a federation.

The rtiexec is a centralized application that supports the RTI services that need centralized synchronization and control (for example, synchronization points, save/restore, and time management). The rtiexec is also required for some network monitoring features in the RTI Assistant. If you do not need these services, you can use the MAK RTI in lightweight mode, in which case you do not need to run the rtiexec. (For information about lightweight mode, please see [“Lightweight Mode,”](#) on page 1-4.)



Only one instance of the rtiexec should be run for a federation.

It is not accurate to equate running the rtiexec with running the MAK RTI. Nor is it sufficient to install and run the rtiexec to use the MAK RTI. Since each federate must have access to an LRC, you must install the MAK RTI on each computer on which a federate is located. For more information about the rtiexec, please see Section 5.1, [“The rtiexec”](#), in *MAK RTI Reference Manual*.

1.3. Lightweight Mode

The MAK RTI supports a lightweight mode, for which the rtiexec is not necessary. You can just run your federates, and all communication occurs directly between their LRCs. In general, lightweight mode is appropriate for small, simple federations that:

- ♦ Do not use the more complex RTI services.
- ♦ Require only best effort transport.
- ♦ Do not require strict compliance with the HLA Specification.

For more information about lightweight mode and when it is appropriate, please see Section 6.3, [“Using Lightweight Mode”](#), in *MAK RTI Reference Manual*.

1.4. Compatibility with Other RTI Implementations

The two questions that HLA users most frequently ask are:

- ♦ Do I have to rebuild my application to use the MAK RTI?

When you build an HLA application, you must link it to RTI libraries. If the libraries support a dynamic-link compatible HLA API, then the application can work with any RTI that uses that same API without rebuilding.

The MAK RTI 1.3 implements the API dictated by the 1.3 version of the HLA Interface Specification, as amended by MSCO's official HLA 1.3 Interpretations document. Therefore, any application built with the MAK RTI can work with any other RTI that supports the HLA 1.3 specification. Similarly any application built with an RTI that supports the HLA 1.3 specification can work with the MAK RTI.

The MAK RTI 1516 implements the SISO DLC HLA API¹ 1516 (SISO-STD-004.1-2004). Applications built with it are compatible with any RTI written to this API.

The MAK RTI HLA Evolved implements the IEEE 1516-2010 specification. Applications built with it are compatible with any RTI written to this API.

- ♦ Can I use different RTIs for the different federates in my federation?

No. Every federate in a federation must use the same RTI. Federates using the MAK RTI must use the same version of the MAK RTI.

Furthermore, as mentioned in “[The rtiexec is not the RTI](#),” on page 1-4, every federate must have a copy of the RTI libraries installed on its local computer.

1.4.1. Obtaining HLA RTI Specification Documents

The HLA 1.3 and IEEE 1516 specifications are copyrighted by their owners and MAK cannot distribute them to customers. You can get documentation for the RTI 1.3 and 1516 interface specifications at:

- ♦ <http://www.msco.mil/TechSpecs.html>
- ♦ <http://shop.ieee.org/store/> (IEEE 1516 specifications)
- ♦ www.sisostds.org (SISO DLC HLA API 1516).

1. Simulation Interoperability Standards Organization Dynamic Link Compatible High Level Architecture Application Program Interface

1.5. The RTI Assistant

The RTI Assistant is the user interface for the MAK RTI. It starts automatically when you log in or start a federate (this behavior is configurable) and monitors the status of all RTI-related activity on your computer. It also communicates with RTI Assistants on other computers on the network. The RTI Assistant lets you:

- ♦ Start the rtiexec.
- ♦ Start RTI Forwarders.
- ♦ Edit RTI settings.
- ♦ View federation information.
- ♦ View information about your network components.
- ♦ Configure connections.
- ♦ Configure some MAK RTI behaviors.
- ♦ Configure and run latency testing.
- ♦ Resign federates and shut down the rtiexec.
- ♦ View federate logs and notification history for federates running on the local machine.

1.5.1. The RTIsPy Diagnostic GUI

The RTIsPy Diagnostic GUI is a set of features integrated into the RTI Assistant for observing the rtiexec, RTI Forwarders, and federates in a federation. The views range from a network map that shows all the components in the federation, to inspection of the internals of a specific federate. You can also monitor multiple RTI components simultaneously for further analysis and side-by-side comparisons. You can:

- ♦ View a network map.
- ♦ View information about the federation and federates.
- ♦ View information about RTI Forwarders.
- ♦ View RTI settings.
- ♦ View an LRC's FOM subscriptions and publications.
- ♦ View a federate's registered and discovered objects.
- ♦ View the interactions a federate has sent and received.
- ♦ View a federate's log of calls invoked by the federate ambassador and RTI ambassador.
- ♦ Monitor a federate's LRC network performance.
- ♦ Profile a federate's interaction with the RTI API.

1.6. The Unlicensed Version of the MAK RTI

The MAK RTI supports an unlicensed (free) mode. The unlicensed mode permits two unlicensed federates to join a federation execution. It is identical to the licensed mode of the MAK RTI and includes support for all RTI services and features except the RTIspy plug-in capability.

The unlicensed version is ideal for evaluating the MAK RTI. It also makes it easy for customers who have MAK RTI licenses to test HLA federates in situations where they may not have access to their licenses, such as when working at home or traveling. You can download a copy of the MAK RTI that supports unlicensed mode at <https://www.mak.com/support/bonus-material>

For more information about running unlicensed federations, please see “[Unlicensed Federations](#),” on page 8-3.



2. Installing the MAK RTI

This chapter explains how to install the MAK RTI on a Windows or a Linux operating system. You must also install the license manager files. (For details, please see [“Installing and Setting Up the MAK License Manager,”](#) on page 2-8.)

Installing the MAK RTI on Windows	2-2
Installing the MAK RTI on a Linux System	2-2
Installing the MAK RTI Java Bindings	2-2
Linux Installation	2-3
Windows Installation.....	2-3
Installing the HLA 1516 Java Fedtime Library	2-3
Managing Licenses for the MAK RTI.....	2-5
Installing and Setting Up the MAK License Manager.....	2-8
Specifying the License Server	2-9
Localizing the Graphical User Interface	2-12
Translating Other Interface Files.....	2-15
Applying the Language Files	2-15

2.1. Installing the MAK RTI on Windows

Windows versions of the RTI are provided as executable installer files on DVD, or as downloaded files. The installers are named to indicate the compiler used to build that version of the RTI and whether it is a 32 bit or 64 bit version.

Before you install the MAK RTI, please read *MAK RTI Release Notes* to see if there are any special instructions for installation.



- ♦ The MAK RTI installation wizard asks if you want to change the PATH environment variable to load the newly installed RTI. You can add it to the PATH for the current user or all users. If you do not add it to one of these PATHs, when you run a federate, it will not be able to find this version of the RTI unless you set the PATH using the RTI Chooser.
 - ♦ You must have administrator privileges to install MAK products on Windows Vista.
-

- To install the RTI, run the installer. Follow the instructions in the installation wizard.

2.2. Installing the MAK RTI on a Linux System

Linux versions of the RTI are provided as compressed tar files on DVD, or as downloaded files. Before you install the RTI, please read *MAK RTI Release Notes* to see if there are any special instructions for installation.

To install the RTI on Linux:

1. Create the directory in which you want to install the RTI.
2. Copy the tar file to the install directory.
3. Uncompress and untar the file:

```
tar -vxzf application.tar.gz
```

where *application* is the product release identification.

2.3. Installing the MAK RTI Java Bindings

The MAK RTI provides a set of Java bindings for each version (1.3, 1516, HLA evolved). The Java binding is a thin layer of C++ code that exposes the native C++ API of the MAK RTI to Java applications.

To use the Java bindings, you must install the Java Development Kit (JDK) 1.6 or later and configure your system as described in the following sections.

2.3.1. Linux Installation

In addition to the normal MAK RTI setup, you must add or edit the following environment variables:

- ♦ Add `%MAK_RTIDIR%/lib/hla.jar` to the CLASSPATH environment variable.
- ♦ Add `%MAK_RTIDIR%/lib` to the LD_LIBRARY_PATH variable.
- ♦ HLA 1516 only: Install the MAK RTI HLA 1516 Java Fedtime Library as described in [“Installing the HLA 1516 Java Fedtime Library”](#).

2.3.2. Windows Installation

In addition to the normal MAK RTI setup, you must add or edit the following environment variables:

- ♦ Add the `%MAK_RTIDIR%\lib\hla.jar` to the CLASSPATH environment variable.
- ♦ `%MAK_RTIDIR%\lib\java` must be the first entry in your PATH environment variable.
- ♦ Add the `%MAK_RTIDIR%\lib` directory to the PATH environment variable.
- ♦ HLA 1516 only: Install the MAK RTI HLA 1516 Java Fedtime Library as described in [“Installing the HLA 1516 Java Fedtime Library”](#).

2.3.3. Installing the HLA 1516 Java Fedtime Library

HLA 1516 specifies that the federate developer shall provide a Logical Time implementation. For Java federates the MAK RTI offers two methods of doing this:

- ♦ You can create an HLA 1516 time implementation in Java and add the new time class files to the CLASSPATH environment variable.
- ♦ You can use the default time implementation provided by the MAK RTI. The default implements time as a Java double.

Configuring UNIX Systems to use the HLA 1516 Java Fedtime Library

Add or edit the following environment variables:

Linux

Add the full path of the directory containing `libjvm.so` to the LD_LIBRARY_PATH variable. This library is distributed as part of the JDK. Its location is system dependent.

Configuring Windows to use the HLA 1516 Java Fedtime Library

Add the full path to the directory containing `jvm.dll` to the PATH variable. This library is distributed as part of the JDK. Its location is system dependent.

Developing a Custom Logical Time Implementation

If you are developing a custom Logical Time implementation:

1. Add the custom classes to the CLASSPATH environment variable.
2. Create a new environment variable named RTI_JAVA_TIME_CLASS and set its value to the fully qualified name of the custom *LogicalTimeFactory* class. The fully qualified name is the name of the class preceded by the full package name. The package name must use “/” instead of “.” as a separator. For example, the fully qualified name for the default MAK RTI LogicalTimeFactory is *com/mak/makRti1516/time/DtLogicalTimeDoubleFactory*. In Java, this corresponds to:

```
package com.mak.makRti1516.time;
public class DtLogicalTimeDoubleFactory implements LogicalTimeFactory;
```

3. Create a new environment variable named RTI_JAVA_TIME_INTERVAL_CLASS and set its value to the fully qualified name of the custom *LogicalTimeIntervalFactory* class. The fully qualified name is the name of the class preceded by the full package name. The package name must use “/” instead of “.” as a separator. For example, the fully qualified name for the default MAK RTI LogicalTimeIntervalFactory is: *com/mak/makRti1516/time/DtLogicalTimeDoubleIntervalFactory*. In Java, this corresponds to:

```
package com.mak.makrti1516.time;
public class DtLogicalTimeDoubleIntervalFactory implements
    LogicalTimeIntervalFactory;
```

2.4. Managing Licenses for the MAK RTI

The MAK RTI rtiexec can check out RTI licenses for federates. When this feature is enabled, the federates do not need to have access to a license server or otherwise configure license management. This can be quite helpful when a federation has federates located on multiple LANs whose security configuration makes it difficult for a federate to connect to a license server.

When rtiexec license checkout is enabled, the rtiexec must have access to a license server. You must have a MAK RTI license for each federate that needs to join. If a federate tries to join the federation and there are no licenses available, the join fails.

The rtiexec and all of the federates using this feature must enable it. The rtiexec license checkout feature does not support dongle licenses.



The license checkout feature only applies to MAK RTI licenses. If you have other MAK products, you must still run a license server and configure license management to serve them. This feature is designed to manage licenses for federates that only need RTI licenses.

To enable rtiexec license checkout:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens (Figure 2-1).

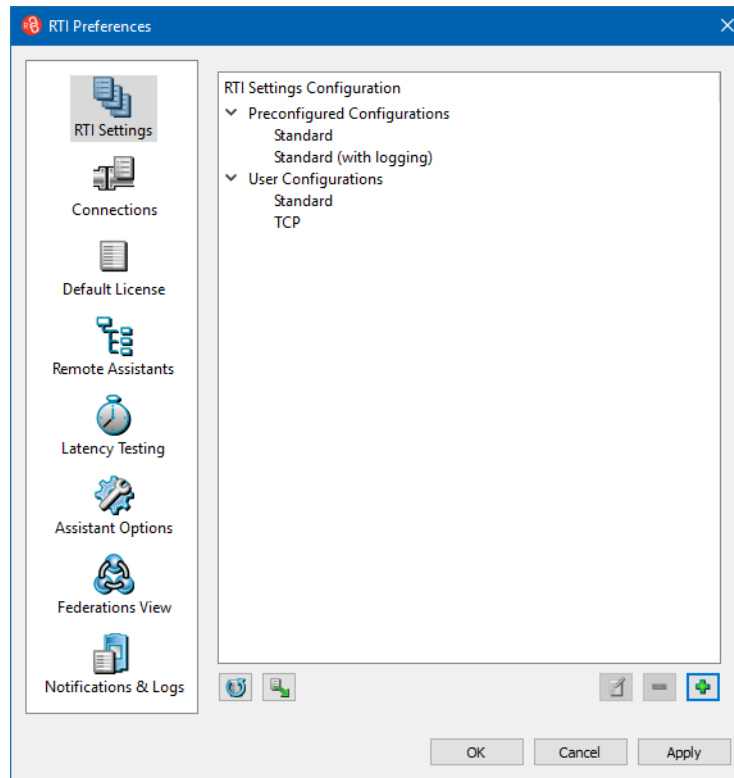


Figure 2-1. RTI Preferences

3. Select the RTI Settings page.
4. Select the settings configuration that you want to edit.
5. Click the Edit button (🔍). The RTI Settings Configuration dialog box opens. It has a set of pages with configuration settings.
6. Select the Licensing page (Figure 2-2).

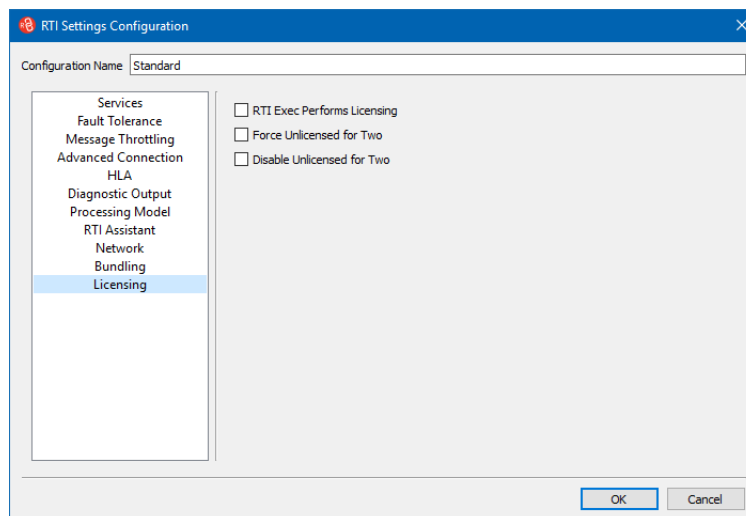


Figure 2-2. Licensing page

7. Select the RTI Exec Performs Licensing check box.
8. Click OK.

2.5. Installing and Setting Up the MAK License Manager

Before you can use a MAK product, you must obtain a valid license file, install the MAK License Manager, and configure the license server and client machines. The License Manager uses a client-server architecture, so you do not need to install the License Manager on every computer on which you install MAK products. You only need to install it on the computer that you will use as the license server.



If you have already installed the License Manager for another MAK product, you do not have to install it again. You just need to make sure you have licenses for your newly installed products.

The License Manager installer is included on MAK installation media. It is separate from the product installers. You can download the installers from our web site at <https://www.mak.com/support/license-support> or you can download directly from:

<ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe>

Complete installation and configuration instructions are included with the License Manager installer. Instructions are also available at the license support page.

Some customers use dongle licenses instead of running a license server. Instructions for using dongles are in the License Manager documentation.



This release requires a Vortex to control ground vehicles. Please see *VR-Engage Release Notes* for details.

2.5.1. Specifying the License Server

The first time you run a MAK application on a particular computer, the License Setup dialog box opens (Figure 2-3). It prompts you to enter the hostname of the license server and optionally, a port number.

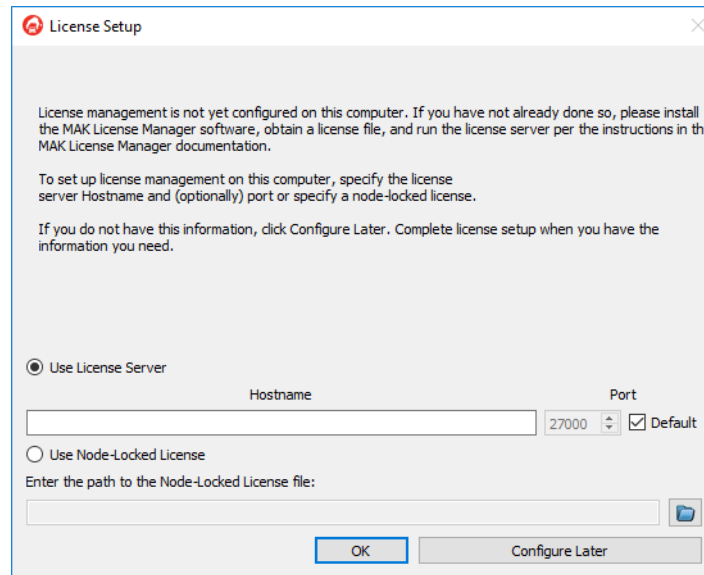


Figure 2-3. License Setup dialog box

If you do not know the hostname of the license server, click Configure Later. When you have the hostname, you can start the application again and complete the dialog box. You will not be able to run any MAK applications until you set up license management.

If you know the hostname, type it in the Hostname box. Then click OK. The application will start.

Under limited circumstances, MAK issues node-locked licenses. If you have a node locked license, select the Use Node Locked License option and enter the path to the license file.



- ♦ If you are running MAK products on the license server machine, it is also a client, so you must specify the license server on that machine too.
 - ♦ If you change the license server, the saved configuration will no longer be valid and the License Setup dialog box will open the next time you start a MAK application.
 - ♦ You can clear the saved license configuration by deleting the cache file. On Windows, it is `C:\Users\user_name\AppData\Roaming\MAK\licenses<n>.txt`. (The `AppData` directory is hidden by default.) On Linux, it is `.mak/license<n>.txt`. (There may be more than one cache file, for example, `licenses1.txt` and `licenses2.txt`.) You can also clear the cache in the RTI Assistant. For details, please see [“Clearing Default License Server Settings,”](#) on page 8-6.
-

The MAKLMGRD_LICENSE_FILE Environment Variable

As an alternative to using the License Setup procedure described in the previous section, you can configure the license server in an environment variable. The MAKLMGRD_LICENSE_FILE environment variable identifies the server machine. If you set this environment variable, it overrides the settings stored by the License Setup procedure.

The syntax for the environment variable is: `@Server_name`. For example, if the server machine is oak, set the environment variable to `@oak`.

The following sections explain how to set environment variables on the different platforms that MAK products run on.

Windows

To add the MAKLMGRD_LICENSE_FILE in Windows:

1. Open the Control Panel.
2. Click System and Security.
3. Click System.
4. In the sidebar menu, click the Advanced System Settings. The System Properties dialog box opens.
5. Click Environment Variables. The Environment Variables dialog box opens.
6. Click New. The New System Variable dialog box opens.
7. In the Variable Name field, enter MAKLMGRD_LICENSE_FILE.
8. In the Variable Value field, enter `@server_name`, where `server_name` is the name of the license server.
9. Click OK to back out of each dialog box and set the variable.



Different versions of Windows have slightly different wording of the various Control Panel options.

Linux

On Linux, you set environment variables in your *.cshrc* (or equivalent startup file). Set the variable similarly to the following example:

```
setenv MAKLMGRD_LICENSE_FILE @oak
```

If you are using the sh or bash shells, you set environment variables in your *.profile* file (or *.bashrc*). Set the variable similarly to the following example:

```
MAKLMGRD_LICENSE_FILE=@oak  
export MAKLMGRD_LICENSE_FILE
```

Do not put spaces around the equal (=) sign.

You are ready to run the license server and use your new licenses or MAK products.

2.6. Localizing the Graphical User Interface

You can localize your MAK product's graphical user interface so that it uses languages other than English. Your product includes a file called *./translations/product_untranslated.ts*, for example *logger_untranslated.ts* or *vrforces_untranslated.ts*. The file contains entries for all of the application-specific menu and dialog box text. The Qt Linguist tool, which is included with RTI, lets you map the English text provided by MAK to your chosen language.

To edit a resource file:

1. Start the Qt Linguist utility. It is *./translations/linguist.exe*
2. In the Qt Linguist window, choose **File** → **Open**. Select the file *./translations/product_untranslated.ts*. The Settings for filename dialog box opens (Figure 2-4).

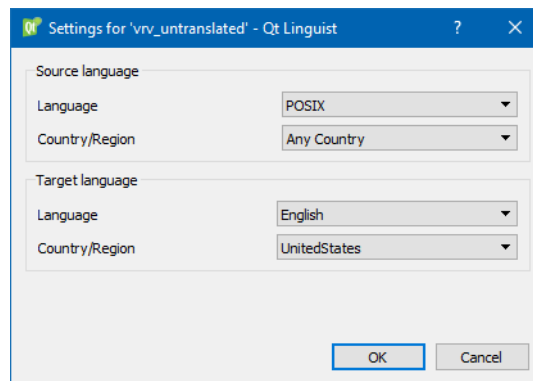


Figure 2-4. Linguist settings dialog box

3. Select values for the source language, target language, and target region or country.
4. Click OK. The translation file is opened and a list of classes is displayed on the left side of the window (Figure 2-5).

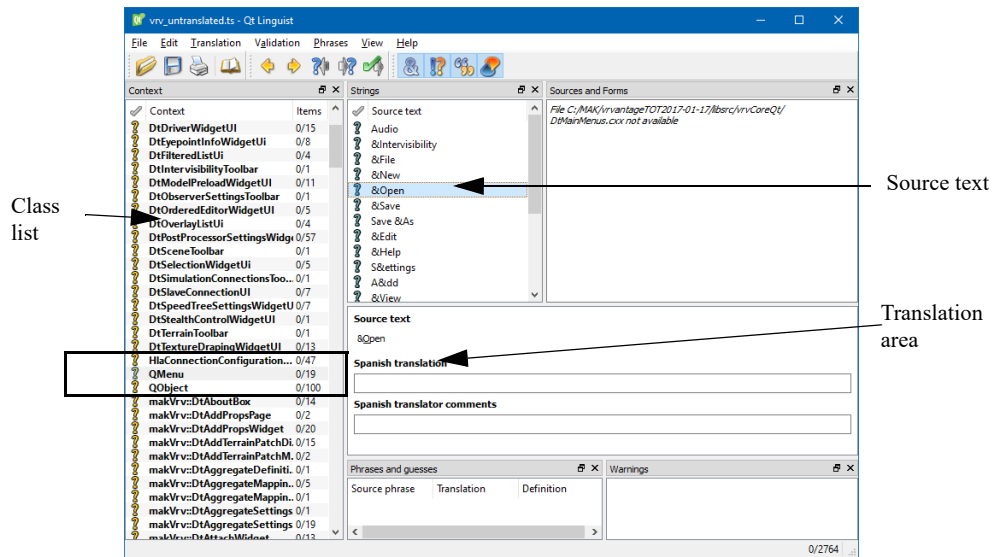


Figure 2-5. Linguist window

5. To translate text, you select each class in turn and translate the text associated with that class. You do not have to understand anything about what the classes mean, you just have to translate the text. In [Figure 2-5](#), *QMenu* is selected. The Strings window lists text that can be translated. Text that has not been translated has a question mark in the checkmark column.
6. In the Source Text list, select the word you want to translate. The text is displayed under Source Text in the translation area (the window below the source text list.) [Figure 2-6](#) shows the text Open selected.

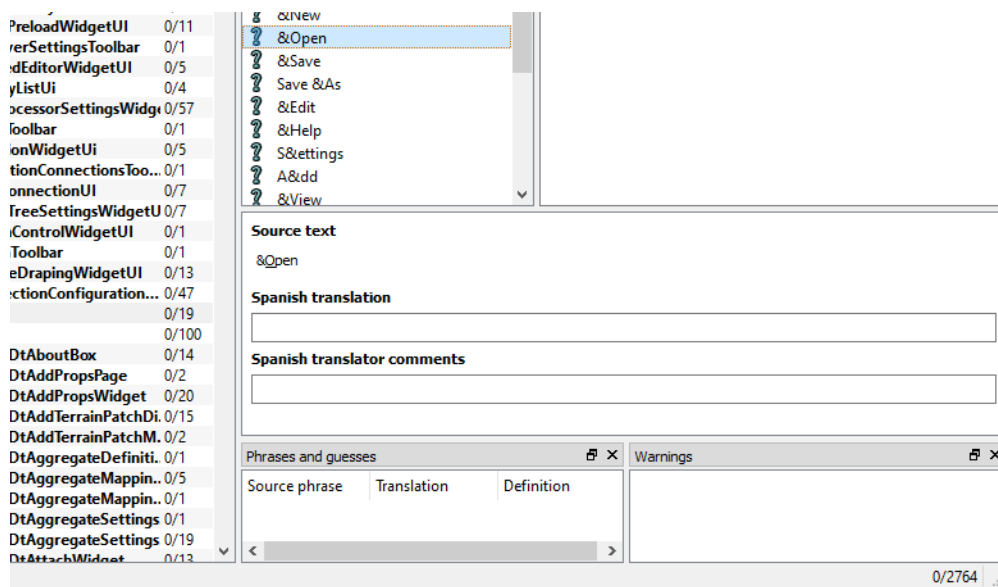


Figure 2-6. Text selected for translation

7. Place the cursor in the text box below *Language* Translation, where *Language* is the target language that you selected when you opened the translation file.
8. Type the translation text.
9. When you are done and you are certain of the translation, select the Done and Next button on the toolbar. A green check mark replaces the question mark (Figure 2-7). If you move to a different text string without clicking Done and Next, a yellow question mark is displayed for the string you translated. When you translate all elements for a class, a green check mark replaces the question mark next to the class name.

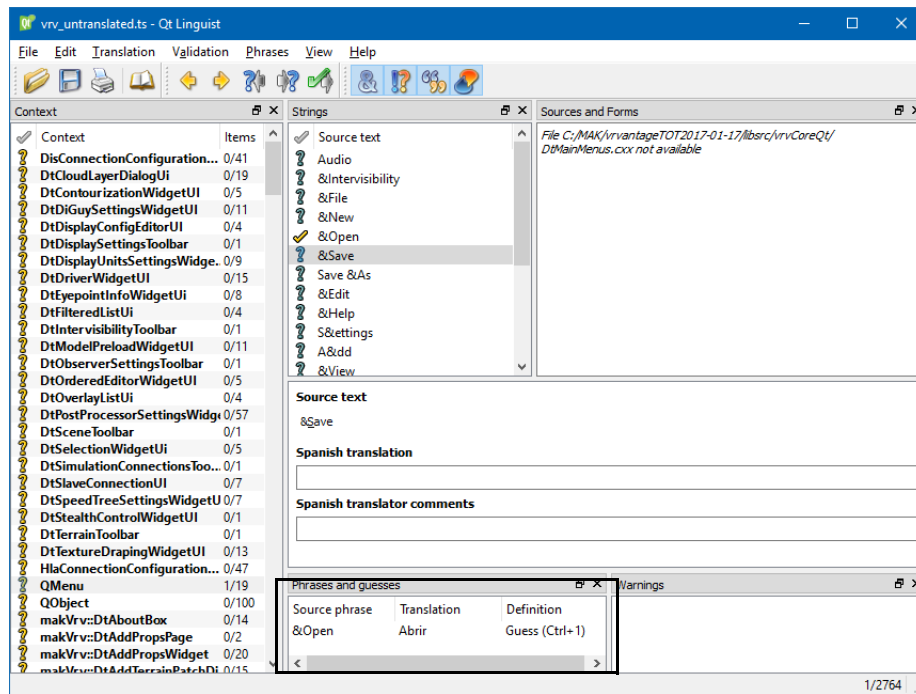


Figure 2-7. Completed translation for Open

10. Repeat this process for each element of source text for each class.
11. Choose **File** → **Release**. The file is saved in the format *filename.qm*.
12. Rename the file to a name that indicates the language to which you have translated the file, using the two-character abbreviation typically used with your language. For example, a Spanish translation would be renamed *product_es.qm*.
13. Save your translated *.ts* file.

2.6.1. Translating Other Interface Files

The *makRti_untranslated.ts* file contains all the text used in the RTI interface. However, there may be other text and messages that are not specific to the RTI that are generated by the software that implements the RTI GUI. To translate this text, you need to translate the file *qt_untranslated.ts*. Follow the same procedure as you did for *makRti_untranslated.ts*.

2.6.2. Applying the Language Files

To put the language file into effect, use one of the following methods:

- ♦ Set your computer's environment to the proper language (LANG environment variable).
- ♦ Set the locale to the proper language.
- ♦ Start the RTI with the *-G locale* command line option, where *locale* is the two-letter country code for the local language.



3. HLA Concepts

This chapter provides a brief introduction to HLA and background information about how MAK has implemented the RTI specifications. This information will help you understand and put to use the various configuration options described in later chapters.

The High Level Architecture	3-2
The RTI.....	3-2
Federates and Federations.....	3-2
RTI Services.....	3-2
Federation Management	3-3
Declaration Management	3-3
Object Management.....	3-3
Ownership Management	3-3
Time Management.....	3-3
Data Distribution Management.....	3-4
FOMs and FED Files.....	3-4
RTI Settings and the RTI Initialization Data (RID) File.....	3-5

3.1. The High Level Architecture

The High Level Architecture was developed by the U. S. Department of Defense to provide a means for the many different types of simulations in use by the DoD to interoperate. As an architecture, the HLA abstracts to a higher level how data is formatted and how it gets sent between simulations. Rather than the rigid and fixed protocol data units typical of DIS, the HLA provides a means for simulations to define their objects and interactions (the Federation Object Model (FOM)) and then provides an infrastructure (the RTI) to manage the data transmissions defined by the FOM.

3.1.1. The RTI

A Run-Time Infrastructure (RTI) is a software library (and possibly supporting executables) that implements the HLA Interface Specification. An RTI manages creation of federations and the distribution of data among federates. The HLA is conceived such that the RTI can distribute federation data without knowing anything about the structure or content of that data, while the federates can send data via the RTI without needing to know anything about the network details.

3.1.2. Federates and Federations

A federate is, technically, a connection to the RTI. Typically a single simulation application can be thought of as a federate.

A federation is a group of HLA federates capable of participating in the same federation execution.

A federation execution represents the actual operation, over time, of a subset of the federates and the RTI initialization data taken from a particular federation. A federation execution is the logical equivalent of a DIS simulation exercise.

3.2. RTI Services

An RTI provides the following set of services to a federation:

- ♦ Federation management
- ♦ Declaration management
- ♦ Object management
- ♦ Ownership management
- ♦ Time management
- ♦ Data distribution management.

These services are described briefly in the rest of this section.

3.2.1. Federation Management

Federation management provides services to the entire federation, including creating and destroying federations, allowing federates to join or resign from the federation, establishing synchronization points, and managing saves and restores.

3.2.2. Declaration Management

Declaration management services control publication of, and subscription to, objects and interactions. Individual federates declare which interactions and object attributes they will publish, that is, generate data for. Individual federates also declare which objects and interactions they want to subscribe to, that is, receive information about. The RTI ensures that federates receive only the data to which they have subscribed. This control of network traffic is one of the ways in which the HLA seeks to be more efficient than simulation methods that send all updates to all participants.

3.2.3. Object Management

Object management services allow a federate to generate objects and interactions that it has published. Federates can register new objects with the RTI (or acquire attribute ownership) and then send attribute updates as required. The RTI causes remote objects to be discovered and updated attributes to be reflected by federates that have subscribed to them. The object attributes represent persistent data whose state must be maintained between transmissions by both the sending and receiving federates (for example, the position of a vehicle). Federates can also send and receive interactions that typically represent a transitory event (for example, the collision of two vehicles, or a request for supplies).

3.2.4. Ownership Management

Ownership management services allow federates to seek ownership of objects they do not own and to give up ownership of objects to other federates.

3.2.5. Time Management

Time management is used in federations to coordinate the advancement of individual federates along a unified time axis. Time management is beneficial for federations that must maintain a causal ordering (that is, timestamp order) to the messages exchanged between federates. Because of the overhead involved in synchronization, time management is not usually used for real-time simulations, which can tolerate out of order messages, but have strict latency and throughput requirements.

3.2.6. Data Distribution Management

Data distribution management (DDM) provides filtering of data beyond that offered by declaration management. In declaration management, federates subscribe to interaction classes and object class attributes to express interest in the type of data. In DDM, the federate can define intervals that can effectively express an interest in a range of data values. A federation can specify dimensions (for 1.3 within routing spaces) that define intervals. Interactions and object attributes have a set of available dimensions that can be used to construct regions. The DDM services associate regions with interactions and object attributes for both sending (update regions) and receiving (subscription regions). An interaction or attribute update is delivered to a subscriber only when the subscription region intersects the update region. For details about how the MAK RTI implements DDM, please see Chapter 12, [Using Data Distribution Management](#), in *MAK RTI Reference Manual*.

3.3. FOMs and FED Files

A Federation Object Model (FOM) defines a federation's objects and interactions. The RTI manages the data transmissions defined by the FOM.

In both the HLA 1.3 specification and the 1516 specifications, you must provide FOM data to the RTI. In the 1.3 specification, this is done with the federation execution data (FED) file. The 1516 specifications use a FOM Document Data (FDD) file.



Since the FED file and FDD file perform the same function, and since for the MAK RTI they are essentially interchangeable, in MAK RTI documentation, unless otherwise noted, references to the FED file should be considered as references to the FED file and the FDD file.

A FED file contains information about your FOM that the RTI requires, including the names of all object and interaction classes, attributes, and parameters, the hierarchical relationships among the classes, and the default transport and order types to use for each class or attribute.

The LRC in each federate must read a valid FED file to successfully join a federation. It is very important that all federates read identical copies of the FED file, because unique handles for classes, attributes, and parameters are assigned based on their ordering in the FED file. You can configure the MAK RTI to have the RTI distribute the FED file used by the federate that creates the federation, or to have each LRC read it from a local disk. For more information, please see Section 6.2, [“Specifying the FED File”](#), in *MAK RTI Reference Manual*.

3.4. RTI Settings and the RTI Initialization Data (RID) File

The MAK RTI has many settings that control which services are enabled and how they function, network settings, performance, and other aspects of RTI configuration. These settings are stored in the RTI initialization data (RID) file. Although you can edit RTI settings using a text editor, the preferred method for editing settings is to use the RTI Assistant.

Because a RID file includes configuration parameters that are implementation-specific, there is no standard format for RID files. The default RID file for the MAK RTI is *rid.mtl*. For more information about the RID file, please see Chapter 4, [RTI Settings](#), in *MAK RTI Reference Manual*.



When this manual and *MAK RTI Reference Manual* refer to RTI settings, they refer to the parameters configured in the RID file.



4. Connecting Federates to the MAK RTI

This chapter explains how to connect a federate to the MAK RTI and how to start the rtiexec, if necessary.

Connecting a Federate to the MAK RTI	4-2
Troubleshooting Federate Connection Problems	4-6
Starting the rtiexec	4-7
Starting the rtiexec when Starting a Federate	4-11
Starting the rtiexec from the Federations View.....	4-13
Shutting Down the rtiexec.....	4-14
Starting an RTI Forwarder.....	4-14
Shutting Down an RTI Forwarder	4-14
Resigning a Federate from a Federation.....	4-15

4.1. Connecting a Federate to the MAK RTI

The HLA specification defines the information that a federate must provide to an RTI to create or join a federation. This information might be built into your application or you might have to specify it when you start the application. For example, applications based on VR-Link have to specify, at a minimum, a federation name and a federate name. Please consult your application's documentation for its startup procedures.

The MAK RTI uses the concept of an RTI connection. An RTI connection specifies a port and multicast address. All of the federates in a federation must use the same RTI connection. If a federation is using the rtiexec, an RTI connection includes the address of the RTI Forwarder being used by the rtiexec.

When you install the MAK RTI, it is configured with a default lightweight connection and a default rtiexec connection.

By default, when you start a federate, the MAK RTI prompts you to select an RTI connection configuration. If you always want to use the same connection, you can configure the MAK RTI to always try to use that connection without prompting you to select it.

You can also configure connections in the RTI settings file. (MAK's VR-Link-based applications have a common set of command-line options for specifying RTI connections.) However these configuration methods are ignored unless you explicitly tell the MAK RTI to use them. (For details about how to force use of the RTI settings file parameters, please see Section 7.4, "[Configuring Federate Connections in the RTI Settings](#)", in *MAK RTI Reference Manual*.)

To connect to the MAK RTI:

1. Start the federate. The Choose RTI Connection dialog box opens (Figure 4-1).



If the federate has a splash screen, it is possible that the Choose RTI Connection dialog box will be obscured by the splash screen. The federate startup process may appear to hang while the Choose RTI Connection dialog box is waiting for input. You will have to use the methods available in your windowing system to get access to the Choose RTI Connection dialog box and specify the RTI connection so that the startup process can continue.

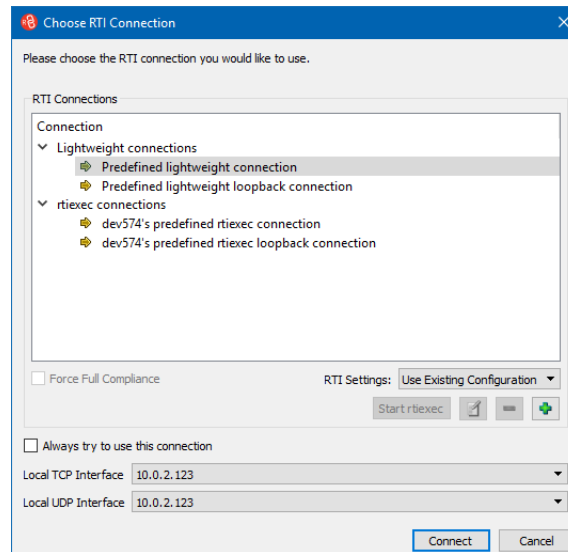


Figure 4-1. Choose RTI Connection dialog box

2. In the Connection window, select the connection that you want to use. (If the connection you want to use is not configured, you can create a new configuration. For details, please see Chapter 7, [Configuring the MAK RTI Startup Environment](#).) The dialog box lists all connections that are configured on the local computer and on any other computers on the network that are running RTI Assistants. Connections are organized into lightweight mode connections and rtiexec connections.

The rtiexec connections are color coded as follows:

- Green arrow: The RTI Assistant has verified that there is an rtiexec running on that connection.
- Yellow arrow: The RTI Assistant knows that a connection configuration exists, but does not know if the rtiexec is running. This might mean that you have to start an rtiexec on this connection, or it might mean that an rtiexec is running, but there is no RTI Assistant on the connection to report its status. You can try to connect to this connection, but it is likely that the connection will fail. (For details about starting the rtiexec, please see [“Starting the rtiexec when Starting a Federate,”](#) on page 4-11.)
- Red “Do Not Enter” icon: The federate cannot connect to it. This usually means that there is a conflict with another active connection. It might also mean that an RTI Forwarder is running on the connection, but there is no rtiexec running.

Lightweight connections are color coded as follows:

- Green arrow: The RTI Assistant has verified that there is a federate running on that connection. You can connect the federate to this connection.
- Yellow arrow: The RTI Assistant knows that a connection configuration exists, but does not know if any federates are using it. You can connect the federate to this connection.
- Red “Do Not Enter” icon: The federate cannot connect to it. This usually means that there is a conflict with another active connection.



If you hover the mouse over a connection, a tooltip displays information about the connection, including, for example, why you cannot use a particular connection ([Figure 4-2](#)).

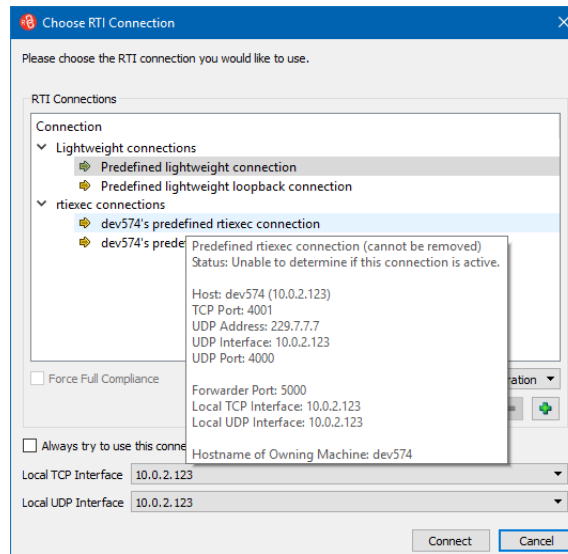


Figure 4-2. Connection details

3. If you want federates on this machine to always try to use the selected connection, select the Always Try to Use this Connection check box.
4. Optionally, specify a local TCP interface or a local UDP interface. In most cases the RTI Assistant chooses a valid network device. In rare cases, particularly if you are connecting over a VPN, you may need to specify the network device to use.
5. Click Connect. The federate will try to connect to the federation. If it cannot do so, the Connection Failed dialog box opens. For details about resolving connection problems, please see [“Troubleshooting Federate Connection Problems,”](#) on page 4-6.

4.1.1. Troubleshooting Federate Connection Problems

When you start a federate, the RTI Assistant monitors the connection process. If the connection fails, the RTI Assistant displays a dialog box indicating the problem (Figure 4-3).

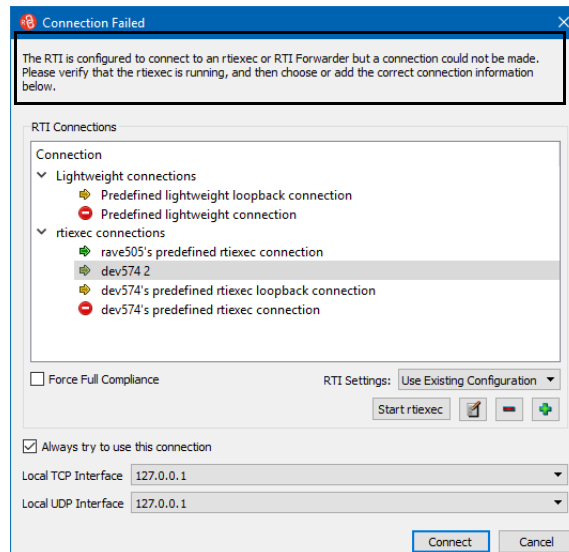


Figure 4-3. Connection Failed dialog box

The most likely reasons that a federate cannot connect are:

- The configuration requires the rtiexec and the rtiexec is not running
- If RID file parameters or command-line options are being used, the port specified does not match the port on which the rtiexec is running.

To connect a federate after a connection has failed, use one of the following methods:

- If the rtiexec is not running, start the rtiexec, then try to connect again. (For details about starting the rtiexec, please see [“Starting the rtiexec when Starting a Federate,”](#) on page 4-11.)
- In the Connection Failed dialog box, select a valid connection (green arrow). Then click Connect.
- Update the RID file or command line to have the correct connection information and restart the federate.
- Edit an existing connection configuration or add a new connection configuration that resolves problems with the available connections. For details, please see [“Managing Connection Configurations,”](#) on page 7-8.

4.2. Starting the rtiexec

Some federations require use of the rtiexec. When you start a federate, the rtiexec might already be running on the connection that you want to use. However, if it is not, you have to start it. You can start the rtiexec directly, or you can start it as part of the process of starting a federate. (For details, please see [“Starting the rtiexec when Starting a Federate,”](#) on page 4-11.)

When you start the rtiexec, you must specify a connection configuration (a port and broadcast address). You can specify connection parameters in the RTI settings, on the command line, or in the RTI Assistant’s Choose Connection Configuration dialog box (the default method). When you start the rtiexec using the RTI Assistant, you can start rtiexecs on remote computers.

Connection configurations override command-line options and settings in the RID file unless you force use of the command-line or RID file settings. For details, please see [“Managing Connection Configurations,”](#) on page 7-8.

i

If the rtiexec is configured to connect to an RTI Forwarder running on a remote computer, you must start the RTI Forwarder before you can start the rtiexec. For details about starting the RTI Forwarder, please see Section 5.3, [“The RTI Forwarder Application”](#), in *MAK RTI Reference Manual*.

To start the rtiexec from the Windows Start menu:

1. On the Start menu, choose **MAK Technologies** → **MAK RTI 4.5** → **rtiexec**. If the RTI Assistant is not already running, it starts and its icon is added to the system tray ([Figure 4-4](#)). The Choose Connection Configuration dialog box opens ([Figure 4-5](#)).

i

If the RTI Assistant has been configured to always try to use a particular connection and that connection is available, the Choose Connection Configuration dialog box is not displayed. The rtiexec starts immediately.



Figure 4-4. RTI Assistant icon

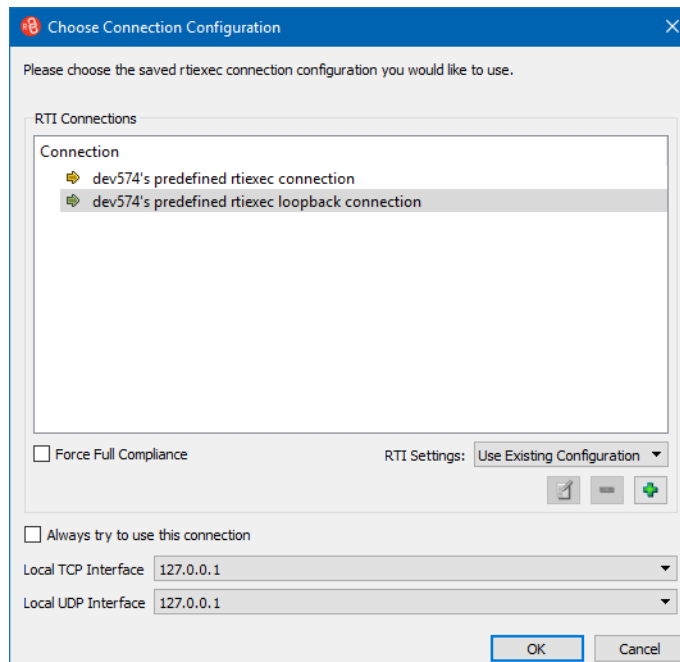


Figure 4-5. Choose Connection Configuration

2. Select the connection configuration that you want to use. (If necessary, edit an existing configuration or add a new one. For details, please see [“Managing Connection Configurations,”](#) on page 7-8.)

i

- ♦ If a connection has a yellow arrow, it probably means that there is no RTI Forwarder or rtiexec running on this connection. You can probably start the rtiexec on this connection.
- ♦ If a connection has a green arrow, it means that an RTI Forwarder is running on that connection, but not an rtiexec. You can start an rtiexec on this connection.
- ♦ If a configuration has a red icon, it is already in use. You cannot use it.

-
3. Optionally, select the Force Full Compliance check box to force the rtiexec to run in fully compliant mode. For details about full compliance, please see Section 5.1, [“The rtiexec”](#) and Section 11.1, [“Enabling and Disabling HLA Services”](#), in *MAK RTI Reference Manual*.
 4. If you always want to use the same connection, select the Always Try to Use This Connection check box. If you want to choose a connection each time you start the rtiexec, clear the check box.
 5. Optionally, specify the TCP and UDP interfaces to use.
 6. Click OK.

- To start the rtiexec on Linux or from the Windows command line, in the `./bin` directory, run rtiexec, for example:

```
rtiexec [options]
```

The rtiexec startup process is the same as described in the Windows startup procedure.



The rtiexec is a server application. Do not run more than one instance of the rtiexec per federation.

Table 4-1 lists command-line options for the rtiexec.

Table 4-1: rtiexec command-line options

Option	Description
(-- --ignore_rest)	Ignore all command-line options that follow this one.
(-A --destAddrString) address	Specifies the multicast group address to use if reliable transport is disabled. Equivalent to the RTI_destAddrString parameter. This option is ignored unless --manual is used.
(-d --doNotUseRoutesFile) address	Disables the use of a routes file for an RTI Forwarder started by this rtiexec.
(-D --distributedForwarder- Port) port	Specifies the port that RTI Forwarders use to communicate with each other.
(-f --forceFullCompliance)	Specifies that the rtiexec is to run in fully compliant mode. This option is ignored unless --manual is used.
(-F --forwarderToConnectTo) address	Specifies the address of an additional forwarder that this forwarder should try to connect to instead of using the routes file.
(-h --help)	Displays a list of command-line options.
(-i --tcpNetworkInter- faceAddr) address	The IP address of the local network interface used for TCP connections (reliable transport).
(-K --autoExit)	Specifies that the rtiexec exits automatically after the last connection exits.
(-l --setLogFileName) filename	Enables logging to a file with the specified name.

Table 4-1: rtiexec command-line options

Option	Description
(-M --manual)	Specifies that the rtiexec use the connection configuration in the RTI settings file or supplied on the command line rather than the configuration stored by the RTI Assistant. It is equivalent to setting RTI_configureConnectionWithRid to 1. Since command-line options override the RTI settings file, using this option enables you to specify the connection configuration from the command line.
(-n --notifyLevel) <i>level</i>	Sets the diagnostic output level.
(-N --udpNetworkInterfaceAddr) <i>address</i>	Specifies the IP address of the network card that you want this forwarder to use for UDP connections (best effort transport).
(-p --usePreconfiguredRid)	Indicates that the RID file specified by -R is the name of a preconfigured RTI Settings instead of a file path.
(-P --setUdpPort) <i>port</i>	Specifies the port to listen to. Equivalent to the RTI_udpPort parameter. This option is ignored unless --manual is used.
(-q --setNotifyQuiet)	Suppresses output of diagnostic messages.
(-r --useReliable)	Specifies that the rtiexec use reliable message transport. Equivalent to selecting Send Internal RTI Messages Reliable (TCP) and setting FOM Data Transport Type to "Specified by FOM" in the RTI Settings Configuration dialog box. This option is ignored unless --manual is used.
(-R --setRidFileName) <i>RID_file</i>	Specifies the RTI settings file to use. -R overrides the RTI_RID_FILE environment variable.
(-t --tcpAddress) <i>address</i>	The IP address used for TCP connections (reliable transport).
(-T --setTcpPort) <i>port</i>	Specifies the port to be used to connect via TCP to the RTI Forwarder. Equivalent to the RTI_tcpPort parameter. This option is ignored unless --manual is used.
(-v --version)	Display version information and exit.

4.2.1. Starting the rtiexec when Starting a Federate

When you start a federate it either displays the Choose RTI Connection dialog box (Figure 4-6) or it tries to connect to a federation using the default connection configuration. If a federate fails to connect to the default connection, the MAK RTI displays the Connection Failed dialog box (Figure 4-7). Both of these dialog boxes let you choose the connection configuration to use, and if necessary, start the rtiexec.

The RTI Assistant lets you start an rtiexec on the local computer or on a remote computer. When you start the rtiexec, it starts an RTI Forwarder on the same computer. If it cannot run the forwarder, it exits. (You cannot run the rtiexec unless a forwarder is running.)

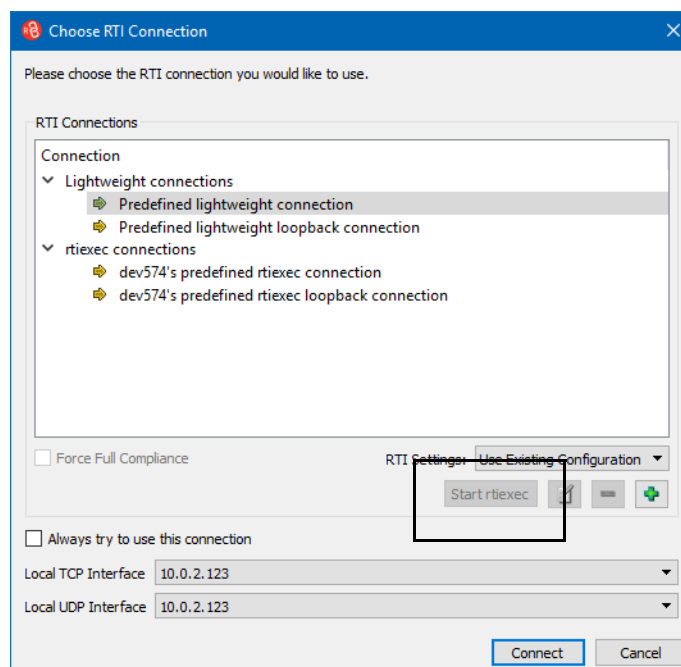


Figure 4-6. Choose RTI Connection dialog box

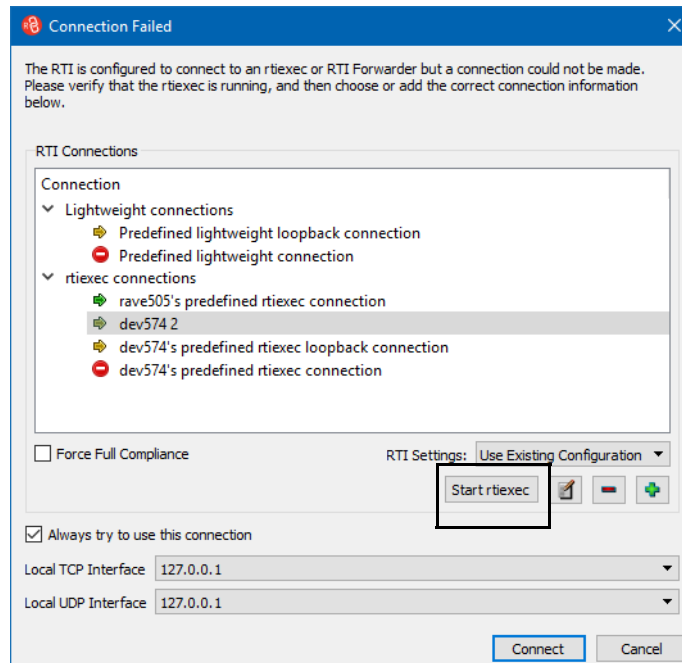


Figure 4-7. Connection Failed dialog box

To start the rtiexec from the Choose RTI Connection dialog box or Connection Failed dialog box:

1. In the dialog box, in the RTI Connections window, select the connection that you want to use for the rtiexec. If this is a valid connection on which to run an rtiexec, the Start rtiexec button is enabled.
2. Click Start rtiexec. The rtiexec starts on the computer specified by the connection.
3. To connect the federate, click Connect.

4.2.2. Starting the rtiexec from the Federations View



You cannot start an rtiexec if one is already running in the selected connection.

To start the rtiexec from the Federations View:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Federations View**. The Federations View opens (Figure 4-8).

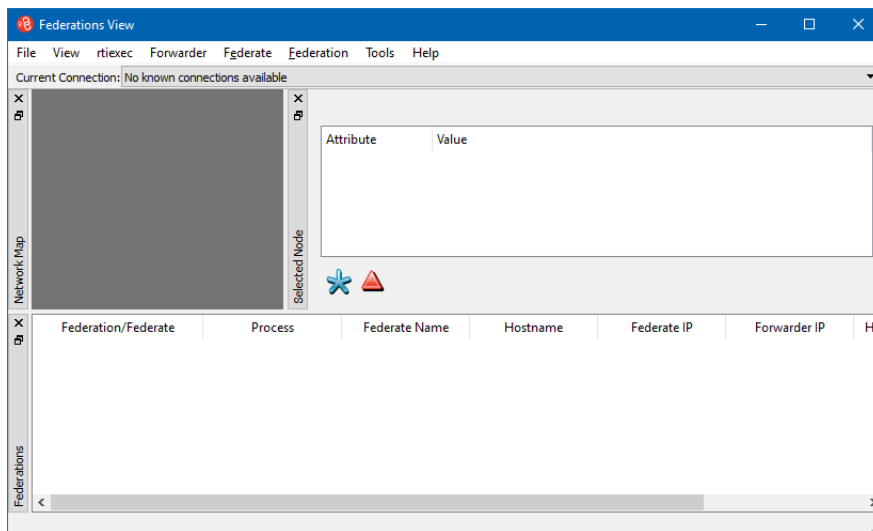


Figure 4-8. Federations View - no rtiexec running

3. Choose **rtiexec** → **Start Up Executive**, or click the Start Up Executive button (🔴) on the Selected Node toolbar.
4. The Choose Connection Configuration dialog box opens.
5. In the RTI Connections group box, select the connection you want to use.
6. Optionally, select the Force Full Compliance check box.
7. Optionally select an RTI Settings configuration.
8. Click OK.

4.2.3. Shutting Down the rtiexec

To shut down the rtiexec, use one of the following procedures:

- ♦ In the Federations View, select the rtiexec you want to shut down and choose **rtiexec** → **Shut Down rtiexec**. (This is the preferred method.)
- ♦ In the Federations View, click the Shut Down rtiexec button  on the Selected Node view.
- ♦ Kill the rtiexec process.
- ♦ Use the RTI Command Line Tool to send a `kill` command. For details, please see Section 5.4, “[The RTI Command Line Tool](#)”, in *MAK RTI Reference Manual*.
- ♦ Choose **Shut Down All** on the RTI Assistant menu. (This may have unintended consequences. It is not recommended for normal shutdown of rtiexecs and federates.)

4.3. Starting an RTI Forwarder

When you start the rtiexec, it automatically starts a forwarder. However, you can start an RTI Forwarder independently.

To start an RTI Forwarder:

1. In the system tray, right-click the RTI Assistant icon . A menu is displayed.
2. Choose **Federations View**. The Federations View opens ([Figure 4-9](#)).
3. Choose **Forwarder** → **Start Up Forwarder**, or click the Start Up Forwarder button  on the Selected Node toolbar.
4. The Choose Connection Configuration dialog box opens.
5. In the RTI Connections group box, select the connection you want to use. (If a forwarder is already running on a connection, the connection is unavailable.)
6. Click OK.

4.3.1. Shutting Down an RTI Forwarder

To shut down an RTI Forwarder

1. In the system tray, right-click the RTI Assistant icon . A menu is displayed.
2. Choose **Federations View**. The Federations View opens ([Figure 4-9](#)).
3. Choose **Forwarder** → **Shut Down Forwarder**, or click the Shut Down Forwarder button  on the Selected Node toolbar.

4.4. Resigning a Federate from a Federation

When you shut down a federate, it resigns from the federation. The RTI Assistant and RTIspy let you force a federate to shut down and resign from a federation.

To force a federate to resign from the federation:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Federations View**. The Federations View opens (Figure 4-9).

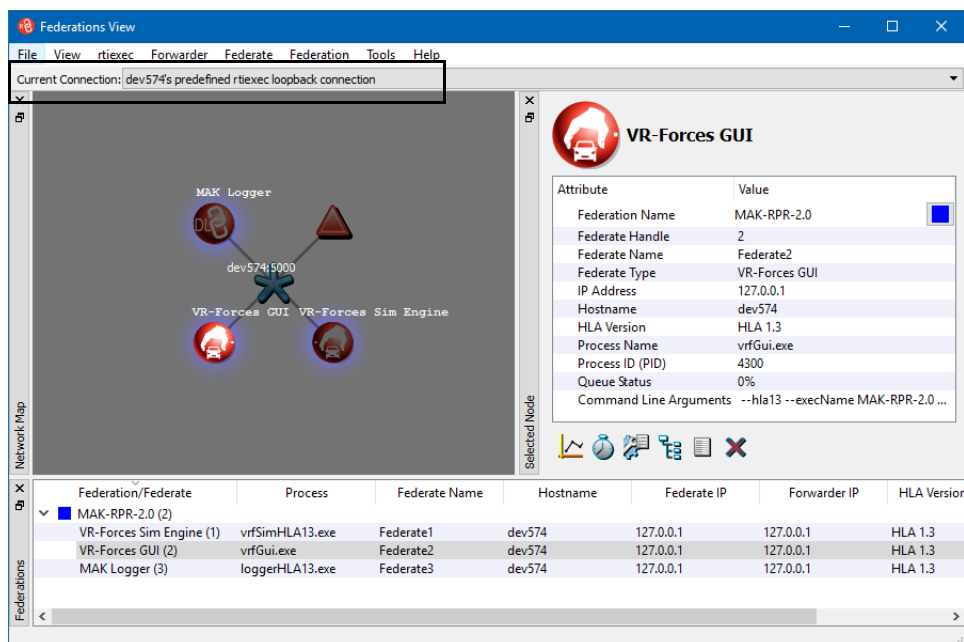


Figure 4-9. Federations View

3. In the Current Connection box, select the connection that you want to view.
4. In the Network Map or the Federations Panel, select the federate that you want to resign.
5. Choose **Federate** → **Resign Selected Federate**.



5. Managing Federations

Although this manual focuses on starting federates, with little attention to broader federation management issues, even if you are running only one federate, you are creating a federation. This chapter covers some basic federation management tasks.

Opening the Federations View	5-2
Destroying Federations	5-2
Shutting Down all Local RTI Components	5-3
Removing Disconnected Federates from the Display	5-4

5.1. Opening the Federations View

The RTI Assistant lets you view and manage federations in the Federations View (Figure 5-1).

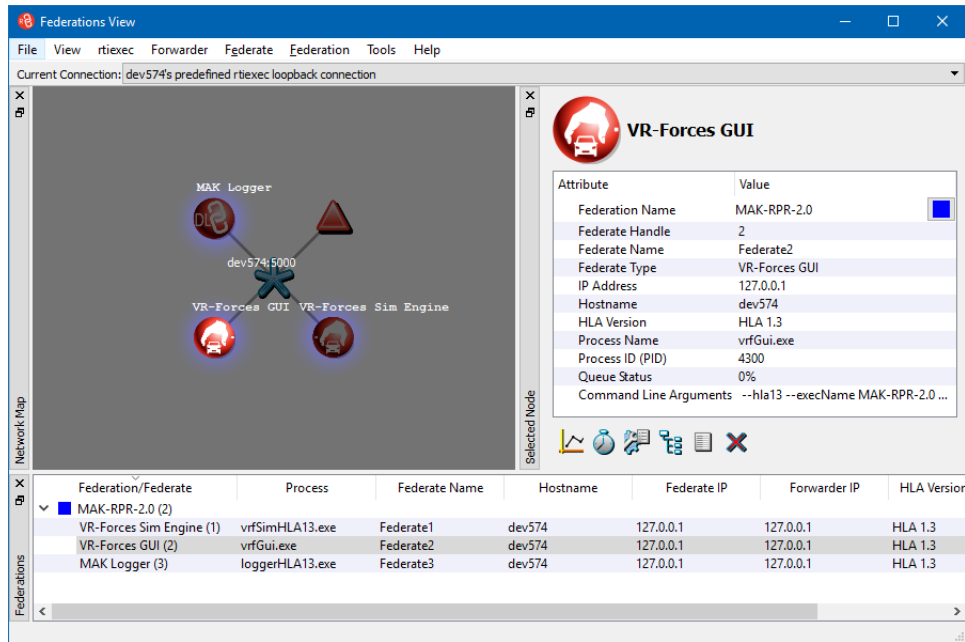


Figure 5-1. Federations View

To open the Federations View:

1. In the system tray, right-click the RTI Assistant icon (RTI Assistant icon). A menu is displayed.
2. Choose **Federations View**. The Federations View opens.

5.2. Destroying Federations

You can destroy federations explicitly in the Federations View. You can also destroy all local rtixecs and federations from the RTI Assistant menu.

To destroy a federation:

1. In the system tray, right-click the RTI Assistant icon (RTI Assistant icon). A menu is displayed.
2. Choose **Federations View**. The Federations View opens (Figure 5-1).
3. In the Current Connection list (at the top of the window), select the connection that you want to view.
4. In the Federations panel, select the federation that you want to destroy
5. Choose **Federation** → **Destroy Selected Federation**.

5.2.1. Shutting Down all Local RTI Components

The Shut Down All command kills all local forwarders and rtiexecs and federations that are using them. It does not affect federates running in lightweight mode. It does not affect remote rtiexecs, even if you started them from the local computer.



If remote federates are using the local rtiexec, shutting down all local RTI components could affect them.

To shut down all local components:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Shut Down All**.

5.3. Removing Disconnected Federates from the Display

When a federate terminates abnormally and does not resign from a federation, a disconnected federate entry is left in the federate list. You can configure the rtiexec to automatically remove disconnected federates from the display.

To automatically remove disconnected federates from the display:

1. In the system tray, right-click the RTI Assistant icon (🔴). A context menu opens.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Federations View page (Figure 5-2).

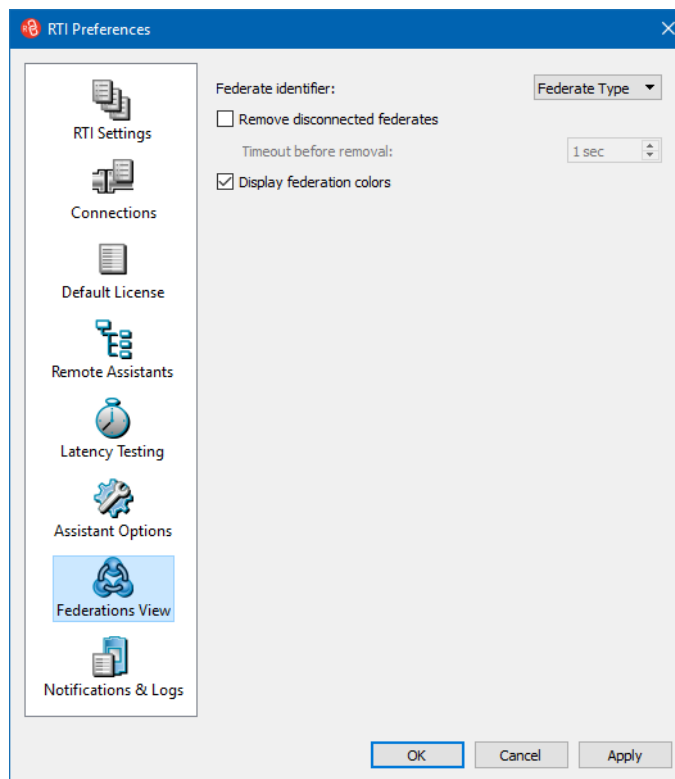


Figure 5-2. RTI Preferences dialog box, Federations View page

4. Select Remove Disconnected Federates.
5. Optionally, specify the time to wait before removing the federates.
6. Click OK.



6. The RTI Assistant

We have already referred to the RTI Assistant numerous times in this manual in the context of various procedures. This chapter focuses on the RTI Assistant itself. By default, the RTI Assistant remembers its state when you log off. If it was running when you logged off, it starts automatically when you log in. If it was off when you logged off, it does not start automatically when you log in. To configure this behavior, please see [“Configuring the RTI Assistant’s Startup Behavior,”](#) on page 6-7.

Starting the RTI Assistant	6-2
The RTI Assistant Windows	6-2
Federations View	6-3
RTI Preferences Dialog Box	6-6
Local Component Notification History Window	6-7
Configuring the RTI Assistant’s Startup Behavior	6-7
Specifying the RTI Assistant’s Exit Policy	6-8
Configuring the RTI’s Crash Behavior	6-8
Connecting to Remote Assistants	6-9
Adding an Assistant Connection	6-10
Removing an Assistant Connection	6-11
Connecting to or Disconnecting from an Assistant Connection	6-11
Shutting Down the RTI Assistant	6-11

6.1. Starting the RTI Assistant

If the RTI Assistant is not already running, it starts automatically when you start a federate, the rtiexec, or the RTI Forwarder. On Windows, you can also start the RTI Assistant directly.

- To start the RTI Assistant, on the Start menu, choose **MAK Technologies** → **MAK RTI 4.5** → **Utilities** → **RTI Assistant**. The RTI Assistant icon is added to the system tray ([Figure 6-1](#)).

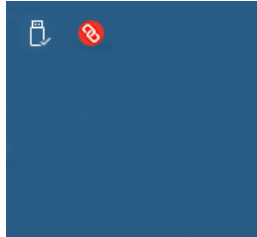


Figure 6-1. RTI Assistant icon

6.2. The RTI Assistant Windows

The RTI Assistant has the following windows and dialog boxes:

- ♦ Federations View
- ♦ RTI Preferences dialog box
- ♦ Local Notification History.

You can open any of these windows by right-clicking the RTI Assistant icon (🔴) in the system tray and selecting it from the menu.

6.2.1. Federations View

The Federations View (Figure 6-2) lists the federations running on each connection and the federates in those federations. It provides options for managing federations, viewing federation information, running latency testing, and starting the RTIsPy diagnostic GUI.

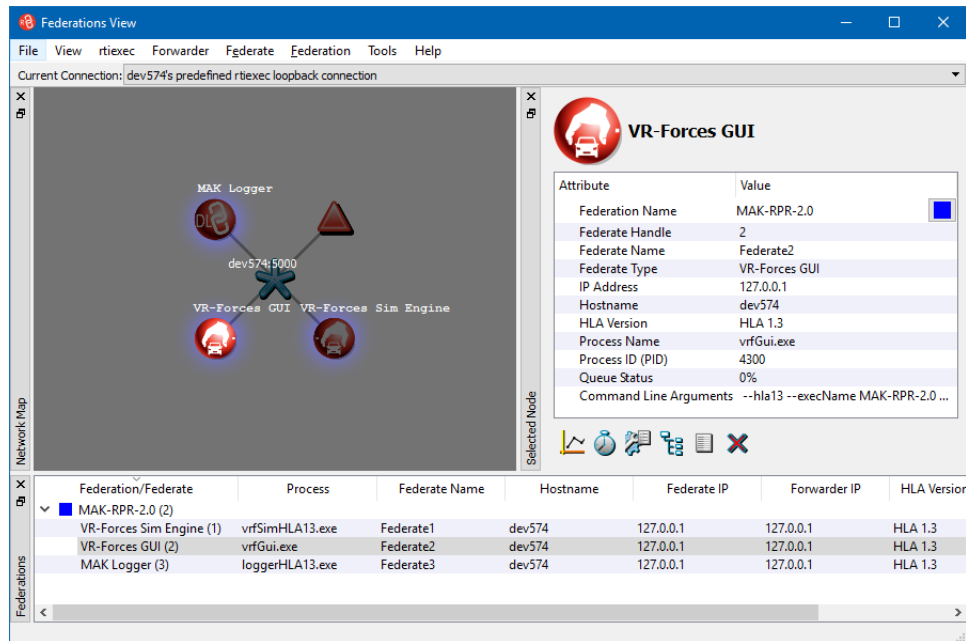


Figure 6-2. Federations View

The Federations View has the following panels:

- Network Map. The Network Map display icons for the RTI Forwarder, the rtiexec, and the federates in the selected connection.
- Selected Node. The Selected Node panel displays details about the node that is selected in the Network Map.
- Federations. The Federations panel lists the federations that are active on the selected connection and the federates in each federation.

The Federations panel lists the applications (federates, rtiexecs, and RTI Forwarders) running on the selected connection. It lists applications without regard to their participation in a federation. For example, an application may be running, but not joined to a federation.

To open the RTI Federations View:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Federations View**.



To quickly open the Federations View, double-click the RTI Assistant icon.

Configuring the Federation Color

Each federation listed in the Federations panel is assigned a color. This color is used to highlight members of the federation in the Network Map ([Figure 6-2](#)). You can change the color used for a federation.

To configure the color assigned to a federation:

1. In the Federations panel, select the federation whose color you want to change. Federation details are displayed in the Selected Node panel.
2. Click the color swatch next to the federation name. A color picker dialog box opens.
3. Select the color that you want to assign to this federation.
4. Click OK.

Changing a Federate's Icon

Federates are represented in the Network Map by an icon. This icon is also displayed on the Selected Node Panel. MAK applications are given their product icons. Other applications use a default icon. You can change the icon associated with an application.

To change the icon for a federate:

1. In the Network Map or Federations Panel, select the federate whose icon you want to change.
2. In the Selected Node Panel, double-click the icon displayed next to the federate's name. The Choose Federate Icon dialog box opens ([Figure 6-3](#)).

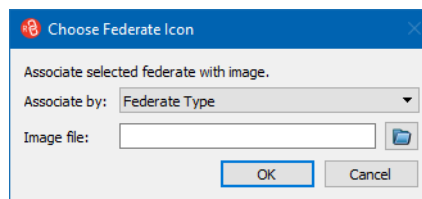


Figure 6-3. Choose Federate Icon dialog box

3. In the Associate By drop-down list choose an option. This setting specifies what information about a federate the RTI Assistant uses to decide what icon to use.
4. In the Image File field, type the path to an icon image or browse to find the image. You can choose BMP, JPG, PNG, GIF, JPEG, or SVG files.
5. Click OK.

6.2.2. RTI Preferences Dialog Box

The RTI Preferences dialog box (Figure 6-4) lets you configure RTI settings and the RTI Assistant. You can also configure latency testing from this dialog box.

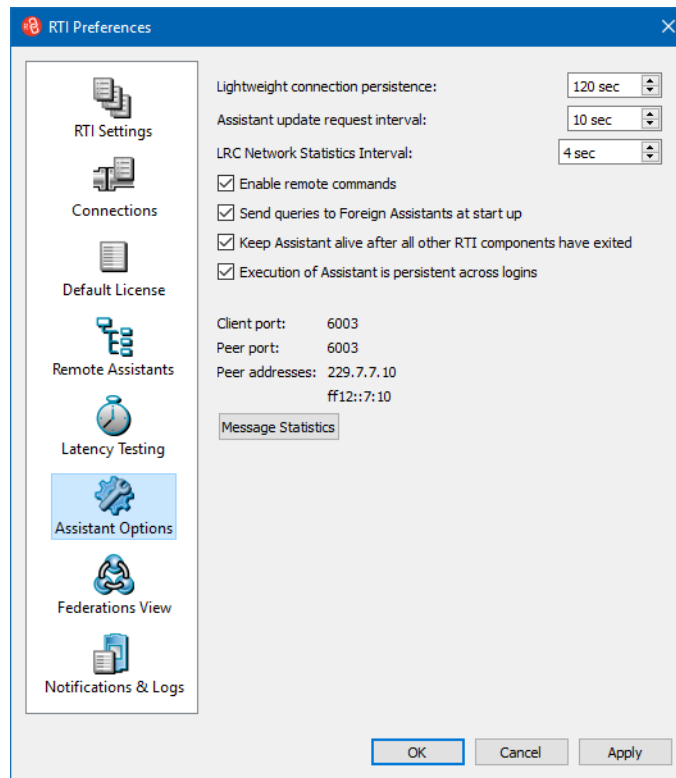


Figure 6-4. RTI Preferences dialog box

- To open the RTI Preferences dialog box, in the system tray, right-click the RTI Assistant icon (🔴) and choose Preferences.

6.2.3. Local Component Notification History Window

The Local Component Notification History window ([Figure 6-5](#)) displays notification messages sent by the local rtiexec and LRCs.

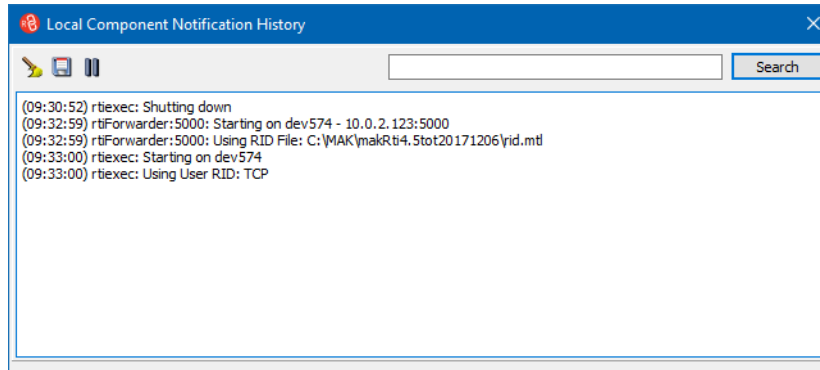


Figure 6-5. Local Component Notification History window

To open the Local Component Notification History window:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Local Notification History**.

6.3. Configuring the RTI Assistant's Startup Behavior

By default, when you log in, the RTI Assistant restores its status from when you logged off; if it was running when you logged off, it starts automatically; if it was not running it does not start.

To configure the RTI Assistant to never start automatically when you log in:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Assistant Options page ([Figure 6-4](#)).
4. Clear the Execution of Assistant is Persistent Across Logins check box.
5. Click OK.

6.4. Specifying the RTI Assistant's Exit Policy

You can configure the RTI Assistant to close when all federates and the rtiexec exit, or you can configure it to keep running until you exit manually.

To configure when the RTI Assistant exits:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Assistant Options page (Figure 6-4).
4. To keep the RTI Assistant running when federates and the rtiexec have shut down, select the Keep Assistant Alive After All Other RTI Components Have Exited check box. To have it shut down automatically when all components have shut down, clear the check box.
5. Click OK.

6.5. Configuring the RTI's Crash Behavior

If an RTI component crashes, the Minidumper can provide some information that MAK can use to analyze the causes of the crash. You can configure whether or not this information is saved. The options are different on Windows and Linux. The options on Windows are:

- No dump. A standard Windows crash message is displayed.
- Minidump. The RTI prompts you to save a dump file.
- Minidump No Prompt. The RTI saves a dump file without prompting you.
- Silent exit. The RTI exits without showing a crash message. No dump file is saved.

On Linux, if you select the Minidump option, the RTI prints out a stack trace. The other options have no effect.

To configure an RTI component's crash behavior:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens (Figure 6-4).
3. Select the RTI Settings page.
4. Select the settings configuration that you want to edit.
5. Click the Edit button (✎). The RTI Settings Configuration dialog box opens. It has a set of pages with configuration settings.
6. Select the Diagnostic Output page (Figure 6-6).

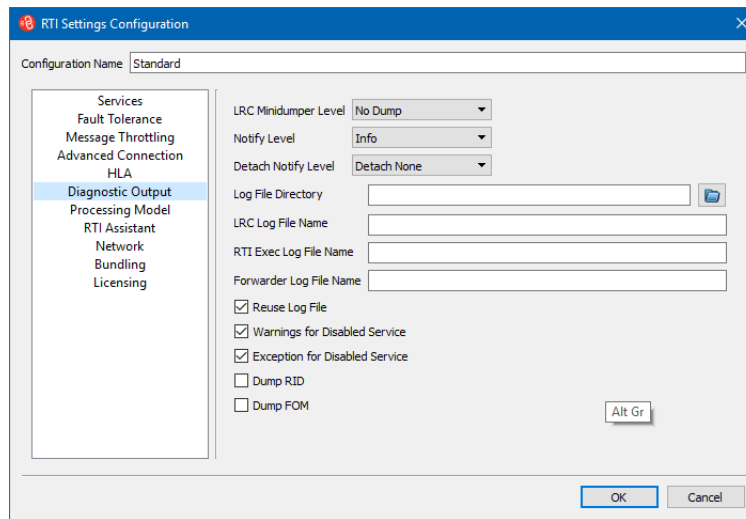


Figure 6-6. Diagnostic Output page

7. Select an option from the LRC Minidumper Level list.
8. Click OK.
9. Use the RTI Settings configuration that you edited the next time you start the RTI component.

6.6. Connecting to Remote Assistants

The RTI Assistant identifies other RTI Assistants on a LAN. It connects to them automatically. By connecting to remote Assistants, you can:

- ♦ View remote RTI components in the Federations View as though they were on the LAN.
- ♦ Start remote RTI components such as the rtiexec and RTI Forwarder.
- ♦ Connect RTI Forwarders to each other.

You can configure Assistant Connections for your LAN and for WANs. You can add connections, remove them, and connect to them. LAN connections identify the host and IP address. WAN connections identify the IP address, port, and either UDP or TCP.

6.6.1. Adding an Assistant Connection

The RTI Assistant automatically adds connections for RTI Assistants that it finds on the network. If you want to connect to an RTI Assistant on a WAN, you must add the connection manually.

For any given IP address and port, you can create two connection, one for UDP and one for TCP. Only one of these connections can be active at a time.

To add an Assistant Connection:

1. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Remote Assistants page.

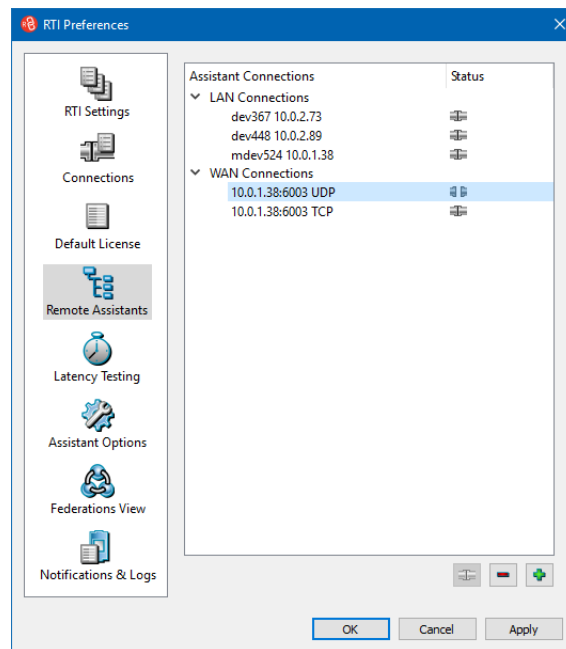


Figure 6-7. Remote Assistants page

4. To add a connection on your LAN, select LAN Connections. To add a WAN connection, select WAN Connections.
5. Click the Add button (+). The Assistant Connection dialog box opens. (You can also right-click the LAN Connection or WAN Connection lines and choose a menu option.)
6. Type an IP address or a hostname.
7. In the Protocol list, select UDP or TCP.
8. In the Port box, specify a port.
9. Click OK. The connection is added to the list.

6.6.2. Removing an Assistant Connection

To remove an Assistant Connection:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Remote Assistants page.
4. Right-click the Assistant Connection you want to remove. A menu is displayed.
5. Choose **Remove Assistant Connection**.

6.6.3. Connecting to or Disconnecting from an Assistant Connection

If you have two WAN connections for a particular IP address/port combination, only one can be connected at a time. If one of the connections is active, the Connect menu option and button are disabled for the other.

To connect to or disconnect from an Assistant Connection:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Remote Assistants page.
4. Right-click the Assistant Connection you want to connect to or disconnect from. A menu is displayed.
5. Choose **Connect** or **Disconnect**.

6.7. Shutting Down the RTI Assistant

If you do not configure the RTI Assistant to shut down automatically, you can shut it down manually. When you shut down the RTI Assistant, it forces all LRCs and rtiexecs on the local computer to shut down. It does not affect RTI components on remote computers.

To shut down the RTI Assistant:

1. In the system tray, right-click the RTI Assistant icon (🔌). A menu is displayed.
2. Choose **Shut Down All**.



7. Configuring the MAK RTI Startup Environment

This chapter explains how to use the RTI Chooser to configure the MAK RTI version and RTI settings to use, and how to configure RTI connections in the Choose RTI Connection dialog boxes and the RTI Assistant.

Configuring Your System for the MAK RTI	7-2
MAK RTI Environment Variables	7-3
Specifying RTI Settings Programmatically in HLA 1516.....	7-3
Configuring the RTI Version and Settings	7-4
Adding an RTI Configuration.....	7-6
Editing an RTI Configuration	7-7
Managing Connection Configurations.....	7-8
Adding rtiexec Connection Configurations.....	7-10
Adding RTI Connection Configurations	7-12
Editing Connections.....	7-14
Viewing Connection Status	7-15
Removing Connection Configurations	7-15
Default Port Values.....	7-16

7.1. Configuring Your System for the MAK RTI

When you start the rtiexec or a federate, the system determines which RTI to use and which settings to use based on several environment variables. If you have several RTIs installed or several versions of the MAK RTI, switching between them can be a cumbersome process of editing environment variables or maintaining batch files or shell scripts. On Windows, you can use the RTI Chooser to simplify this process. For details, please see [“Configuring the RTI Version and Settings,”](#) on page 7-4.

The RTI dynamic library needs to be located in your dynamic library search path. This path is specified in an environment variable, as listed in [Table 7-1](#).

Table 7-1: Path environment variable

System	Environment Variable
Linux	LD_LIBRARY_PATH
Windows	PATH



If you have more than one RTI installed on your computer, federates will try to use the first RTI found in the path. If you want to use the MAK RTI or a particular version of the MAK RTI, make sure that it is the first RTI in your path.

7.1.1. MAK RTI Environment Variables

The MAK RTI uses the following environment variables if they are set. Some of these may simplify federate configuration, and some may be required by older federates that rely on the environment settings used by variants of the DMSO RTI.

- ♦ RTI_RID_FILE – Specifies the absolute path to the RTI settings file including the name of the file (for example, *rid.mtl*).
- ♦ RTI_CONFIG – Specifies the absolute path to the directory containing the RTI configuration files, such as FED, FDD, routes configuration file and fixed grid DDM configuration. The MAK RTI automatically searches in this directory for these configuration files after looking in the current working directory.
- ♦ MAK_RTIDIR – Specifies the location of the MAK RTI binaries and libraries. It is set during installation (unless opted out) and when running the RTI Chooser. If modified outside of the RTI Chooser, it should be set to the full path of the MAK RTI. For example, on Windows, by default this is *C:\MAK\makRti4.5*. It is used to add the MAK RTI to search paths for libraries and executables. It is also used to build the RTI examples.
- ♦ RTI_HOME – This environment variable is not used by the MAK RTI, but some legacy federates may require it. RTI_HOME should be set to the full path to the RTI.
- ♦ RTI_BUILD_TYPE – This environment variable is not used by the MAK RTI, but some legacy federates may require it (these federates will also require RTI_HOME). RTI_BUILD_TYPE should be set to “.”. This variable was used by the DMSO RTI to indicate a platform type that was appended to RTI_HOME to determine the location of the RTI libraries.

The following environment variables are used by the RTI Assistant. For more information, please see Section 5.2, “[The RTI Assistant](#)”, in *MAK RTI Reference Manual*.

- ♦ RTI_ASSISTANT_PORT
- ♦ RTI_ASSISTANT_CLIENT_PORT
- ♦ RTI_ASSISTANT_ADDRESS
- ♦ RTI_ASSISTANT_DISABLE.

7.1.2. Specifying RTI Settings Programmatically in HLA 1516

You can specify RTI settings programmatically in HLA 1516. For details, please see Section 4.8, “[Specifying RTI Settings Programmatically in HLA 1516](#)”, in *MAK RTI Reference Manual*.

7.2. Configuring the RTI Version and Settings

If you have more than one version of the MAK RTI installed, you can use the RTI Chooser to quickly change which one to use without needing to manually edit your path or other environment variables. You can also specify which RTI settings to use with a particular MAK RTI version.



The RTI Chooser is only available on Windows.



When you change the RTI or RTI settings in the RTI Chooser, any federates on the computer are forcibly resigned and the rtiexec, RTI Forwarder, and RTI Assistant are shut down.

To configure the MAK RTI version and RTI settings to use:

1. On the Start menu, choose **MAK Technologies** → **MAK RTI 4.5** → **Configuration** → **RTI Chooser**. The RTI Chooser opens (Figure 7-1). The RTI Chooser window lists the known RTI configurations.

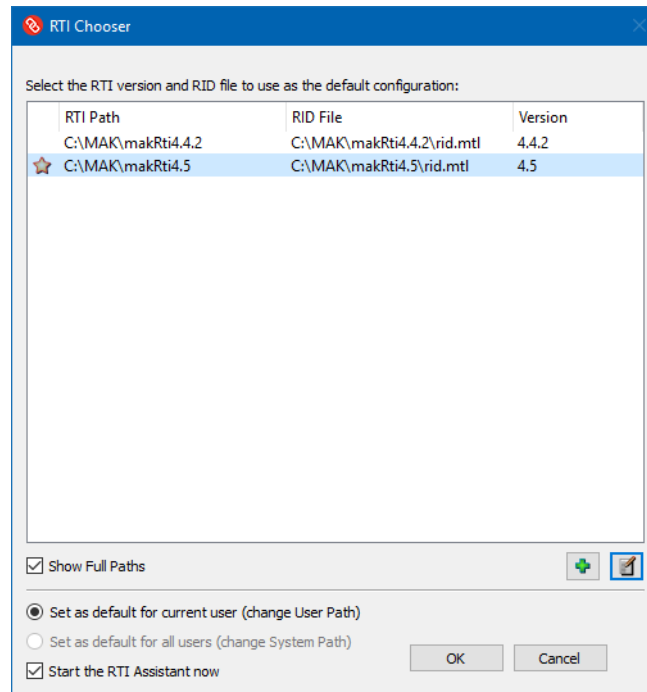


Figure 7-1. RTI Chooser

2. If you want to use an existing configuration, select it in the window. (You can add configurations. For details, please see [“Adding an RTI Configuration,”](#) on page 7-6.)
3. Specify whether you want to change the PATH environment variable for the current user (yourself), or for all users of this computer (system PATH).
4. If the RTI Assistant is not running and you want to start it, select Start the RTI Assistant Now.
5. Click OK. The PATH is updated. The next time you start the rtiexec or a federate, it will use the MAK RTI version and settings in the selected configuration.

7.2.1. Adding an RTI Configuration

You can add configurations to the RTI Chooser. Each configuration specifies a version of the MAK RTI and the RTI settings that you want to use with it.

To add a configuration to the RTI Chooser:

1. On the Start menu, choose **MAK Technologies** → **MAK RTI 4.5** → **Configuration** → **RTI Chooser**. The RTI Chooser opens (Figure 7-1).
2. Click the Add button (+). The Add RTI Configuration dialog box opens (Figure 7-2).

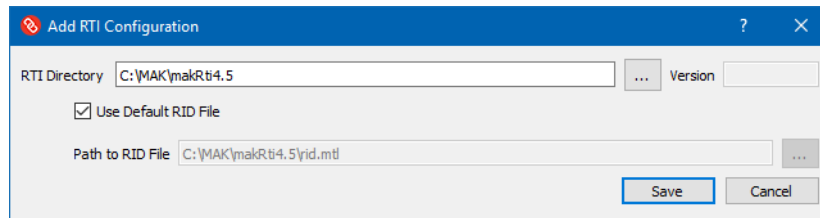


Figure 7-2. Add RTI Configuration dialog box

3. Type the path to the MAK RTI version that you want to configure, or click the browse button and select the installation directory for the MAK RTI version. The RTI Chooser automatically tries to detect the compiler version of the MAK RTI installation.
4. To use the default RID file (*rid.mtl*), select Use Default RID File. To specify a different RID file, type its path in the Path To RID File text box or select it using the chooser.
5. Click Save.

7.2.2. Editing an RTI Configuration

You can edit RTI configurations in the RTI Chooser.

To edit a configuration to the RTI Chooser:

1. On the Start menu, choose **MAK Technologies** → **MAK RTI 4.5** → **Configuration** → **RTI Chooser**. The RTI Chooser opens (Figure 7-1).
2. In the window, select the configuration you want to edit.
3. Click the Edit button (✎). The Edit RTI Configuration dialog box opens (Figure 7-3).

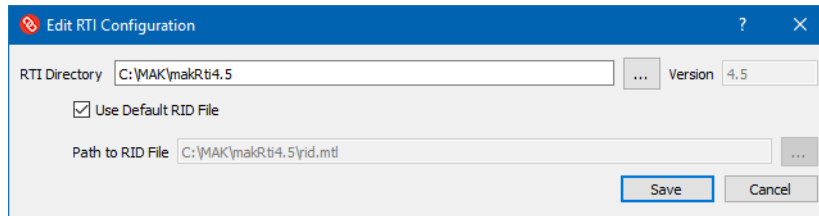


Figure 7-3. Edit RTI Configuration dialog box

4. Change the MAK RTI version or RID file.
5. Click Save.

7.3. Managing Connection Configurations

The RTI Assistant manages two types of connection configurations, rtiexec connections and RTI connections.



The MAK RTI supports IPv4 and IPv6, but you cannot mix the two protocols. If your federation is using IPv4, then all IP addresses in the RTI settings and RTI Assistant must be in the IPv4 format. The same applies to use of IPv6.

- An rtiexec connection configuration specifies the port and multicast or broadcast address an rtiexec should use to connect to the network. Optionally, it can specify that the rtiexec run in full compliance mode. The MAK RTI installs a predefined connection, which uses port 4001 and the address 229.7.7.7, running in full compliance mode.

You can create and remove rtiexec connection configurations in the Choose Connection Configuration dialog box and in the RTI Preferences dialog box, Connections page. You can also set a default connection.

- An RTI connection specifies the port and multicast or broadcast address that a federate should use when it connects to the network. This might be the same configuration that an rtiexec is using or it could be a configuration for running in lightweight mode (no rtiexec).

You can create and remove RTI connection configurations in the Choose RTI Connection dialog box and in the RTI Preferences dialog box, Connections page. You can also set a default connection.

[Table 7-2](#) describes the parameters in a connection. [Table 7-3](#) lists parameter combinations that must be unique.

Table 7-2: Connection parameters

Parameter	Connection	Default	Description
Hostname or IP address	rtiexec	Local computer	IP address or hostname of rtiexec computer.
Reliable Port	rtiexec	4001	Port used for reliable communications.
Best Effort Port	rtiexec	rtiexec: 4001	Port used for best effort communications.
	lightweight	lightweight: 4000	
Best Effort Address	rtiexec	229.7.7.7	Multicast address used for best effort traffic.
	lightweight		

Table 7-2: Connection parameters

Parameter	Connection	Default	Description
Network Interface Address	rtiexec lightweight	First device found	Local network interface on which RTI traffic will be sent and received.
Forwarder Port	rtiexec	5000	Port RTI Forwarders use to communicate with each other.

Table 7-3: Connection parameter uniqueness requirements

Parameter combinations that must be unique	Connection	Reason
Hostname/IP,Reliable Port,Interface	rtiexec	Two rtiexecs cannot be bound to the same port on the same machine.
Hostname/IP,Forwarder Port	rtiexec	Two rtiexecs cannot be bound to the same port on the same machine.
Best Effort Port,Best Effort Address,Interface	rtiexec lightweight	Two connections using the same best effort address and port will overlap traffic.

Connection configurations managed from the RTI Assistant override command-line options and RTI settings file parameters unless you force the MAK RTI to use them. The following settings are affected as indicated:

- ♦ `RTI_forceFullCompliance`, `RTI_useRtiExec`, `RTI_udpPort`, `RTI_destAddrString`, `RTI_distributedForwarderPort`, `RTI_networkInterfaceAddr`, `RTI_tcpPort`, and `RTI_tcpForwarderAddr` are all set based on the selected configuration.
- ♦ Send Internal RTI Messages Reliable (TCP) is enabled.
- ♦ FOM Data Transport Type is set to “Specified by FOM” and `RTI_mcastDiscoveryEnabled` is always forced to 0.
- ♦ The `RTI_forwarderRoutesFile` parameter is ignored. If you want to use a routes file, you must use the configuration in the RTI settings file, not the RTI Assistant connection configurations.
- To force RTI Forwarders, rtiexecs, and federates to use the values specified for these parameters in the RTI settings file instead of those in the connection configuration, set `RTI_configureConnectionWithRid` to 1.
- To force the rtiexec or RTI Forwarder to use the connection values specified by command-line options instead of those in the connection configuration, use the `--manual` command-line option.

7.3.1. Adding rtiexec Connection Configurations

You can add rtiexec connection configurations for your computer in the Choose Connection Configuration dialog box and in the RTI Preferences dialog box, Connections page. If you have specified that the rtiexec always try to use a selected connection, the Choose Connection Configuration dialog box does not open when the rtiexec starts up. In that case you would have to make changes in the RTI Preferences dialog box. (For details, please see [“Adding RTI Connection Configurations,”](#) on page 7-12.)

To add an rtiexec connection configuration in the Choose Connection Configuration dialog box:

1. Start the rtiexec. The Choose Connection Configuration dialog box opens ([Figure 7-4](#)).

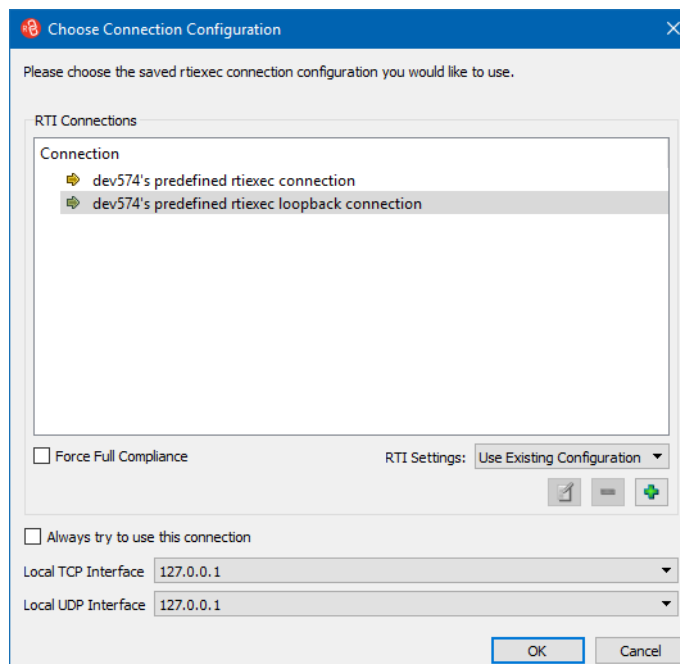


Figure 7-4. Choose Connection Configuration dialog box

2. Click the Add button (+). The Add New RTI Connection dialog box opens ([Figure 7-5](#)). The dialog box is populated with default parameters and a connection name built from those parameters.
3. Optionally, change the connection name.



The name can be seen everywhere on the network when the connection becomes active. You can change the name after the connection is created, but only on the machine on which the connection was created, and only when it is not being used.

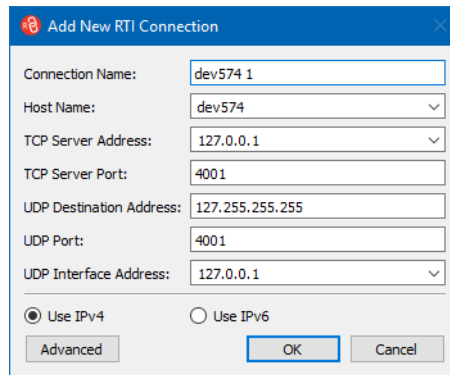


Figure 7-5. Add New Local rtiexec Connection dialog box



You can only add rtiexec connection configurations for the local host.

4. Type the ports you want to use in the TCP Server Port (reliable connection) and UDP Port (best effort connection) text boxes.
5. Type the best effort address that you want to use in the UDP Destination Address text box. It should be a multicast or broadcast address.
6. If your computer has multiple network cards, select the UDP interface address that you want to use.
7. Optionally, click Advanced and configure connections for distributed forwarding. For details, please see Section 10.6.1, “[Configuring RTI Forwarders Using the RTI Assistant](#)”, in *MAK RTI Reference Manual*.
8. Click OK.

7.3.2. Adding RTI Connection Configurations

You can add RTI connection configurations on the RTI Preferences dialog box, Connections page and in the Choose RTI Connection dialog box. The MAK RTI installs a predefined lightweight connection, which uses port 4000 and the address 229.7.7.7.

To add a connection configuration on the RTI Preferences, Connections page, or the Choose RTI Connection dialog box:

1. To open the Choose RTI Connection dialog box (Figure 7-6), start a federate.

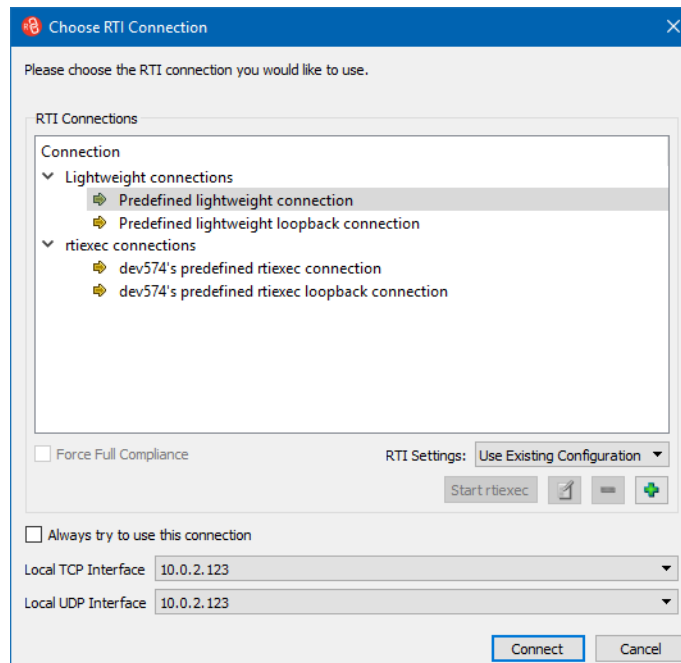


Figure 7-6. Choose RTI Connection dialog box

To open the RTI Preferences, Connections page:

- a. In the system tray, right-click the RTI Assistant icon (🔴). A menu is displayed.
- b. Choose **Preferences**. The RTI Preferences dialog box opens.
- c. Select the Connections page (Figure 7-7).



You can also open the RTI Preferences dialog box to the Connections page from the Start menu by choosing **MAK Technologies** → **MAK RTI 4.5** → **Configuration** → **Configure Connections**.

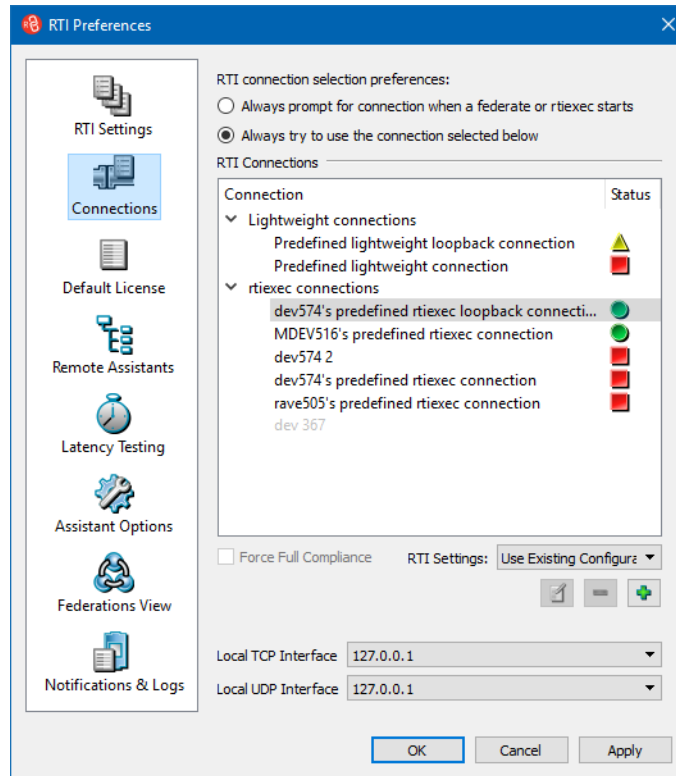


Figure 7-7. RTI Preferences dialog box, Connections page

2. Click the Add button (+). The Add New RTI Connection dialog box opens (Figure 7-8). The dialog box is populated with default parameters and a connection name built from those parameters.

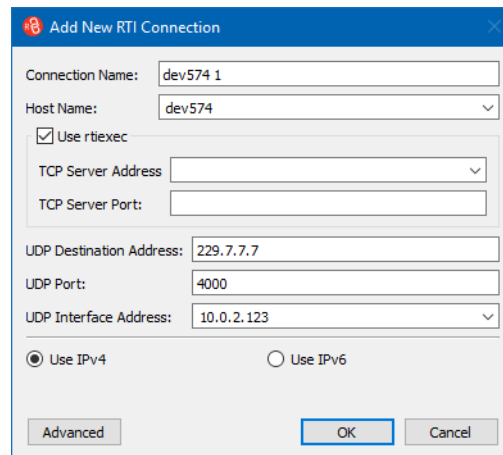


Figure 7-8. Add RTI Connection dialog box

3. Optionally, change the connection name.



The name can be seen everywhere on the network when the connection becomes active. You can change the name after the connection is created, but only on the machine on which the connection was created, and only when it is not being used.

4. If you want to add an rtiexec connection configuration:
 - a. Select the Use rtiexec check box.
 - b. Type the hostname or IP Address of the computer on which the rtiexec will run.
5. Type values in the TCP Server Port and UDP Port text boxes.
6. Type the best effort address in the UDP Destination Address text box. This should be a multicast or broadcast address.
7. If you have multiple network cards on the computer, type the device address of the one you want to use in the UDP Interface Address text box.
8. Optionally, click Advanced and configure connections for distributed forwarding. For details, please see Section 10.6.1, “[Configuring RTI Forwarders Using the RTI Assistant](#)”, in *MAK RTI Reference Manual*.
9. Click OK. If you selected Use rtiexec, the configuration is added to the list of rtiexec connections in the RTI Connections window. If you did not select Use rtiexec, the configuration is added to the Lightweight connections list.
10. In the RTI Settings list, select the RTI settings configuration that you want to use.
11. If you want to force full compliance, select the Force Full Compliance check box.

7.3.3. Editing Connections

Connections can only be edited on the machine on which they were originally defined. When a connection is used after being edited, other RTI Assistants on the network are updated to be aware of the changed parameters. You cannot edit predefined connections or connections that are in use.

To edit a connection:

1. In the Choose Connection Configuration dialog box ([Figure 7-4](#)), Choose RTI Connection dialog box ([Figure 7-6](#)), or RTI Preferences dialog box, Connections page ([Figure 7-7](#)), select the connection that you want to edit. If the connection can be edited (it was created on this machine and it is not in use) the edit icon is activated.
2. Click the Edit icon. The Edit RTI Connection dialog box ([Figure 7-8](#)), or Edit Local rtiexec Connection dialog box ([Figure 7-5](#)) opens.
3. Change the connection parameters as desired.
4. Click OK.

7.3.4. Viewing Connection Status

The Connections page on the RTI Preferences dialog box ([Figure 7-7](#)) displays a status icon for each connection. Status icons have the following meanings:

- ♦ Green circle. This connection is *verified*. For a lightweight connection, this means another federate is actively using it. For rtiexec connections this means there is a known rtiexec and RTI Forwarder using this connection.
- ♦ Green half circle. This connection is *forwarder-verified*. It has an RTI Forwarder using it, but no rtiexec.
- ♦ Yellow triangle. The RTI Assistant knows of no rtiexecs, forwarders, or federates using this connection. Therefore it cannot determine if the connection is active.
- ♦ Red square. This connection conflicts with another active (verified or forwarder-verified) connection.

In addition to the information conveyed by status icons, the RTI Assistant provides detailed information about each connection in tooltips. All parameters are listed as well as the current status, owning machine (the machine that defined the connection and has permission to edit it), and reason for conflict if it conflicts with another connection.

7.3.5. Removing Connection Configurations

To remove a connection configuration:

1. In the Choose Connection Configuration dialog box, the Choose RTI Connection dialog box, or on the Connections page of the RTI Preferences dialog box, select the configuration that you want to remove.

i

- ♦ You cannot remove a connection that is being used.
 - ♦ You cannot remove the predefined connections.
-

2. Click the Delete button (-).
3. Click OK.

7.4. Default Port Values

Table 7-4 lists the default ports for the various MAK RTI components.

Table 7-4: Default port values

Port	Use	Required	Asst‡	RTI Setting
4000	UDP Port	Yes	Yes	RTI_udpPort
4000	TCP Port	No	Yes	RTI_tcpPort
6001	Multicast Discovery Port	No	No	RTI_mcastDiscoveryPort
	Notes:	Only needed if multicast discovery is on.		
5000	RTI Forwarder Port	No	Yes	RTI_distributedForwarderPort
	Notes:	The port used by RTI Forwarders to communicate with each other.		
6003	RTI Assistant Port	No	No	RTI_ASSISTANT_PORT (environment variable)
	Notes:	The port used by RTI Assistants to communicate with each other.		

‡ Overridden by RTI Assistant



8. Licensing Issues

This chapter explains how to handle problems checking out a license.

Configuring a Federate to Let the rtiexec Check Out a License	8-2
Unlicensed Federations.....	8-3
Troubleshooting License Problems.....	8-4
Specifying a Default License Server	8-5
Displaying Default License Server Settings.....	8-5
Clearing Default License Server Settings	8-6

8.1. Configuring a Federate to Let the rtiexec Check Out a License

You can configure the rtiexec to check out licenses for federates. When rtiexec license checkout is enabled, federates do not have to configure license management. For details, please see [“Managing Licenses for the MAK RTI,”](#) on page 2-5.



If, as a result of relying on rtiexec license checkout, you do not configure license management on the PC on which a federate is running, or do not have access to a license server, you cannot run the federate in lightweight mode (except in unlicensed mode).

8.2. Unlicensed Federations

A federate can use the MAK RTI without a license; however, only two federates can participate in a federation. Unlicensed mode has the following conditions and restrictions:

When using an rtiexec:

- ♦ A federation execution will not allow a third unlicensed federate to join.
- ♦ Licensed and unlicensed federates cannot interoperate.
 - If a licensed federate has joined a federation execution, then an unlicensed federate will not be allowed to join even if it would only be the second federate.
 - If an unlicensed federate has joined a federation execution, then a licensed federate will not be allowed to join.
- ♦ If a federate resigns from a two federate federation, another federate can take its place.

If you are using lightweight mode (no rtiexec):

- ♦ A third unlicensed federate that attempts to join will be forcibly resigned by the MAK RTI. Any subsequent call to services that require the federate to be joined will result in a `FederateNotExecutionMember` exception.
- ♦ If unlicensed and licensed federates are running simultaneously, they will not exchange any messages even if they join the same federation execution. Other than a warning message, unlicensed federates behave as if licensed federates do not exist and vice versa.

You can disable unlicensed mode (even if the Force Unlicensed for Two is enabled). When unlicensed mode is disabled, if a license is not available, the MAK RTI exits with an error indicating that unlicensed mode has been disabled. Default: 0 (unlicensed mode is enabled).

To disable unlicensed mode:

1. In the system tray, right-click the RTI Assistant icon (🔧). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the RTI Settings page. The RTI Settings Configuration dialog box opens.
4. Select the settings configuration that you want to edit.
5. Click the Edit button (✎). The RTI Settings Configuration dialog box opens. It has a set of pages with configuration settings.
6. Select the Licensing page ([Figure 2-2](#)).
7. Select the Disable Unlicensed for Two check box.
8. Click OK.

8.3. Troubleshooting License Problems

To run the licensed MAK RTI, you must have a license for each LRC and a license server must be running. When you try to start the rtiexec or a federate, if the RTI Assistant detects a license problem, it displays the License Not Found dialog box to help you resolve the issue (Figure 8-1).

If the RTI Assistant is disabled, Disable Unlicensed for Two is disabled, and the MAK RTI cannot check out a license, the federate automatically runs in unlicensed mode.

The License Not Found dialog box lets you choose to run in unlicensed mode. However, if you need to run in licensed mode and there are no license servers running, none of the options in the dialog box will help you. You must start a license server. Then start the rtiexec (if you are using it) and your federates.

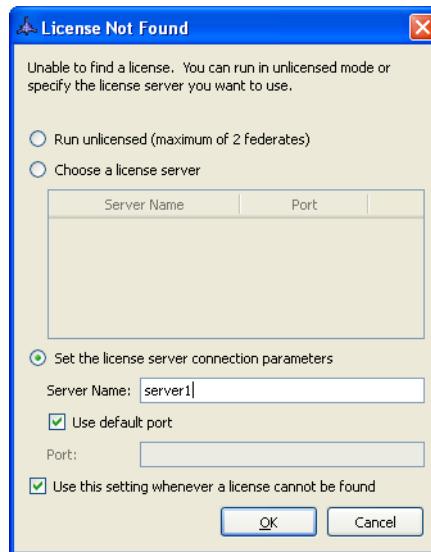


Figure 8-1. License Not Found dialog box

To resolve license issues:

1. In the License Not Found dialog box, select one of the following options:
 - Run in unlicensed mode.
 - Select a known license server in the list.
 - Specify the license server to use. Enter a server name, and optionally, a port.



For the second two options to work, a license server must be running. If there is no server running, you must start one before you can run in licensed mode.

2. Click OK.

8.3.1. Specifying a Default License Server

If the MAK RTI cannot connect to a license server and displays the License Not Found dialog box, you can save the server name and port to the RTI Assistant. Thereafter, if the MAK RTI cannot find a server, the RTI Assistant will try to connect to the default server.

To save default license server data:

1. In the License Not Found dialog box, select the Use This Setting Whenever a License Cannot Be Found check box.
2. Click OK.

8.3.2. Displaying Default License Server Settings

To display the license server default settings that have been saved in the License Not Found dialog box:

1. In the system tray, right-click the RTI Assistant icon (🔧). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Default License page. The license server settings are displayed (Figure 8-2).

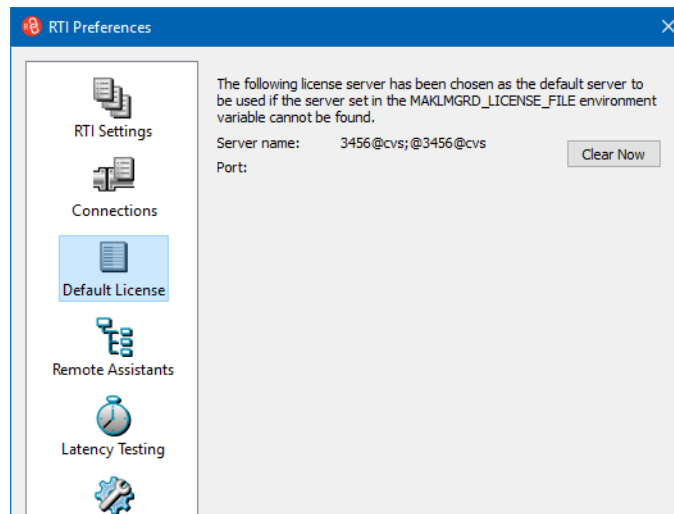


Figure 8-2. Default license server

8.3.3. Clearing Default License Server Settings

To clear the default license server settings:

1. In the system tray, right-click the RTI Assistant icon (🔧). A menu is displayed.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Default License page. The license server settings are displayed (Figure 8-2).
4. Click Clear Now.
5. Click OK.



MAK Products Glossary

This glossary defines terms used in MAK product documentation.

absolute dead reckoning

A method of [dead-reckoning](#) in which an application uses the time stamps contained within state updates if the sender is using absolute time stamping. Contrast with [relative dead reckoning](#).

aggregate

The combination of individual entities, aggregates, or both into a single object. For example, an organizational group such as a platoon.

API

Application Programming Interface.

articulated part

A part of an entity that is capable of movement relative to the entity, such as a turret or gun.

attached part

An object that is attached to another object, such as a rocket attached to a jet.

body coordinate frame

A coordinate framework centered on an entity. To represent the orientation of an entity, an application uses Euler angles or a rotation matrix that rotates from a reference frame to the body coordinate frame.

channel

A channel renders the view of an observer in a 3D visualization application.

clipping

A feature that prevents the display of terrain and objects or parts of objects, that are closer than or farther than specific distances from the observer.

clipping planes

The range of distances from the observer (near clipping and far clipping) in which an application clips objects.

culling

The process of discarding database objects which are not within view, and therefore need not be rendered and displayed.

Cartesian coordinates

A system for indicating location by means of three planes intersecting at right angles to one another at the origin.

Data Distribution Management

The set of HLA services used to specify the routing of data between publishers and subscribers. DDM adds greater control over the Declaration Management (DM) services that only specify filtering based on the class of the data (e.g., ground vehicle versus aircraft). DDM uses the data content to express regions of interest and regions of affect whose overlap establish a communication channel. An example is geographic filtering where a publisher indicates a region around the entity or effect and a subscriber indicates a region expressing its sensor range.

dead-reckoning

A process by which an application calculates the expected location of an entity during periods between state updates, based on velocity, acceleration, and rotational velocity.

DDM

[Data Distribution Management.](#)

Defense Modeling and Simulation Office

The former name of the executive secretariat for the Executive Council on Modeling and Simulation (EXCIMS), which provided a full-time focal point for information concerning Department of Defense modeling and simulation (M&S) activities. Renamed Modeling and Simulation Coordination Office. Currently the MSCO promulgates M&S policy, initiatives, and guidance to promote cooperation among DoD components to maximize efficiency and effectiveness.

DIS

[Distributed Interactive Simulation.](#)

DIS data packets

See [Protocol Data Unit](#).

display engine

An object in an application that can render 3D data.

Distributed Interactive Simulation

The DIS protocol is a set of standards that govern how participating applications share information about the virtual world. The protocol specifies a set of packets, called protocol data units (PDUs), that communicate this information. Each PDU identifies the sender and contains other information depending on the PDU type. The DIS protocol also specifies when and how frequently PDUs are sent.

The DIS protocol has been superseded by the [High-Level Architecture](#) for use by the Department of Defense.

DMSO

See [Defense Modeling and Simulation Office](#).

element definition

In VR-Forces and VR-Vantage, an element definition specifies all of the visualizers that control the various aspects of visualizing a scene element. For example, an element definition includes the main model or military symbology, trailing effects, cockpit displays, and so on.

emitter

A simulated device on an entity that emits electromagnetic radiation, for example, radar.

entity

An element in a simulation, such as a vehicle or a person, that is represented in the simulation through the issuance of state messages.

entity maintenance

A process by which the MAK Data Logger compensates for discontinuities following a time jump. The Logger sends out interim state messages for entities that were present before the time jump so that they do not time out before the next update message arrives.

entity-type

A list of seven components as specified by the DIS protocol and the HLA RPR FOM. If none of the seven components contains a wildcard (-1) value, then the entity type refers to a specific, narrowly-defined entity. If some components contain a wildcard, then the entity type refers to a class of entities. A larger number of wildcards indicates a broader, more general class.

The components of an entity-type are:

- ♦ Entity kind
- ♦ Domain
- ♦ Country
- ♦ Category
- ♦ Subcategory
- ♦ Specific
- ♦ Extra.

environmental process

According to the IEEE 1278.1a specification (DIS), environmental process PDUs communicate simple environment variables, small scale environmental updates, and embedded processes.

Euler angles

A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x, y, and z). These angles specify successive rotations needed to transform from a reference coordinate system to the entity's body coordinate system.

event

An interaction between objects or between an object and the terrain, such as firing of a munition, or a collision of entities.

exercise

The DIS term for a one or more interacting simulation applications. Compare to [federation execution](#) in HLA.

exercise connection

The object (*DtExerciseConn*) through which a VR-Link application connects to the network. State messages are sent through the exercise connection and information about remote entities is received through the exercise connection. (All MAK products are VR-Link applications.)

exercise ID

A numeric identification for a DIS simulation exercise.

FED file

Federation Execution Data file. The Federation Execution Data is the subset of [FOM](#) data needed and read by the [RTI](#). VR-Link applications also read the FED file.

federate

A connection to the [RTI](#). Typically a single simulation application can be thought of as a federate.

federation

A group of HLA federates capable of playing in the same [federation execution](#).

Federation Object Model

Defines the data content of a federation execution.

federation execution

The federation execution represents the actual operation, over time, of a subset of the federates and the RTI initialization data taken from a particular federation. A federation execution is the logical equivalent of a DIS simulation exercise.

field of view

Controls the perspective of the observer.

A wide field of view creates an effect like that of a wide-angle camera lens. Objects appear smaller and farther away from the observer, since the observer coverage spans a wider area. Depth become exaggerated.

A narrow field of view creates an effect like that of a telephoto lens. Objects appear larger and closer to the observer, and the overall scene depth appears flattened. The distances between objects appears compressed.

filter range

A setting that prevents distant entities from being processed while allowing distant terrain to appear normally.

FOM

See [Federation Object Model](#).

FOM Module

In HLA Evolved, defines additional modular data content for a federation execution, usually extensions to an existing FOM. (Also supported by the HLA 1.3 and HLA 1516 versions of the MAK RTI. Per the HLA Evolved interface specification:

“A partial FOM (containing some or all OMT tables) that specifies a modular component of a FOM. A FOM module may contain classes not inherent to it but upon which the FOM module depends, i.e., superclasses to the modular components. These superclasses will be included in the FOM module either as complete or scaffolding definitions.”

frame rate

The rate at which the application displays updated images.

ground clamping

A process by which the a 3D visualization application keeps an entity anchored to the surface of its terrain database, regardless of the altitude data contained in its entity state message.

geocentric coordinates

A coordinate system calculated with respect to the earth's center. The origin of the geocentric coordinate system is the center of the earth. The positive X-axis passes through the prime meridian at the equator; the positive Y-axis passes through 90 degrees east longitude at the equator; and the positive Z-axis passes through the north pole.

geodetic coordinates

A coordinate system in which position is determined relative to a reference ellipsoid, such as the surface of the earth at sea level. In MAK applications, geodetic coordinates consist of latitude and longitude in radians, and altitude in meters above the reference ellipsoid.

gridded data

Data that has been processed in a rectangular array of points, in X, Y or latitude/longitude, at which single data values define a two dimensional function. According to the IEEE 1278.1a specification, gridded data transmits information about large-scale or high-fidelity spatially and temporally varying ambient fields and about environmental processes and features.

guise

An alternative entity type used to display an object depending on the force ID. For example, a tank could look like an M1A1 to friendly forces and a T72 to hostile forces.

head-up display

A set of indicators and readouts superimposed onto a graphics display. Also called an overlay.

heartbeat

In DIS, the frequency with which current PDUs are sent to the network regardless of whether or not the entity's state has changed.

High-Level Architecture

The High Level Architecture (HLA) for simulations is a U. S. Department of Defense (DOD)-wide initiative to provide an architecture to support interoperability and reuse of simulations. The HLA supersedes DIS for the DOD.

HLA

See [High-Level Architecture](#).

interaction

A [message](#) describing a simulation event. An interaction describes an event, it does not update an object's state.

Logger control PDUs

A set of DIS PDUs or HLA interactions that let you control the MAK Logger remotely.

logical range

A suite of TENA Resources, sharing a common object model, that work together for a given range event. Similar in concept to the HLA Federation.

Local RTI Component

The libraries on a computer that implement an RTI. A federate communicates with the HLA network through the LRC.

LRC

See [Local RTI Component](#).

MAK Technologies Lisp

An adaptation of the Lisp language used in configuration files for MAK products.

message

A general term used to refer to [DIS PDUs](#) and [HLA interactions](#) and state updates. In TENA, a message is similar to an HLA interaction.

MIM

The MOM & Initialization Module (MIM) allows for extensions to be made to the HLA standard MOM. It is a subset of the FOM that contains OMT tables that describe the HLA MOM. The MIM also contains additional predefined HLA constructs such as object and interaction roots, data types, transportation types and dimensions. HLA specifies a standard MIM that is incorporated into all FDDs automatically by the RTI. The standard MIM can be replaced with a user-supplied MIM containing the standard MIM plus extensions.

model definition

In VR-Forces and VR-Vantage, a model definition specifies the 3D model and other basic attributes of a model. It is referenced by an element definition.

Modeling and Simulation Coordination Office

The executive secretariat for the Executive Council on Modeling and Simulation (EXCIMS), which provided a full-time focal point for information concerning Department of Defense modeling and simulation (M&S) activities. The MSCO promulgates M&S policy, initiatives, and guidance to promote cooperation among DoD components to maximize efficiency and effectiveness. (Formerly DMSO)

MSCO

Modeling and Simulation Coordination Office. (Formerly DMSO.)

MTL

See [MAK Technologies Lisp](#).

orientation clamping

Adjusts the pitch and roll of an entity so that it appears properly seated when the terrain is inclined. For example, if a tank is moving horizontally across the face of a hill, orientation clamping prevents the tank from appearing level, and therefore, partially embedded into the hillside. Used with [ground clamping](#).

object

An element in a simulation that has persistence, as opposed to an interaction, which is a transient element.

object handle

An integer that an application uses to identify a particular object in RTI service calls. An object handle is meaningful only to a particular federate. The same object can be known to different federates by different object handles.

object name

A character string that can be used to identify an object. In HLA, the object name is known to the RTI, and the RTI provides functions to find out an object's name, given its handle, and vice versa. Object names can be chosen by applications that register the objects with the RTI, however if you do not want to choose names for objects, the RTI will assign names for you.

observer

In MAK 3D applications, the point of view into the scene. (Called the eyepoint in MAK Stealth 6.x and earlier.)

PDU

See [Protocol Data Unit](#).

Protocol Data Unit

A unit of data message (packet) that is passed on a network between DIS simulation applications.

radio transmitter

A simulated device on an entity, capable of transmitting radio communications.

radio receiver

A simulated device on an entity, capable of receiving radio communications.

Realtime Platform Reference FOM

An [HLA](#) reference [FOM](#) based on the [DIS](#) protocol.

recording

A Data Logger file that stores a history of the interactions in a simulation for playback.

reflected entity list

A list of entities simulated by remote applications (federates in HLA) and about which the local simulation has received information over the network.

reference FOM

A [FOM](#) designed to be used as a whole, or with modification, by a wide variety of similar [federation executions](#).

relative dead reckoning

A method of dead reckoning in which an application approximates the location of an object based on the local time of receipt of the last state update message.

remote display engine

A display engine running in an application that is separate from the master application, and which is controlled by the master application.

RID file

See [RTI Initialization Data](#).

RTI Initialization Data

The initialization data required by the RTI for operation.

Data required by an RTI during initialization, independent of the FOM being used. RID data is usually dependent on a specific implementation of the RTI.

RPR FOM

See [Realtime Platform Reference FOM](#).

RTI

See [Run-Time Infrastructure](#).

Run-Time Infrastructure

A library and other supporting software that implements the HLA interface specification. All federates communicate with one another in an HLA environment through RTI functions.

scene element

In VR-Vantage and VR-Forces, any object that is rendered in the scene, such as an entity model, interaction, or tactical graphic.

Simulation Interoperability Standards Organization

A group that seeks to promote modeling and simulation interoperability and reuse for the benefit of diverse M&S communities, including developers, procurers, and users, world-wide.

simulation time

In VR-Forces, simulation time is used in dead-reckoning of remote entities and thresholding of local entities. Typically, simulation time is set once during each iteration of the application's main simulation loop so that all entities are dead-reckoned based on the same value of current time.

SISO

See [Simulation Interoperability Standards Organization](#).

smooth period

The period of time over which [trajectory smoothing](#) takes place.

smoothing

A method of ensuring that transitions from an entity's dead-reckoned position to its actual position are not so abrupt as to be visually disconcerting.

state

The current status of an object, including location, direction of movement, extent of damage, and so on.

tape

An alternative term for a Logger recording. Not a physical magnetic tape.

terrain following

In a 3D visualization application, causes the observer (eyepoint) to maintain a constant distance above the terrain surface. The observer's height changes in tandem with the peaks and valleys as it passes over the geography.

timeout

The period of time in which an application continues to display an entity after that entity's update messages have stopped appearing on the network. Time-outs are usually not used in HLA because there is no set frequency (no heart-beat) for transmitting messages.

timescale

A factor by which time is accelerated or slowed during playback of a Data Logger recording.

topographic coordinates

A right-handed Cartesian coordinate system whose X-Y plane is tangent to the earth's surface at the origin, with the positive X axis pointing north, the positive Y axis pointing east, and the positive Z axis pointing down. There are an infinite number of topographic coordinate systems – one for each point on the earth's surface.

topographic coordinate frame

A coordinate frame in the context of the terrain.

trajectory smoothing

A method used by to smooth positional discontinuities that could occur when new state updates arrive.

UDP port

A network channel through which an application sends and receives data for DIS exercises.

Universal Transverse Mercator

In general, a non-Cartesian coordinate system in which the X, Y, and Z correspond to easting (nearly east), northing (nearly north), and height above an ellipsoid that approximates the surface of the earth.

UTM

See [Universal Transverse Mercator](#).

view control messages

A set of programmatic messages that let you control the view in VR-Vantage applications remotely.



Index

- autoExit command-line option 4-9
- destAddrString command-line option 4-9
- distributedForwarderPort command-line option 4-9
- doNotUseRoutesFile command-line option 4-9
- forceFullCompliance command-line option 4-9
- forwarderToConnectTo command-line option 4-9
- help command-line option 4-9
- ignore_rest command-line option 4-9
- manual command-line option 4-10, 7-9
- notifyLevel command-line option 4-10
- setLogFileName command-line option 4-9
- setNotifyQuiet command-line option 4-10
- setRidFileName command-line option 4-10
- setUdpPort command-line option 4-10
- tcpAddress command-line option 4-10
- tcpNetworkInterfaceAddr command-line option 4-9
- udpNetworkInterfaceAddr command-line option 4-10
- usePreconfiguredRid command-line option 4-10
- useReliable command-line option 4-10
- version command-line option 4-10
- i command-line option 4-9
- p command-line option 4-10
- t command-line option 4-10
- command-line option 4-9

A

- A command-line option 4-9
- Add RTI Configuration dialog box 7-6
- Add RTI Connection dialog box 7-13
- adding
 - configuration to RTI Chooser 7-6
 - remote assistant connection 6-10
 - RTI connection configuration 7-12
 - rtiexec connection configuration 7-10
- address, broadcast 4-7
- Assistant Connection dialog box 6-10
- Assistant Options page 6-7, 6-8

B

- broadcast address 4-7
- button, Start rtiexec 4-12

C

- checking out license 8-2
- Choose Connection Configuration dialog box 4-7, 7-8, 7-10
- Choose RTI Connection dialog box 4-3, 7-12
 - starting rtiexec from 4-11
- choosing, RTI connection for federate 4-2
- CLASSPATH 2-3
- clearing, default license server 8-6
- closing, RTI Assistant 6-11
- color, federation 6-4
- color coding, connection 4-3
- command-line option
 - 4-9

command-line option (continued)

- A 4-9
- autoExit 4-9
- D 4-9
- d 4-9
- destAddrString 4-9
- distributedForwarderPort 4-9
- doNotUseRoutesFile 4-9
- F 4-9
- f 4-9
- forceFullCompliance 4-9
- forwarderToConnectTo 4-9
- h 4-9
- help 4-9
- i 4-9
- ignore_rest 4-9
- K 4-9
- l 4-9
- M 4-10
- manual 4-10, 7-9
- N 4-10
- n 4-10
- notifyLevel 4-10
- overriding, connection options 4-7
- P 4-10
- p 4-10
- q 4-10
- R 4-10
- r 4-10
- rtiexec 4-9
- setLogFileName 4-9
- setNotifyQuiet 4-10
- setRidFileName 4-10
- setUdpPort 4-10
- t 4-10
- tcpAddress 4-10
- tcpNetworkInterfaceAddr 4-9
- udpNetworkInterfaceAddr 4-10
- usePreconfiguredRid 4-10
- useReliable 4-10
- v 4-10
- version 4-10

compliant mode, specifying 4-9

configuration

- adding to RTI Chooser 7-6
- connection 4-7
- editing in RTI Chooser 7-7
- file 3-5

configuring

- connection 4-7

configuring (continued)

- Minidumper 6-8
- RTI Assistant 6-6
 - startup 6-7
- RTI version and RID file 7-4
- rtiexec connection 7-8
 - system 7-2
- connecting, remote assistant 6-11
- connection
 - color coding 4-3
 - failed 4-6, 4-11
 - lightweight 4-4
 - rtiexec, configuring 7-8
- connection configuration 4-7
 - adding rtiexec 7-10
 - lightweight, adding 7-12
 - overriding with RID file 7-9
 - removing 7-15
 - rtiexec, adding 7-12
- Connection Failed dialog box 4-5, 4-6, 4-11
- Connections page 7-12
- conventions, manual xiii
- crash
 - dump file 6-8
 - RTI 6-8
- custom, logical time implementation 2-4

D

- D command-line option 4-9
- d command-line option 4-9
- data distribution management
 - See also* DDM
 - definition of 3-4
- DDM 3-4
- declaration management, definition of 3-3
- default
 - license, clearing 8-6
 - license server 8-5
- Default License page 8-5, 8-6
- Defense Modeling and Simulation Office
 - 1-2
- deleting
 - connection configuration 7-15
 - federate 4-15
- destroying, federation 5-2
- diagnostic
 - level 4-10
 - message 4-10

- dialog box
 - Add RTI Configuration 7-6
 - Add RTI Connection 7-13
 - Assistant Connection 6-10
 - Choose Connection Configuration 4-7, 7-8, 7-10
 - Choose RTI Connection 4-3, 7-12
 - Connection Failed 4-5, 4-6, 4-11
 - License Not Found 8-4, 8-5
 - License Setup 2-9
 - RTI Preferences 5-4, 6-6
 - Connections page 7-8
 - disabling, unlicensed mode 8-3
 - disconnecting federate, removing 5-4
 - disconnecting, remote assistant 6-11
 - distributed forwarder 1-3
 - DLL, search path 7-2
 - DM. *See* declaration management 3-3
 - DMSO RTI
 - compatibility with 7-3
 - RTI_BUILD_TYPE environment variable 7-3
 - RTI_HOME environment variable 7-3
 - dump file, crash 6-8
 - dynamic library, search path 7-2
 - dynamic linking 1-5
- E**
- editing, configuration in RTI Chooser 7-7
 - environment variable 7-2
 - MAK_RTIDIR 7-3
 - MAKLMGRD_LICENSE_FILE 2-10
 - PATH 7-5
 - path 7-2
 - RTI_ASSIS-TANT_PORT 7-16
 - RTI_BUILD_TYPE 7-3
 - RTI_CONFIG 7-3
 - RTI_HOME 7-3
 - RTI_RID_FILE 7-3
 - exiting
 - RTI Assistant 6-8, 6-11
 - rtiexec 4-9
- F**
- F command-line option 4-9
 - f command-line option 4-9
 - failed connection 4-6
 - FDD file 3-4
 - path to 7-3
 - FED file 3-4
 - path to 7-3
 - federate 6-3
 - choosing RTI connection 4-3
 - failed connections 4-11
 - icon, specifying 6-5
 - license 8-2
 - removing disconnected 5-4
 - resigning 4-15
 - shutting down 5-3
 - starting 4-2
 - synchronizing 3-3
 - troubleshooting connection problems 4-6
 - unlicensed 1-7, 8-3
 - federation 6-3
 - color 6-4
 - destroying 5-2
 - management 3-3
 - unlicensed 8-3
 - Federation Object Model. *See* FOM
 - Federations View 5-2, 6-3
 - federate, icon 6-5
 - federation, color 6-4
 - fedtime, installing Java 2-3
 - file, configuration 3-5
 - FLEXlm 2-5, 2-8
 - FOM 3-2, 3-4
 - FOM Document Data 3-4
 - forwarder, starting 4-14
 - free mode 1-7
- G**
- GUI, localizing 2-12
- H**
- h command-line option 4-9
 - High Level Architecture
 - See also* HLA
 - introduction 1-2, 3-2
 - HLA
 - Interface Specification 3-2
 - introduction 1-2, 3-2
 - specification 1-5
 - hostname, license server 2-9

I

- icon
 - federate, specifying 6-5
 - RTI Assistant 6-2
- IEEE 1516 1-5
- installation 2-1
- installing
 - HLA 1516 Java fedtime library 2-3
 - Java bindings 2-2
- IP address 4-9

J

- Java
 - binding, installing 2-2
 - fedtime library, installing 2-3
- jvm.dll* 2-3

K

- K command-line option 4-9

L

- l command-line option 4-9
- LAN, connecting to remote assistant 6-9
- latency testing 6-3
- LD_LIBRARY_PATH 2-3
- libjvm.so* 2-3
- library, RTI 1-4
- license
 - checking out 8-2
 - failure to find 8-4
 - server and port 8-4
 - unlicensed mode 1-7
- License Manager 2-5, 2-8
- License Not Found dialog box 8-4, 8-5
- license server
 - default 8-5
 - displaying default setting 8-5
 - hostname 2-9
- License Setup dialog box 2-9
- lightweight, connection 4-4
- lightweight mode 1-4
 - unlicensed federation 8-3
- link-compatibility 1-5
- linking, dynamic 1-5
- Linux, installing on 2-2

- Local Component Notification History
 - window 6-7
- Local RTI Component 1-2, 1-3, 1-4, 3-4
- localizing, GUI 2-12
- logical time
 - implementing, custom 2-4
- LogicalTimeFactory 2-4
- LRC 1-4, 1-6, 3-4
 - messages 6-7

M

- M command-line option 4-10
- MAK_RTIDIR environment variable 7-3
- MAKLMGRD_LICENSE_FILE,
 - environment variable 2-10
- manual, conventions xiii
- message
 - diagnostic 4-10
 - LRC 6-7
 - notification 6-7
 - reliable 4-10
- Minidumper, configuring 6-8
- Modeling and Simulation Coordination
 - Office 1-2
- multicast
 - address 4-2
 - discovery, port 7-16
- multicast group, address 4-9

N

- N command-line option 4-10
- n command-line option 4-10
- network, traffic 3-3
- notification level 4-10

O

- object management, definition of 3-3
- ownership management, definition of 3-3

P

- P command-line option 4-10
- parameter
 - RTI_configureConnectionWithRid 4-10, 7-9
 - RTI_destAddrString 4-9, 7-9
 - RTI_distributedForwarderPort 7-9, 7-16

- parameter (continued)
 - RTI_forceFullCompliance 7-9
 - RTI_forwarderRoutesFile 7-9
 - RTI_mcastDiscoveryEnabled 7-9
 - RTI_mcastDiscoveryPort 7-16
 - RTI_networkInterfaceAddr 7-9
 - RTI_tcpForwarderAddr 7-9
 - RTI_tcpPort 7-9, 7-16
 - RTI_udpPort 4-10, 7-9, 7-16
 - RTI_useRtiExec 7-9
- PATH 2-3
- path
 - environment variable 7-2
 - FED file 7-3
 - RID file 7-3
 - search 7-2
- PATH environment variable 7-5
- port 4-2, 4-7
 - default values 7-16
 - license server 8-4
 - RTI Assistant 7-16
 - RTI Forwarder 7-16
 - specifying 4-10
- product
 - technical support xii
 - upgrades xii
- Q**
 - q command-line option 4-10
 - Qt Linguist 2-12
- R**
 - R command-line option 4-10
 - r command-line option 4-10
 - reliable transport 4-10
 - remote assistant, connecting and disconnecting 6-11
 - remote rtiexec, starting 4-7
 - removing
 - connection configuration 7-15
 - federate 4-15
 - resigning, federate 4-15
 - RID file 3-5, 4-2
 - adding configuration to RTI Chooser 7-6
 - editing configuration in RTI Chooser 7-7
 - overriding, connection configuration 7-9
 - overriding connection parameters 4-7
 - RID file (continued)
 - selecting 7-2
 - specifying 7-4
 - path 7-3
 - rid.mtl* 3-5
 - routes file 4-9
 - RTI
 - crash 6-8
 - definition 1-2
 - executive 1-4
 - introduction 3-2
 - library 1-4
 - services 3-2
 - settings 3-5, 4-2
 - specification 1-5
 - specifying version to use 7-4
 - version, selecting 7-2
 - RTI Assistant 1-3, 1-6
 - adding connection configurations 7-12
 - clearing, default license 8-6
 - configuring 6-6
 - default license server 8-5
 - default license setting 2-6, 8-3, 8-5
 - exit policy 6-8
 - icon 6-2
 - port 7-16
 - remote connection, adding 6-10
 - shutting down 5-3, 6-11
 - starting
 - starting, RTI Assistant 6-2
 - startup, configuring 6-7
 - RTI Chooser 1-3, 7-2, 7-4
 - adding configuration 7-6
 - editing configuration 7-7
 - RTI connection, choosing for federate 4-2
 - RTI Executive 1-3
 - RTI Forwarder 1-6
 - port 7-16
 - starting 4-14
 - RTI Preferences, Connections page 7-12
 - RTI Preferences dialog box 5-4, 6-6
 - Connections page 7-8, 7-12
 - RTI_ASSIS-TANT_PORT environment variable 7-16
 - RTI_BUILD_TYPE environment variable 7-3
 - RTI_CONFIG environment variable 7-3
 - RTI_configureConnectionWithRid parameter 4-10, 7-9
 - RTI_destAddrString parameter 4-9, 7-9

- RTI_distributedForwarderPort parameter 7-9, 7-16
- RTI_forceFullCompliance parameter 7-9
- RTI_forwarderRoutesFile parameter 7-9
- RTI_HOME environment variable 7-3
- RTI_JAVA_TIME_CLASS 2-4
- RTI_JAVA_TIME_INTERVAL_CLASS 2-4
- RTI_mcastDiscoveryEnabled parameter 7-9
- RTI_mcastDiscoveryPort parameter 7-16
- RTI_networkInterfaceAddr parameter 7-9
- RTI_RID_FILE environment variable 7-3
- RTI_tcpForwarderAddr parameter 7-9
- RTI_tcpPort parameter 7-9, 7-16
- RTI_udpPort parameter 4-10, 7-9, 7-16
- RTI_useRtiExec parameter 7-9
- rtiexec 1-3, 1-4, 1-6
 - automatic exit 4-9
 - checking out license 8-2
 - command-line options 4-9
 - connection configuration, adding 7-10
 - destroying federation 5-2
 - notification messages 6-7
 - remove disconnected federate 5-4
 - resigning federate 4-15
 - shutting down 5-3
 - starting 4-7
 - remote computer 4-7
 - starting from Choose RTI Connection dialog box 4-11
- rtiexec connection 4-4
- RTIspy Diagnostic GUI 1-6
- RTIspy diagnostic GUI 6-3
- Run-Time Infrastructure, introduction 3-2

S

- search order 7-2
- selecting
 - RID file 7-2
 - rti connection for federate 4-2
 - RTI version 7-2
- service
 - data distribution management 3-4
 - declaration management 3-3
 - federation management 3-3
 - object management 3-3
 - ownership management 3-3
 - time management 3-3
- services, RTI 3-2

- settings, RTI 3-5, 4-2
- Shut Down All 5-3
- shutting down
 - RTI Assistant 6-11
 - RTI components 5-3
- SISO DLC HLA API 1516 1-5
- SISO-STD-004.1-2004 1-5
- specification, obtaining HLA and RTI 1-5
- specifying
 - compliant mode 4-9
 - path
 - FED and FDD files 7-3
 - RID file 7-3
 - port 4-10
 - port and broadcast address 4-7
 - RTI version and RID file 7-4
- stack trace 6-8
- Start rtiexec button 4-12
- starting
 - federate 4-2
 - RTI Forwarder 4-14
 - rtiexec 4-7
 - remote 4-7
- startup
 - RTI Assistant, configuring 6-7
- support, technical xii
- synchronizing, federates 3-3
- system tray 6-2

T

- technical support xii
- testing, latency 6-3
- time
 - logical, creating custom 2-4
- time management, definition of 3-3
- timestamp order 3-3
- translating, GUI text 2-12
- trial mode 1-7
- troubleshooting, license problems 8-4

U

- UDP port 7-16
- unlicensed, federation 8-3
- unlicensed mode 1-7, 8-4
 - disabling 8-3
- upgrades xii

V

- v command-line option 4-10
- variable, environment 7-2
- version, specifying RTI 7-4

W

- WAN, connecting to remote assistant 6-9
- WAN connection, connecting and disconnecting 6-11
- Windows, installing on 2-2



Link - Simulate - Visualize

RTI-4.5-3-180312