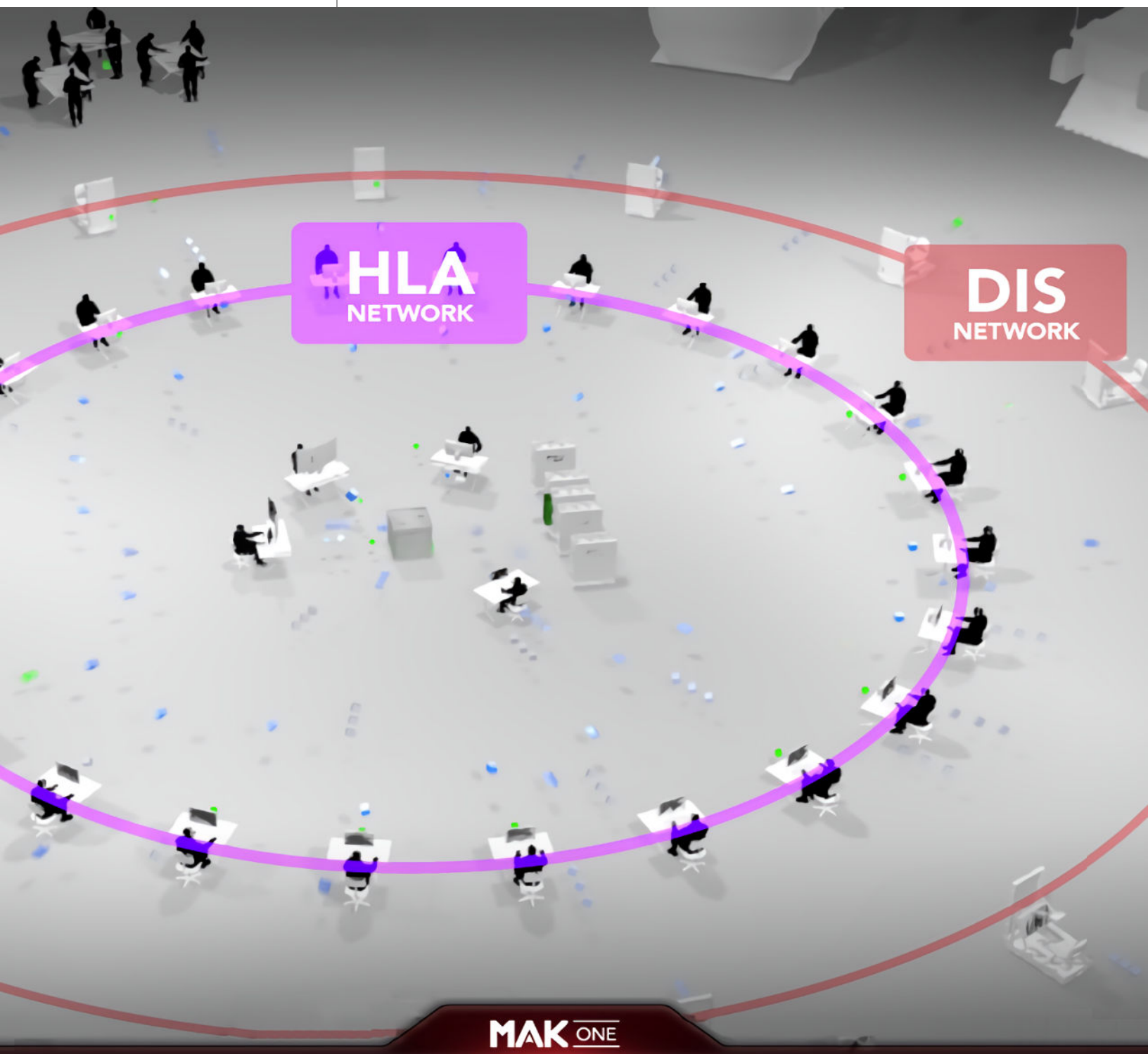




Copyright © 2025 MAK Technologies
All rights Reserved. Printed in the United States.

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

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

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.

RFView® is a registered trademark of Information Systems Laboratories (ISL), Inc.

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, visit the MAK ONE Product Support Knowledge Base (at <https://www.mak.com/knowledge>) and refer to the Third-Party/Open Source List under Frequently Asked Questions (FAQ).

MAK Technologies
10 Fawcett Street, Suite 204
Cambridge, MA 02138 USA

Voice: +1 617.876.8085

Fax: +1 617.876.9208

info@mak.com

www.mak.com

Revision RTI-5.0.1-02-251125



Contents

Preface

How the Manual Is Organized	8
Documentation Set	8
Try the Help	9
Links Between PDFs	9
MAK Products	9
How to Contact Us	11
Telephone	11
E-mail	12
Internet	12
Post	12
Document Conventions	12
Mouse Button Naming Conventions	13
Third-Party Licenses	13

1. Introduction to the MAK RTI

1.1. The MAK RTI	16
1.1.1 MAK RTI Components	17
1.2. The rtiexec is not the RTI	17
1.3. Lightweight Mode	18
1.4. Compatibility with Other RTI Implementations	18
1.4.1 Obtaining HLA RTI Specification Documents	19
1.5. The RTI Assistant	19
1.5.1 The RTIs Spy Diagnostic GUI	19
1.6. The Unlicensed Version of the MAK RTI	20

2. Installing the MAK RTI

2.1. Required Environment Variables	22
2.2. Installing MAK RTI on Windows	22
2.2.1 Default Windows Installation Directory	22
2.3. Installing MAK RTI on Linux Systems	22
2.4. Installing PDB Files for Debugging Purposes	23
2.4.1 Installing PDB Files on Windows	23
2.4.2 Reporting an Issue Using the .callstack.log and .dmp Files	24
2.4.3 Specifying the Location for .log and .dmp Files	24
2.5. Configuring MAK RTI with Environment Variables	24
2.6. Installing the MAK RTI Java Bindings	26
2.6.1 Linux Installation	26
2.6.2 Windows Installation	26
2.6.3 Installing the HLA 1516 Java Fedtime Library	26
2.7. Uninstalling from Windows	27
2.8. Uninstalling from Linux	28
2.9. Managing Licenses for the MAK RTI	28
2.10. Installing and Setting Up the MAK License Manager	29
2.10.1 Specifying the License Server	30
2.11. Localizing the Graphical User Interface	31
2.11.1 Translating Other Interface Files	34
2.11.2 Applying the Language Files	34

3. HLA Concepts

3.1. The High Level Architecture	36
3.1.1 The RTI	36
3.1.2 Federates and Federations	36
3.2. RTI Services	36
3.2.1 Federation Management	36
3.2.2 Declaration Management	37
3.2.3 Object Management	37
3.2.4 Ownership Management	37
3.2.5 Time Management	37
3.2.6 Data Distribution Management	37
3.2.7 Miscellaneous Support Services	38
3.3. FOMs and FED Files	38
3.4. RTI Settings and the RTI Initialization Data (RID) File	38

4. Connecting Federates to the MAK RTI

4.1. Connecting a Federate to the MAK RTI	40
4.1.1 Troubleshooting Federate Connection Problems	42

4.2. Starting the rtiexec	43
4.2.1 Starting the rtiexec when Starting a Federate	47
4.2.2 Starting the rtiexec from the Federations View	49
4.2.3 Shutting Down the rtiexec	49
4.3. Starting an RTI Forwarder from the Federations View	50
4.3.1 Shutting Down an RTI Forwarder	50
4.4. Resigning a Federate from a Federation	50
 5. Managing Federations	
5.1. Opening the Federations View	54
5.2. Destroying Federations	54
5.2.1 Shutting Down all Local RTI Components	54
5.3. Removing Disconnected Federates from the Display	55
 6. The RTI Assistant	
6.1. Starting the RTI Assistant	58
6.2. The RTI Assistant Windows	58
6.2.1 Federations View	58
6.2.2 RTI Preferences Dialog Box	60
6.2.3 Local Component Notification History Window	61
6.3. Configuring the RTI Assistant's Startup Behavior	61
6.4. Specifying the RTI Assistant's Exit Policy	62
6.5. Configuring the RTI's Crash Behavior	62
6.6. Connecting to Remote Assistants	63
6.6.1 Adding an Assistant Connection	63
6.6.2 Removing an Assistant Connection	64
6.6.3 Connecting to or Disconnecting from an Assistant Connection	65
6.7. Shutting Down the RTI Assistant	65
 7. Configuring the MAK RTI Startup Environment	
7.1. Configuring Your System to Use the MAK RTI	68
7.1.1 MAK RTI Environment Variables	68
7.1.2 Specifying RTI Settings Programmatically in HLA 1516	69
7.2. Configuring the MAK RTI Version and RTI Settings to Use	69
7.2.1 Adding an RTI Configuration	70
7.2.2 Editing an RTI Configuration	71
7.3. Managing Connection Configurations	71
7.3.1 Adding rtiexec Connection Configurations	73
7.3.2 Adding RTI Connection Configurations	75
7.3.3 Editing Connections	77
7.3.4 Viewing Connection Status	77

- 7.3.5 Removing Connection Configurations 78
 - 7.4. Default Port Values 78
- 8. Licensing Issues**
 - 8.1. Configuring a Federate To Allow the rtiexec Check Out a License 82
 - 8.2. Unlicensed Federations 82
 - 8.3. Troubleshooting License Problems 83
 - 8.3.1 Specifying a Default License Server 84
 - 8.3.2 Displaying Default License Server Settings 84
 - 8.3.3 Clearing Default License Server Settings 85
- MAK Products Glossary 87**
- Index 99**



Preface

This manual is for people 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 the *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, see the *MAK RTI Release Notes*.

How the Manual Is Organized

This manual is organized as follows:

"Chapter 1. Introduction to the MAK RTI" on page 15 describes the MAK RTI and its features.

"Chapter 2. Installing the MAK RTI" on page 21 describes installation and configuration tasks, including setting up the license server.

"Chapter 3. HLA Concepts" on page 35, 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" on page 39, 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" on page 53 describes some basic federation management procedures, primarily how to remove inactive federates and how to destroy a federation.

"Chapter 6. The RTI Assistant" on page 57 provides a high-level introduction to the RTI Assistant.

"Chapter 7. Configuring the MAK RTI Startup Environment" on page 67, 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" on page 81 explains how to let the MAK RTI check out licenses and how to troubleshoot license issues.

The "MAK Products Glossary" on page 87 defines terms common to HLA and real-time simulations.

Documentation Set

MAK RTI documentation is provided as manuals in PDF format, online help, and class documentation. Electronic versions of the documentation are in `./makRtix.x/doc`. On Windows, you can access the documentation from the MAK RTI folder on the Start menu. You can also find the most current documentation at <https://www.mak.com/mak-one/support/help/specs/mak-rti>. The MAK RTI documentation set includes:

- The *MAK RTI User 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.
- The *MAK RTI Reference Manual* provides detailed information about configuring the MAK RTI and designing network topologies for HLA federations.
- 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`.

- HTML5 help. The MAK RTI has HTML help, which you can access from the RTI Assistant system tray icon's context menu, the Federations View window, the Documentation shortcuts folder, or `./doc/help`. It has a natural language search and combines both the *MAK RTI User Guide* and the *MAK RTI Reference Manual*.
- The *MAK ONE Products Interoperability Guide* explains how MAK products work together and provides troubleshooting information.
- The *MAK RTI Release Notes* provides system information, new features and updates, a list of fixed bugs, and a list of known problems.

Try the Help

MAK's HTML5 help is the best way to use our documentation. It's easy to navigate, with improved natural language search tools and filters to focus your search. You can click the illustrations to see high-resolution enlargements that show more detail. It's built into the product and installed locally — simply press **F1** in the Federations View or choose Help from the RTI Assistant context menu to launch the help in your browser. You do not need to be connected to the internet to use it.

 If you attempt to view the help in a browser that does not fully support HTML5, you may experience issues with some features.

Links Between PDFs

Many MAK PDF documents link to each other. For example, the description of a new feature in the release notes might include links to detailed information in a user guide, and there are also many links from user guides to the *Adding Content to MAK ONE Applications Reference Manual* and back. These links between PDFs rely on two things:

- The linked PDF files must be in the same directory. MAK installers put the PDFs for each product in that product's *doc* directory — in Windows, `./makRti/doc`. If you move or copy a PDF away from that directory or download a new one to another location, any links between that PDF and the others stop working.
- You must use Acrobat Reader to read MAK PDFs. Links between PDFs do not work when you read with a web browser. Acrobat Reader is free; you can get it from Adobe at <https://get.adobe.com/reader>.

MAK Products

The MAK Technologies product line includes:

- 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 NETN FOM and 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, RTIsby®, 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 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 VR-Vantage and the VR-Vantage Toolkit.
 - VR-Vantage 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.
 - **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™.** VR-TheWorld 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™.** DI-Guy is an SDK that allows you to embed the DI-Guy library in your real-time application and populate your world with lifelike human characters. DI-Guy provides high-fidelity human characters that move realistically, respond to simple high-level commands, and travel throughout the environment as directed by the hosting visual application. It comes with thousands of ready-to-use characters, appearances, and motions. DI-Guy character simulation is an integral part of all MAK's visual applications, including VR-Engage, VR-Forces, and VR-Vantage.
- **RFView® for MAK ONE.** Synthetic Aperture Radar (SAR) imaging is an indispensable tool for today's modern militaries. Developed in partnership with ISL, RFView for MAK ONE acts as a SAR server and delivers simulated SAR images that are both physically accurate and fully correlated with out-the-window views and EO/IR images of MAK ONE synthetic environments. It includes a toolkit that enables you to build a client application to request SAR images from the server.
- **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, a sensor operator, 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.
- **MAK Legion.** MAK Legion is a collection of applications and an SDK designed to support simulations with millions of simulation objects.

How to Contact Us

For MAK 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
	Fax:	617-876-9208

E-mail

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

Internet

MAK website home page:	www.mak.com
License key requests:	https://www.mak.com/mak-one/support/portal
Technical support:	https://www.mak.com/mak-one/support/portal
Product version and platform information:	https://www.mak.com/mak-one-specs
For the free, unlicensed MAK RTI:	https://www.mak.com/mak-one/support/help/bonus-material

Post

Send postal correspondence to:	MAK Technologies 10 Fawcett Street, Suite 204 Cambridge, MA 02138 USA
--------------------------------	---

When requesting support, tell us the product you are using, the version, and the platform on which you are running. If you are using VR-Vantage, VR-Forces, VR-Engage, or DI-Guy, please also include your data compatibility version and date.

Document Conventions

This manual uses these typographic conventions:

Sans Serif Bold	Indicates a key on the keyboard or menu options. Mentions of letter keys do not require you to press the Shift key. If Shift is necessary, that will be included, for example, Shift+Q .
	Also indicates parameters.
Monospaced	Indicates values you enter such as commands or arguments.
<i>Monospaced</i>	Indicates command variables that you replace with appropriate values.
<i>Italic</i>	
{ }	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>--help</code> <code>-h</code>).
/	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames.
<i>Italic</i>	Indicates a file name, pathname, or a class name.
►	Indicates a 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 .



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 MAK RTI home directory. For example, the directory `./makRti5.0.1/doc` appears in the text as `./doc`.

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. For an up-to-date list of supported third-party and open source libraries and the associated license information, please visit the MAK ONE Product Support Knowledge Base (at <https://www.mak.com/knowledge>) and refer to the Third-Party/Open Source List under Frequently Asked Questions (FAQ).



1. Introduction to the MAK RTI

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

1.1. The MAK RTI	16
1.1.1 MAK RTI Components	17
1.2. The rtiexec is not the RTI	17
1.3. Lightweight Mode	18
1.4. Compatibility with Other RTI Implementations	18
1.4.1 Obtaining HLA RTI Specification Documents	19
1.5. The RTI Assistant	19
1.5.1 The RTIsby Diagnostic GUI	19
1.6. The Unlicensed Version of the MAK RTI	20

1.1. The MAK RTI

The MAK RTI is a software library (and supporting executables) that implements the High Level Architecture (HLA) 1.3, HLA 1516 (SISO DLC HLA API 1516), HLA Evolved (IEEE 1516-2010), and HLA 4 (IEEE 1516-2025) 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 FOM data through RTI calls, which means that all HLA applications must use an RTI.

i When we use the terms "HLA 1516" or "1516", we mean the SISO DLC HLA API 1516 specification. When we use the term "HLA Evolved", we mean IEEE 1516-2010. When we use the term "HLA 4", we mean IEEE 1516-2025. When we discuss a subject that applies equally to HLA 1516, HLA Evolved, and HLA 4, we refer generically to the "HLA 1516 specifications".

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 only want to start their federate and participate in the federation.

An analogy is that of calling someone on a telephone. When you use 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 user 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 the [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.

No verification has been announced for HLA 4 (IEEE HLA 1516-2025). To learn about it, read "[HLA Evolves Again](#)" on the mak.com blog (<https://www.mak.com/learn/blog>).

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 ("[The RTI Forwarder Application](#)" in the *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" on page 57).
- The RTI Chooser (Windows only). The RTI Chooser helps you configure your MAK RTI environment. ("7.2. Configuring the MAK RTI Version and RTI Settings to Use" on page 69)
- The RTI Command Line Tool. This tool lets you send a limited number of commands to the RTI from a console window ("[The RTI Command Line Tool](#)" in the *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, see "1.3. Lightweight Mode" on the next page.)



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 where a federate is located. For more information about the rtiexec, please see "[The rtiexec](#)" in the *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 "[Using Lightweight Mode](#)" in the *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.

The MAK RTI HLA 4 implements the IEEE 1516-2025 specification. Applications built with it are compatible with any RTI written to this API.

 All 1516 specifications include the SISO DLC HLA API specification and are compatible with any other RTI that adheres to the HLA 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 "1.2. The rtiexec is not the RTI" on the previous page, every federate must have a copy of the RTI libraries installed on its local computer.

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

1.4.1 Obtaining HLA RTI Specification Documents

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

- <https://ieee.org>
- <https://sisostds.org>

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/mak-one/support/help/bonus-material>.

For more information about running unlicensed federations, see "8.2. Unlicensed Federations" on page 82.



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, see "2.10. Installing and Setting Up the MAK License Manager" on page 29.)

2.1. Required Environment Variables	22
2.2. Installing MAK RTI on Windows	22
2.2.1 Default Windows Installation Directory	22
2.3. Installing MAK RTI on Linux Systems	22
2.4. Installing PDB Files for Debugging Purposes	23
2.4.1 Installing PDB Files on Windows	23
2.4.2 Reporting an Issue Using the .callstack.log and .dmp Files	24
2.4.3 Specifying the Location for .log and .dmp Files	24
2.5. Configuring MAK RTI with Environment Variables	24
2.6. Installing the MAK RTI Java Bindings	26
2.6.1 Linux Installation	26
2.6.2 Windows Installation	26
2.6.3 Installing the HLA 1516 Java Fedtime Library	26
2.7. Uninstalling from Windows	27
2.8. Uninstalling from Linux	28
2.9. Managing Licenses for the MAK RTI	28
2.10. Installing and Setting Up the MAK License Manager	29
2.10.1 Specifying the License Server	30
2.11. Localizing the Graphical User Interface	31
2.11.1 Translating Other Interface Files	34
2.11.2 Applying the Language Files	34

2.1. Required Environment Variables

The MAK RTI requires some environment variables: MAK_RTIDIR, RTI_RID_FILE, and LD_LIBRARY_PATH or PATH. See "7.1.1 MAK RTI Environment Variables" on page 68 to learn more.

2.2. Installing MAK RTI on Windows

Windows versions of MAK RTI are provided as downloaded files.

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

Note the following:

- The installers are named to indicate the compiler used to build that version of MAK RTI.
- The MAK RTI installation wizard asks whether 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 PATH variables, 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.
- If you have a Java-based application that needs to connect to the MAK RTI, you must install the Java binding for the RTI. See "2.6. Installing the MAK RTI Java Bindings" on page 26.
- When you install large applications on Windows, there may be a delay of up to several minutes from the time you try to run the setup program to the time that an installation dialog box displays. This is due to how Windows scans setup programs before it executes them. If you experience this problem, turning off User Access Control can reduce or eliminate this delay.

2.2.1 Default Windows Installation Directory

The default root directory for installing MAK ONE products is *C:/MAK*.

You can use an environment variable, MAK_INSTALL_DIR, to specify a different location where all MAK ONE products and data should be installed by default. For example, if you define it as MAK_INSTALL_DIR = D:\MyProject, then all MAK ONE products automatically install to that location. The data packages and products would then install in that directory by default: *D:/MyProject/SharedData*, *D:/MyProject/vrlink5.10*, and so on.

2.3. Installing MAK RTI on Linux Systems

Linux versions of MAK RTI are provided as compressed TAR files.

► To install a product from a TAR file:

1. Create the directory in which you want to install the product.
2. Copy the tar file to the install directory.

3. Uncompress and untar the tar file:

```
tar -xzf applicationxx-os.tar.gz
```

where *xx-os* is the release, operating system, and other product release coding.

2.4. Installing PDB Files for Debugging Purposes

In cases where a MAK ONE application crashes, it can be difficult to determine the cause. The application generates a *.dmp* file that contains valuable information about the application's state at the time of the crash. MAK technical support can analyze this file to help diagnose the issue.

In addition, with the application's PDB (program database) files, the application generates a call stack log file (*.callstack.log*) of the application at the time of the crash. The call stack log provides information in plain text about the sequence of functions that were called leading up to the crash, which is also helpful in identifying the specific cause.

By providing the *.dmp* or *.callstack.log* files to technical support, you help support personnel to resolve the issues. They use specialized debugging tools to analyze the files and determine the cause of the crash, and that information aids in the development of a solution to prevent future crashes.

Due to security reasons or organization policies, some customers cannot send the *.dmp* file to technical support. Because the call stack log file is a plain text file, it is easier to inspect and analyze than the *.dmp* file. Users can easily attach it to an email and pass it out of secure areas with visual inspection, making it a convenient option for customers with restrictions on sharing binary files.

It is important to note that the call stack log file alone might not be sufficient for support personnel to fully diagnose the issue. They may require more information to gain a complete understanding of the issue. A detailed procedure for reproducing the crash often helps them solve the problem you encountered.



- Support for the generation of the call stack log file was introduced in MAK ONE 2024. It is supported only in MAK ONE products part of that release and later versions.
- The PDB files are part of the Linux installation. You do not need to install them separately.

2.4.1 Installing PDB Files on Windows

Before installing the PDB files on Windows, you must install the MAK RTI. Next, make sure that the PDB files you are installing are from the same build as the application. When you download the installer from the [MAK ONE Product Downloads](#) page, the appropriate PDB installer is also available for download.

► To install the PDB files:

1. After installing the MAK RTI, run the PDB installer for that product.
2. The installer looks for the matching product directory by searching the registry and then it automatically sets the install directory to the appropriate path. If that is not correct for your product, see the note below.
3. Complete the installation wizard. The necessary PDB files are installed in the product's `./bin64` directory.

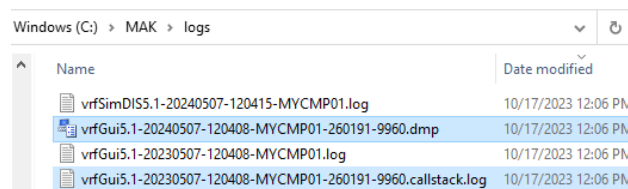


- The default installation directory is `C:\MAK`. See "2.2.1 Default Windows Installation Directory" on page 22 to learn about setting the default to other locations.
- Be sure that the folder in the Select Destination Location page of the installation wizard is correct. It should be the top-level folder for the product installation.
- If the PDB installer displays a warning that the SVN revisions do not match, make sure that you have the correct PDB installer for the version of the MAK RTI that you installed.

2.4.2 Reporting an Issue Using the `.callstack.log` and `.dmp` Files

Send the `.callstack.log` and `.dmp` files (or only the `.callstack.log` if you are unable to provide the `.dmp` file) to MAK support through the Support Portal (<https://www.mak.com/mak-one/support/portal>) or support@mak.com. The files will have similar names and are located in the central directory for MAK ONE log files (Figure 1). Please also provide information on anything you experienced when the issue occurred.

Figure 1. Example of `.callstack.log` and `.dmp` files for VR-Forces



2.4.3 Specifying the Location for `.log` and `.dmp` Files

The default directory for MAK ONE log files is `C:/MAK/logs`. You can customize the location by specifying a different path in the `MAK_LOG_DIR` environment variable.

The MAK RTI has an additional RID file parameter, `RTI_logFileDirectory`, that you can use to set a different location for RTI-specific log (`.log`) files.

2.5. Configuring MAK RTI with Environment Variables

This guide has references to how you can use environment variables to manage the MAK RTI behavior on your machine.

The following is a list of the variables mentioned in this guide, with brief descriptions and links to the topics where they are discussed in detail.

Table 1. MAK RTI Environment Variables

Environment Variable	Purpose	To learn more, see...
MAK_RTIDIR	Defines the location of the MAK RTI binaries and libraries.	"7.1.1 MAK RTI Environment Variables" on page 68.
PATH (Windows) LD_LIBRARY_PATH (Linux)	Specifies the path to the RTI dynamic library.	"7.1. Configuring Your System to Use the MAK RTI" on page 68.
RTI_ASSISTANT_IPV4_ADDRESS RTI_ASSISTANT_IPV6_ADDRESS	Specifies the multicast addresses that RTI Assistants use to communicate with each other.	"Specifying the RTI Assistant's Multicast Address" in the <i>MAK RTI Reference Manual</i> .
RTI_ASSISTANT_CLIENT_PORT	Sets a port for communication between a user's local RTI components and the RTI Assistant	"Specifying the RTI Assistant's Ports" in the <i>MAK RTI Reference Manual</i> .
RTI_ASSISTANT_DISABLE	Disables the RTI Assistant.	"Disabling the RTI Assistant" in the <i>MAK RTI Reference Manual</i> .
RTI_ASSISTANT_PORT	Specifies the RTI Assistant's peer and client ports.	"Specifying the RTI Assistant's Ports" in the <i>MAK RTI Reference Manual</i> .
RTI_ASSISTANT_REMOVE_SHUT_DOWN_ALL	Prevents users from shutting down the RTI Assistant.	"Shutting Down the RTI Assistant" in the <i>MAK RTI Reference Manual</i> .
RTI_ASSISTANT_TTL	Specifies the RTI Assistant's multicast time-to-live.	"Specifying the RTI Assistant's Multicast Time-to-Live" in the <i>MAK RTI Reference Manual</i> .
RTI_BUILD_TYPE	Used only by some legacy federates. If needed, set it to "." (period).	"7.1.1 MAK RTI Environment Variables" on page 68.
RTI_CONFIG	The absolute path to the directory containing the RTI configuration files.	"7.1.1 MAK RTI Environment Variables" on page 68.
RTI_HOME	Used only by legacy federates to point to the MAK RTI.	"7.1.1 MAK RTI Environment Variables" on page 68.
RTI_RID_FILE	Defines the absolute path to the RTI settings file, including the file name.	"7.1.1 MAK RTI Environment Variables" on page 68.

2.6. Installing the MAK RTI Java Bindings

The MAK RTI provides a set of Java bindings for each version (HLA 1.3, HLA 1516, HLA Evolved, HLA 4). 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 HLA 4 Java bindings, you must install a Java Development Kit (JDK) that supports Java 11 or later. To use Java bindings for all other HLA versions (HLA 1.3, HLA 1516, or HLA Evolved), you must install a JDK that supports Java 1.8. Once you have installed the required JDK, configure your system as described in the following sections.

2.6.1 Linux Installation

In addition to the normal MAK RTI setup, you must add or edit these 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 "2.6.3 Installing the HLA 1516 Java Fedtime Library" below.

2.6.2 Windows Installation

In addition to the normal MAK RTI setup, you must add or edit these 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 "2.6.3 Installing the HLA 1516 Java Fedtime Library" below.

2.6.3 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 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;
```

2.7. Uninstalling from Windows

MAK ONE product installations include an uninstaller application (*unins000.exe*), which you can find in the installation directory. Run that program to uninstall the product.

When you run other installers for the product, such as the PDBs, there are additional uninstallers. If you run the uninstallers from the highest number down (for example, *unins002.exe* and then *unins001.exe*), running *unins000.exe* uninstalls everything else and removes the install folder.

When you uninstall VR-Vantage, VR-Forces, or VR-Engage, files such as scenarios or SMSs that reside in the product directory are also deleted. Before uninstalling, copy any of your custom files from the product directory to another folder.

The uninstaller does not remove the data package. That is in a separate directory (for example, *./SharedData/##/latest*).

i If you run only the product uninstaller, you can then simply delete the rest of the product's install folder. Note that if you do not run all of the uninstallers, running only the product uninstaller may not remove all of the shortcuts from the Windows Start menu.

2.8. Uninstalling from Linux

To remove installations from Linux machines, simply delete the installation directory. This catches all of the files associated with the product.

When you uninstall VR-Vantage, VR-Forces, or VR-Engage, files such as scenarios or SMSs that reside in the product directory are also deleted. Before uninstalling, copy any of your custom files from the product directory to another folder.

Deleting the installation directory does not remove the data package. That is in a separate directory (for example, `./SharedData/##/latest`).

2.9. 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.

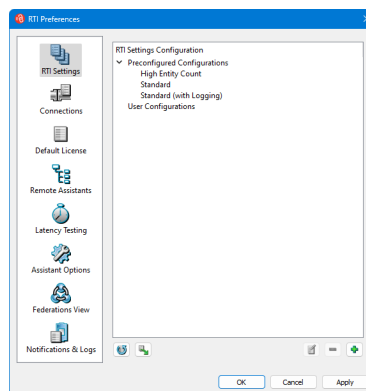
The `rtiexec` and all of the federates using this feature must enable it. The `rtiexec` license checkout feature does not support dongle licenses.

i 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:

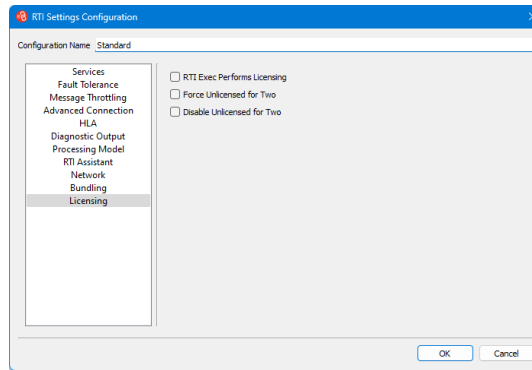
1. In the system tray, right-click the RTI Assistant icon (🔧). A context menu displays.
2. Choose **Preferences**. The RTI Preferences dialog box opens (Figure 2).

Figure 2. 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 3).

Figure 3. Licensing page



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

2.10. Installing and Setting Up the MAK License Manager

Before you can use a MAK ONE 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 ONE products. You only need to install it on the computer that you will use as the license server.

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

The License Manager installer is included with MAK installation media. It is separate from the product installers. You can download the installers from the [MAK ONE Product Downloads](https://www.mak.com/mak-one-downloads) page at <https://www.mak.com/mak-one-downloads>.

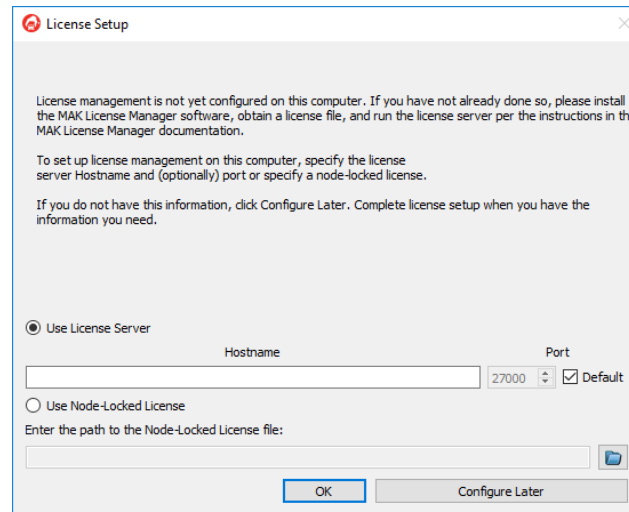
Complete installation and configuration instructions are included with the License Manager installer. The *MAK ONE Installation Guide* is available at <https://mak.com/mak-one-installation-guide>.

Some customers use dongle licenses instead of running a license server. Instructions for using dongles are also in the *MAK ONE Installation Guide*.

2.10.1 Specifying the License Server

The first time you run a MAK ONE product on a particular computer, the License Setup dialog box opens (Figure 4). It prompts you to enter the hostname of the license server and, optionally, a port number.

Figure 4. 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 ONE products until you set up license management.

If you know the hostname, enter it in the Hostname box and 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 ONE 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 ONE product.
- If you have changed the license server, you may want to clear the license cache. 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, see "8.3.3 Clearing Default License Server Settings" on page 85.

Setting 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 or node-locked license 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.

For license servers, 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 ONE products run on.

Windows

► To set MAKLMGRD_LICENSE_FILE in Windows:

1. On the Start menu, click the Settings button (⚙️). The Settings window opens.
2. In the Search box, enter `environment`. A list of choices displays.
3. Select `Edit the System Environment Variables`. The System Properties dialog box opens to the Advanced tab.
4. Click the Environment Variables button. The Environment Variables dialog box opens.
5. Below the System Variables list, click `New`. The New System Variable dialog box opens.
6. In the Variable Name field, enter `MAKLMGRD_LICENSE_FILE`.
7. In the Variable Value field, enter `@server_name`, where `server_name` is the name of the license server.
8. Click `OK` to save out of each dialog box and set the variable.

Linux

On Linux, you set environment variables in your `.cshrc` (or equivalent startup file).

► To specify a license server, set the variable similarly to:

```
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:

```
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 ONE products.

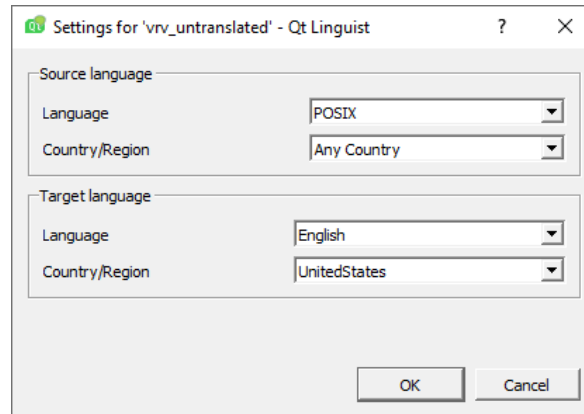
2.11. Localizing the Graphical User Interface

You can localize your MAK ONE 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, `makRti_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 MAK RTI, lets you map the English text provided by MAK to your chosen language.

► To edit a resource file:

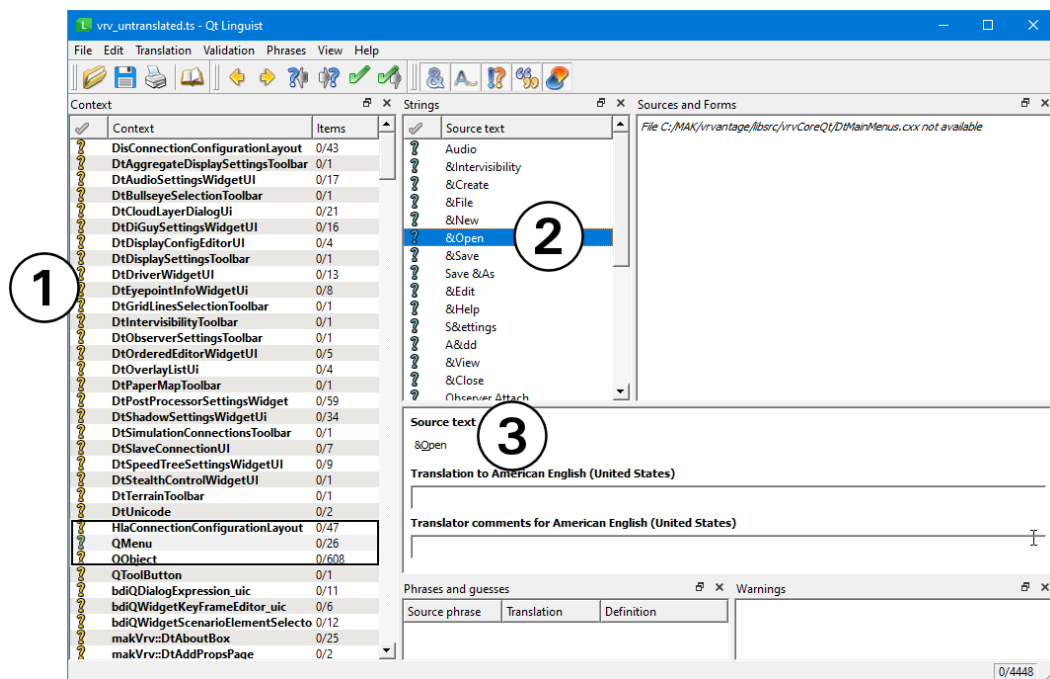
1. Start the Qt Linguist utility. It is `./translations/linguist.exe`
2. In the Qt Linguist window, choose **File > Open**. Select the file `./translations/product_untranslated.ts`. The Settings for *filename* dialog box opens (Figure 5).

Figure 5. Linguist settings dialog box



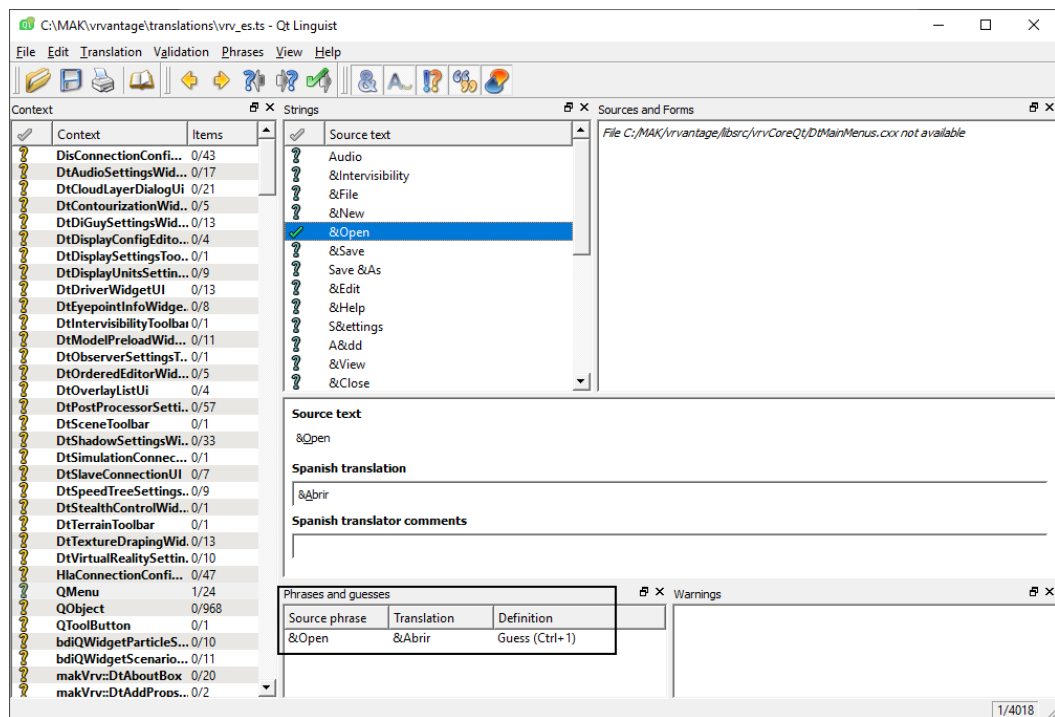
3. Select values for the source language, target language, and target region or country.
4. Click OK. The translation file opens and a list of classes displays on the left side of the window (callout 1 in Figure 6).

Figure 6. Linguist window



5. To translate text, 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 only need to translate the text. In Figure 6, **QMenu** is selected. The Strings window lists text that can be translated (callout 2). Text that has not been translated has a blue question mark in the checkmark column.
6. In the Source Text list, select the word you want to translate. The text displays under Source Text in the translation area (callout 3 in Figure 6).
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 7). If you move to a different text string without clicking Done and Next, a yellow question mark displays 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 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.

2.11.1 Translating Other Interface Files

The *makRti_untranslated.ts* file contains all the text used in the MAK RTI interface. However, there may be other text and messages that are not specific to MAK RTI that are generated by the software that implements the MAK 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*.

2.11.2 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 MAK RTI with the (`--locale` | `-G`) *language* command-line option, where *language* is the two-letter 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.

3.1. The High Level Architecture	36
3.1.1 The RTI	36
3.1.2 Federates and Federations	36
3.2. RTI Services	36
3.2.1 Federation Management	36
3.2.2 Declaration Management	37
3.2.3 Object Management	37
3.2.4 Ownership Management	37
3.2.5 Time Management	37
3.2.6 Data Distribution Management	37
3.2.7 Miscellaneous Support Services	38
3.3. FOMs and FED Files	38
3.4. RTI Settings and the RTI Initialization Data (RID) File	38

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 a set of services to a federation:

- Federation management
- Declaration management
- Object management
- Ownership management
- Time management
- Data distribution management
- Miscellaneous support services

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 the [Using Data Distribution Management](#) chapter in the *MAK RTI Reference Manual*.

3.2.7 Miscellaneous Support Services

Miscellaneous support services include mapping between names and handles, managing switches, manipulating regions, evoking callbacks, and so on.

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.

i 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 "[Specifying the FED File](#)" in the *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 the [RTI Settings](#) chapter in the *MAK RTI Reference Manual*.

i 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.

4.1. Connecting a Federate to the MAK RTI	40
4.1.1 Troubleshooting Federate Connection Problems	42
4.2. Starting the rtiexec	43
4.2.1 Starting the rtiexec when Starting a Federate	47
4.2.2 Starting the rtiexec from the Federations View	49
4.2.3 Shutting Down the rtiexec	49
4.3. Starting an RTI Forwarder from the Federations View	50
4.3.1 Shutting Down an RTI Forwarder	50
4.4. Resigning a Federate from a Federation	50

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 IP address and port 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.

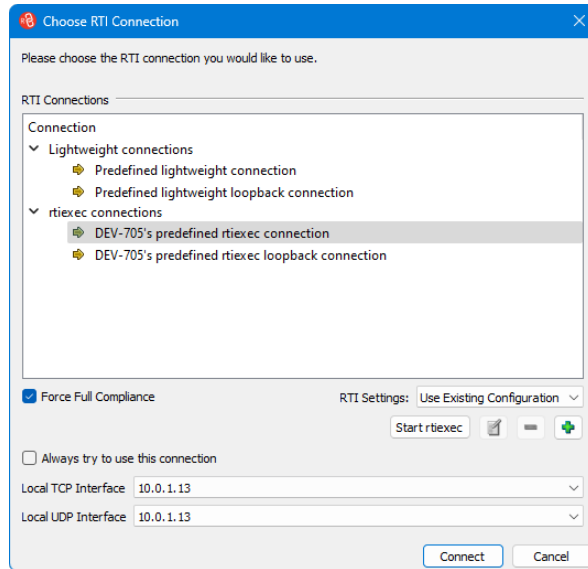
By default, when you start a federate, the MAK RTI prompts you to select an RTI connection configuration. If you always want to use the same connection, you can configure the MAK RTI to always try to use that connection without prompting you to select it.

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 "[Configuring Federate Connections in the RTI Settings](#)" in the *MAK RTI Reference Manual*.)

► To connect to the MAK RTI:

1. Start the federate. The Choose RTI Connection dialog box opens (Figure 8).

 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 8. 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, see "7. Configuring the MAK RTI Startup Environment" on page 67.) 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:

- 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, see "4.2.1 Starting the rtiexec when Starting a Federate" on page 47.)
- 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:

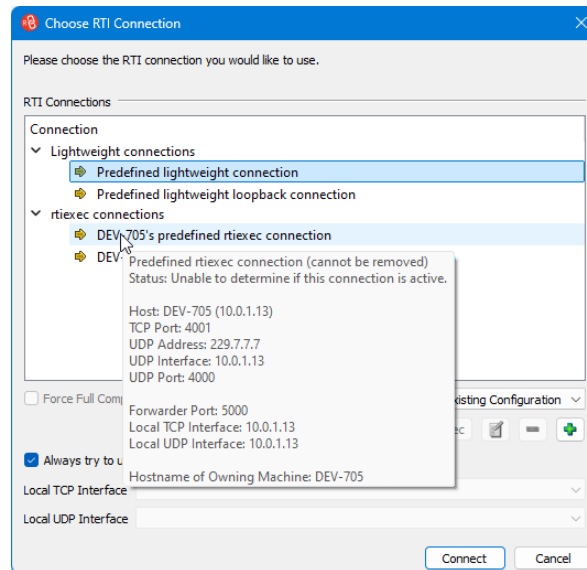
- Green arrow: The RTI Assistant has verified that there is a federate running on that connection. You can connect the federate to this connection.
- Yellow arrow: The RTI Assistant knows that a connection configuration exists, but does not know if any federates are using it. You can connect the federate

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.

i 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 9).

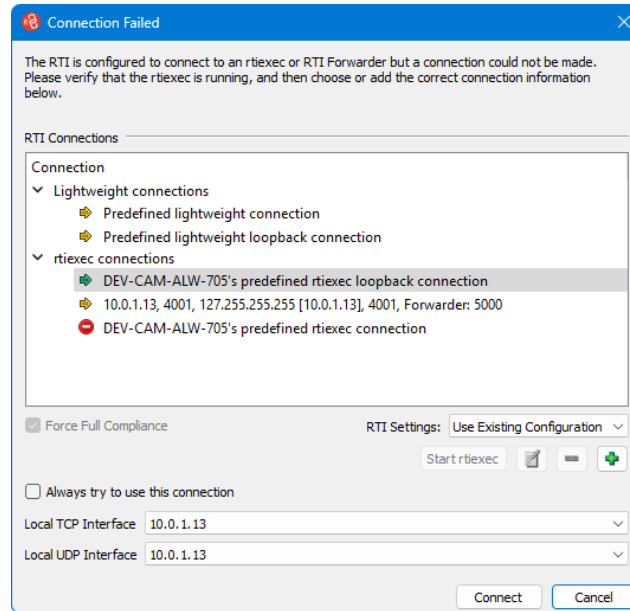
Figure 9. Connection details



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, see "4.1.1 Troubleshooting Federate Connection Problems" below.

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 10).

Figure 10. 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 and 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, see "4.2.1 Starting the rtiexec when Starting a Federate" on page 47.)
- 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, see "7.3. Managing Connection Configurations" on page 71.

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, see "4.2.1 Starting the rtiexec when Starting a Federate" on page 47.)

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, see "7.3. Managing Connection Configurations" on page 71.

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 "[The RTI Forwarder Application](#)" in the *MAK RTI Reference Manual*.

► **To start the rtiexec from the Windows Start menu:**

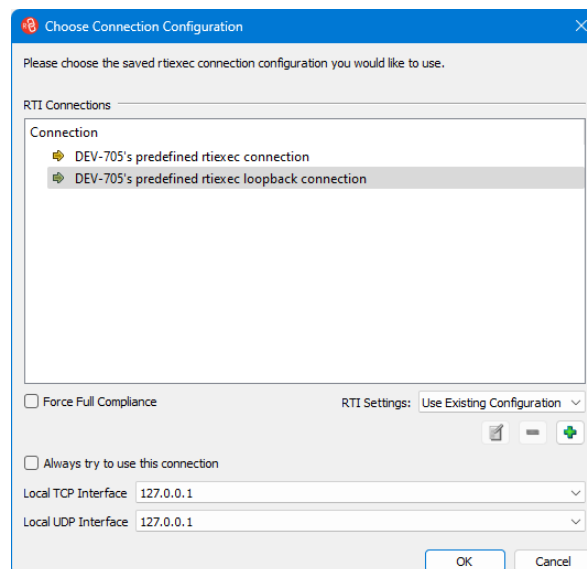
1. On the Start menu, choose **MAK Technologies > rtiexec**. If the RTI Assistant is not already running, it starts and its icon is added to the system tray (Figure 11). The Choose Connection Configuration dialog box opens (Figure 12).

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 does not display. The rtiexec starts immediately.

Figure 11. RTI Assistant icon



Figure 12. 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, see "7.3. Managing Connection Configurations" on page 71.)



- 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 "[The rtiexec](#)" and "[Enabling and Disabling HLA Services](#)", both in the *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 2 lists command-line options for the rtiexec.

Table 2. rtiexec command-line options

Option	Description
<code>(--autoExit -K)</code>	Specifies that the rtiexec exit automatically after the last connection exits.
<code>(--destAddrString -A) address</code>	Specifies the multicast group address to use if reliable transport is disabled. Equivalent to the RTI_destAddrString parameter. This option is ignored unless <code>--manual</code> is used.
<code>(--distributedForwarderPort -D) port</code>	Specifies the port that RTI Forwarders use to communicate with each other.

Table 2. rtiexec command-line options (continued)

Option	Description
(--doNotUseRoutesFile -d) <i>address</i>	Disables the use of a routes file for an RTI Forwarder started by this rtiexec.
(--enableProfiling -e)	Enables profiling. Primarily for internal use.
(--forceFullCompliance -f)	Specifies that the rtiexec run in fully compliant mode. This option is ignored unless --manual is used.
(--forwarderToConnectTo -F) <i>address</i>	Specifies the address of an additional forwarder that the forwarder started by the rtiexec should try to connect to instead of using the routes file. Once connected to the designated forwarder, this forwarder will connect to the forwarders connected to the designated forwarder. As a result, this forwarder will join the fully connected mesh of forwarders.
(--manual -M)	Specifies that the rtiexec will 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.
(--notifyLevel -n) <i>notification_level</i>	Specifies the notification level for application messages, where <i>notification_level</i> is a number from 0 through 4. The least verbose message level is 0, the highest is 4. The default is 2.
(--setLogFileName -l) <i>filename</i>	Specifies the log file name for the rtiexec.
(--setNotifyQuiet -q)	Suppresses output of diagnostic messages (sets the notification level to 0).
(--setRidFileName -R) <i>RID_file</i>	Specifies the RTI initialization data (RTI settings) file to use (for example, <i>rid.mtl</i>). --setRidFileName overrides the RTI_RID_FILE environment variable.
(--setTcpPort -T) <i>port</i>	Specifies the port to use when connecting to the application using TCP. Equivalent to the RTI_tcpPort parameter. This option is ignored unless --manual is used.

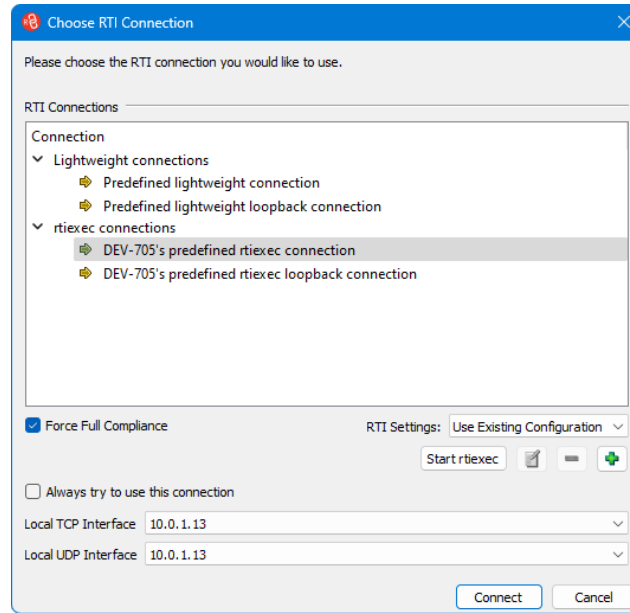
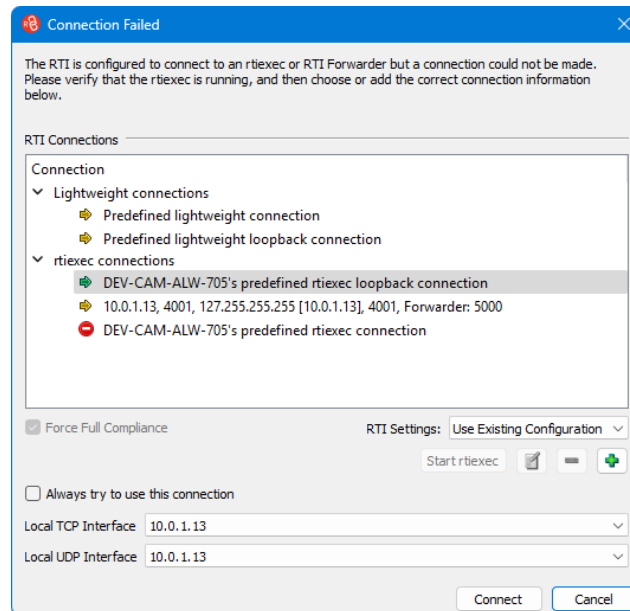
Table 2. rtiexec command-line options (continued)

Option	Description
(--setUdpPort -P) <i>port</i>	Specifies the port to listen to. Equivalent to the RTI_udpPort parameter. This option is ignored unless <code>--manual</code> is used.
(--tcpAddress -t) <i>address</i>	Specifies the IP address used for TCP connections (reliable transport).
(--tcpNetworkInterfaceAddr -i) <i>address</i>	The IP address of the local network interface used for TCP connections (reliable transport).
(--udpNetworkInterfaceAddr -N) <i>address</i>	Specifies the IP address of the network card that you want this forwarder to use for UDP connections (best effort transport).
(--usePreconfiguredRid -p)	Indicates that the RID file specified by <code>--setRidFileName</code> is the name of a preconfigured RTI Settings instead of a file path.
(--useReliable -r)	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 <code>--manual</code> is used.
(--ignore_rest --)	Ignores all command-line options that are listed after this argument. This option is useful for disabling part of a command line in a script for test purposes so that you do not have to retype all of the command options at a later point.
(--version -v)	Displays version information.
(--help -h)	Displays command-line usage information in a console window.

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 13) 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 14 on the next page). Both of these dialog boxes allow you to 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 13. Choose RTI Connection dialog box**Figure 14. 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.

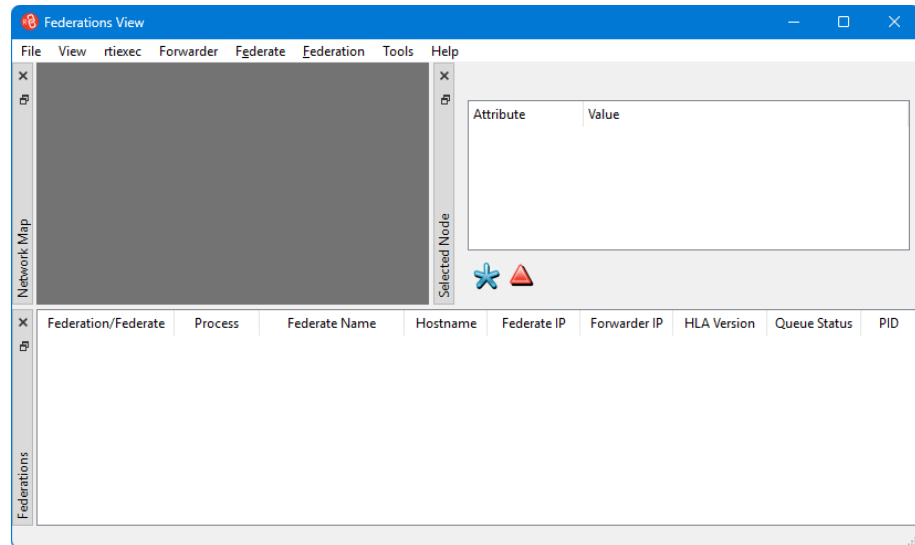
4.2.2 Starting the rtiexec from the Federations View

i 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 context menu displays.
2. Choose **Federations View**. The Federations View opens (Figure 15).

Figure 15. Federations View - no rtiexec running



3. Choose **rtiexec > Start rtiexec**, or click the Start Executive button (🔴) on the Selected Node toolbar. The Choose Connection Configuration dialog box opens (Figure 12 on page 44).
4. In the Choose Connection Configuration group box, select the connection you want to use.
5. Optionally, select the Force Full Compliance check box.
6. Optionally select an RTI Settings configuration.
7. Click OK.

4.2.3 Shutting Down the rtiexec

► **To shut down the rtiexec, use one of these 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 (the right-hand 🔴) on the Selected Node view.
- Kill the rtiexec process.

- Use the RTI Command Line Tool to send a `kill` command. For details, please see "[The RTI Command Line Tool](#)" in the *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 from the Federations View

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 context menu displays.
2. Choose **Federations View**. The Federations View opens (Figure 16 on the facing page).
3. Choose **Forwarder > Start Forwarder**, or click the Start Forwarder button () on the Selected Node toolbar. The Choose Connection Configuration dialog box opens.
4. 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.)
5. 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 context menu displays.
2. Choose **Federations View**. The Federations View opens (Figure 16 on the facing page).
3. Choose **Forwarder > Shut Down Forwarder**, or click the Shut Down Forwarder button (the left-hand ) on the Selected Node toolbar.

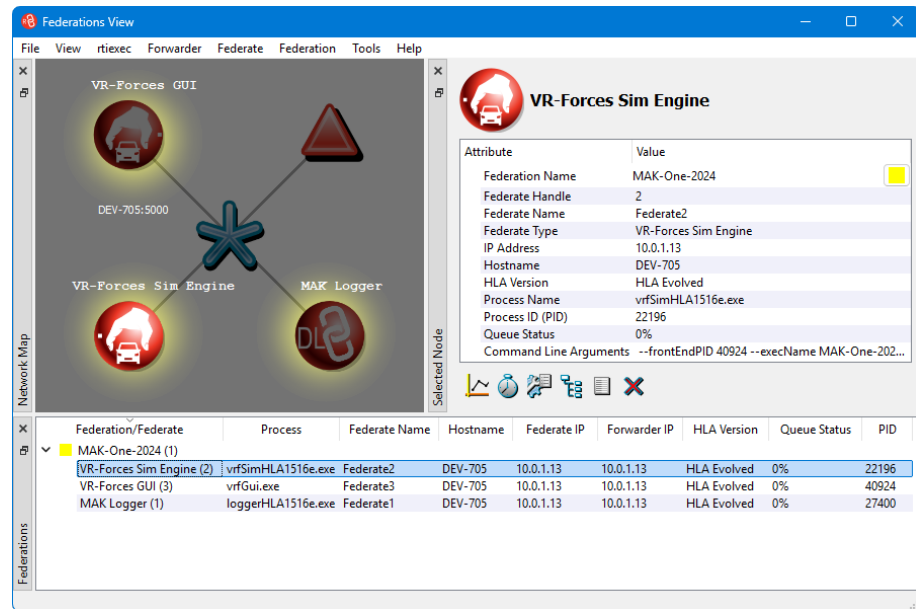
4.4. Resigning a Federate from a Federation

When you shut down a federate, it resigns from the federation. The RTI Assistant and RTIs Spy allow you to 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 context menu displays.
2. Choose **Federations View**. The Federations View opens (Figure 16).

Figure 16. 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

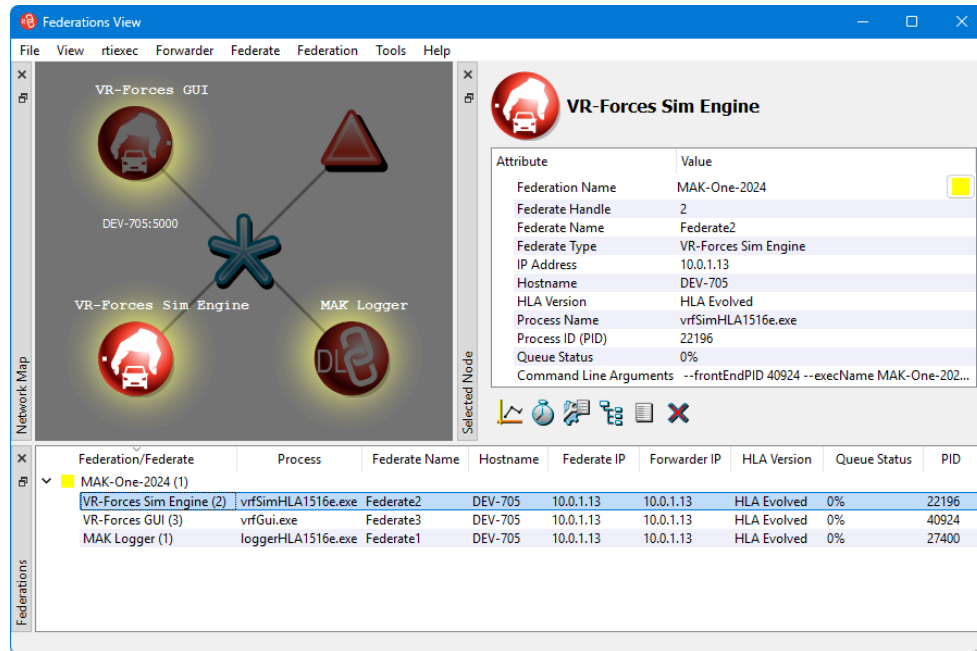
Although this manual focuses on starting federates, with little attention to broader federation management issues, you are creating a federation even when you run only one federate. This chapter covers some basic federation management tasks.

5.1. Opening the Federations View	54
5.2. Destroying Federations	54
5.2.1 Shutting Down all Local RTI Components	54
5.3. Removing Disconnected Federates from the Display	55

5.1. Opening the Federations View

The RTI Assistant helps you view and manage federations in the Federations View (Figure 17).

Figure 17. Federations View



► To open the Federations View:

1. In the system tray, right-click the RTI Assistant icon (RTI Assistant icon). A context menu displays.
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 rtiexecs 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 context menu displays.
2. Choose **Federations View**. The Federations View opens (Figure 17 above).
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 context menu displays.
2. Choose **Shut Down All**.

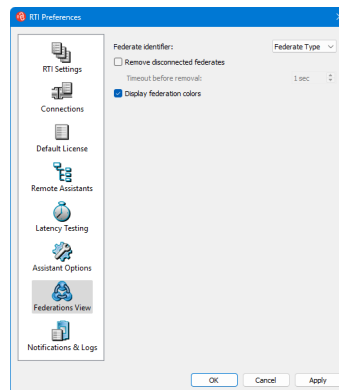
5.3. Removing Disconnected Federates from the Display

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 18).

Figure 18. 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, see "6.3. Configuring the RTI Assistant's Startup Behavior" on page 61.

6.1. Starting the RTI Assistant	58
6.2. The RTI Assistant Windows	58
6.2.1 Federations View	58
6.2.2 RTI Preferences Dialog Box	60
6.2.3 Local Component Notification History Window	61
6.3. Configuring the RTI Assistant's Startup Behavior	61
6.4. Specifying the RTI Assistant's Exit Policy	62
6.5. Configuring the RTI's Crash Behavior	62
6.6. Connecting to Remote Assistants	63
6.6.1 Adding an Assistant Connection	63
6.6.2 Removing an Assistant Connection	64
6.6.3 Connecting to or Disconnecting from an Assistant Connection	65
6.7. Shutting Down the RTI Assistant	65

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:**

1. On the Windows Start menu, choose **MAK Technologies > MAK RTI 5.0.1 Utilities**. The Utilities folder opens.
2. Double-click the RTI Assistant shortcut. The RTI Assistant icon is added to the system tray (Figure 19).

Figure 19. RTI Assistant icon



6.2. The RTI Assistant Windows

The RTI Assistant has these windows and dialog boxes:

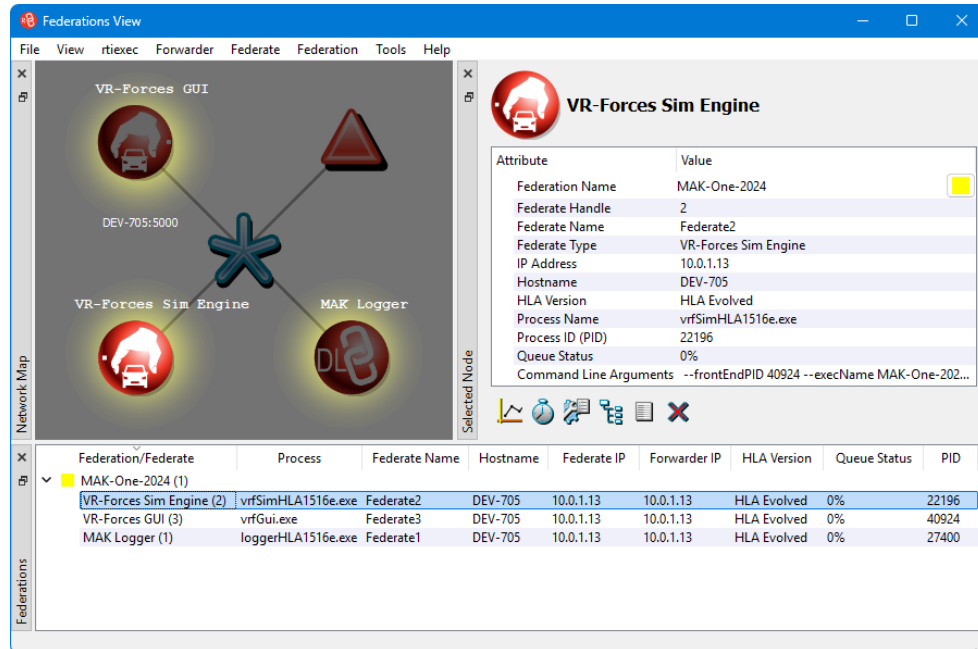
- Federations View
- RTI Preferences dialog box
- Local Notification History.

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 20) 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 RTIs spy diagnostic GUI.

Figure 20. Federations View



The Federations View displays:

- **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 context menu displays.
2. Choose **Federations View**.

i 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 20). 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 display 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.

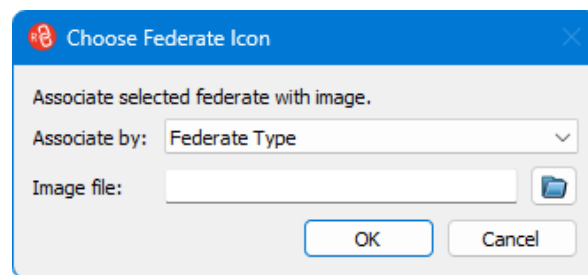
Changing a Federate's Icon

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. At the top of the Selected Node Panel, double-click the icon displayed next to the federate's name. The Choose Federate Icon dialog box opens (Figure 21).

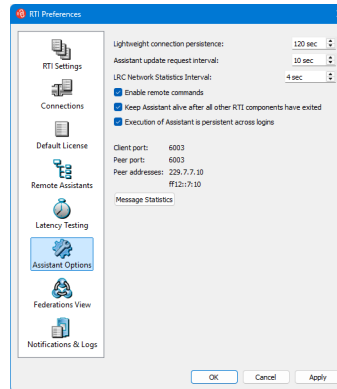
Figure 21. Choose Federate Icon dialog box



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, enter the path to an icon image or browse to find the image. You can choose a BMP, JPG, JPEG, PNG, GIF, or SVG file.
5. Click OK.

6.2.2 RTI Preferences Dialog Box

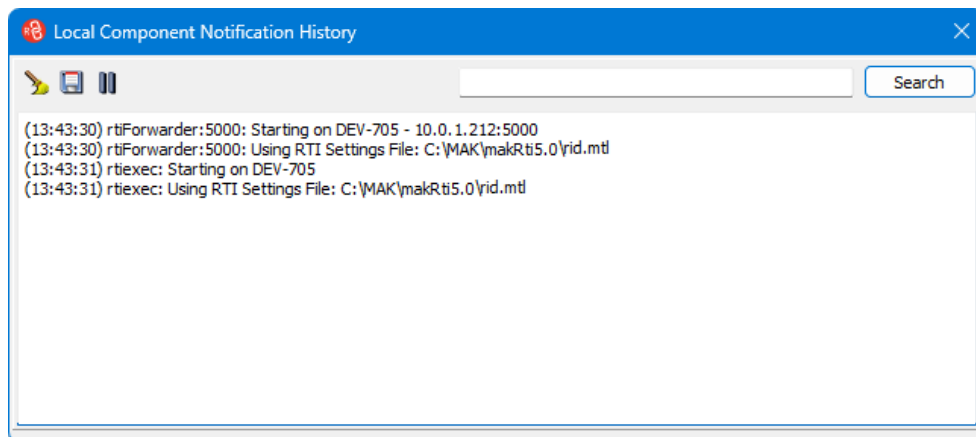
Use the RTI Preferences dialog box (Figure 22) to configure RTI settings and the RTI Assistant. You can also configure latency testing.

Figure 22. RTI Preferences dialog box

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

6.2.3 Local Component Notification History Window

The Local Component Notification History window (Figure 23) displays notification messages sent by the local rtiexec and LRCs.

Figure 23. 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 opens.
 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 context menu displays.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Assistant Options page (Figure 22 on the previous page).
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 opens.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Assistant Options page (Figure 22 on the previous page).
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 displays.
- 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.

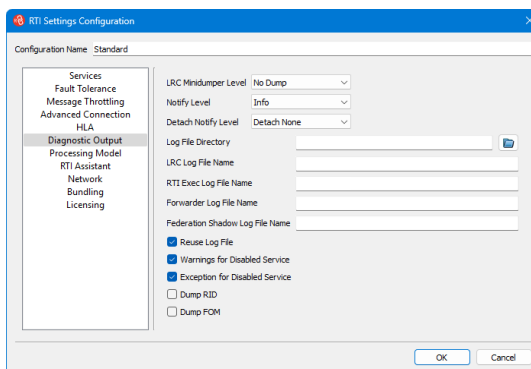
On Linux, if you select the Minidump option, the RTI prints out a stack trace. The other options have no effect.

► **To configure an RTI component's crash behavior:**

1. In the system tray, right-click the RTI Assistant icon (). A context menu displays.
2. Choose **Preferences**. The RTI Preferences dialog box opens (Figure 22 on the previous page).
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 24).

Figure 24. 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.

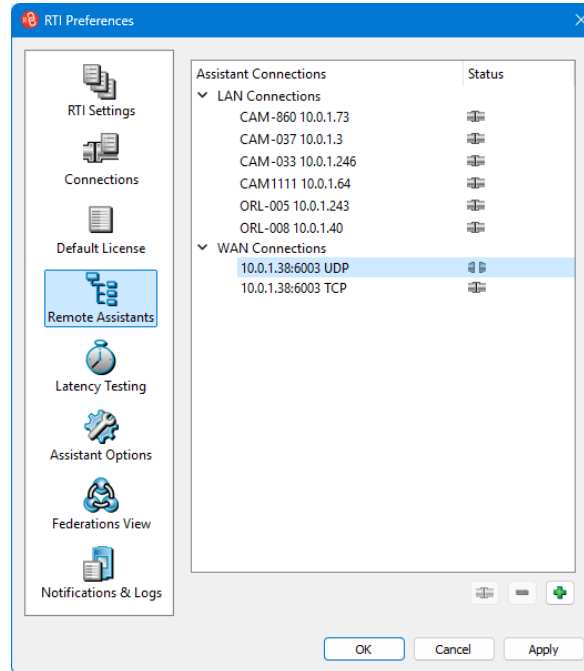
For any given IP address and port, you can create two connection, one for UDP and one for TCP. Only one of these connections can be active at a time.

► To add an Assistant Connection:

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

3. Select the Remote Assistants page.

Figure 25. 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. Enter 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 (RTI Assistant icon). A menu opens.
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 opens.
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 context menu displays.
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 opens.
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 opens.
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.

7.1. Configuring Your System to Use the MAK RTI	68
7.1.1 MAK RTI Environment Variables	68
7.1.2 Specifying RTI Settings Programmatically in HLA 1516	69
7.2. Configuring the MAK RTI Version and RTI Settings to Use	69
7.2.1 Adding an RTI Configuration	70
7.2.2 Editing an RTI Configuration	71
7.3. Managing Connection Configurations	71
7.3.1 Adding rtiexec Connection Configurations	73
7.3.2 Adding RTI Connection Configurations	75
7.3.3 Editing Connections	77
7.3.4 Viewing Connection Status	77
7.3.5 Removing Connection Configurations	78
7.4. Default Port Values	78

7.1. Configuring Your System to Use 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, see "7.2. Configuring the MAK RTI Version and RTI Settings to Use" on the facing page.

The RTI dynamic library must be located in your dynamic library search path. This path is specified in an environment variable, as listed in Table 3.

Table 3. 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\makRti5.0.1*. It is used to add the MAK RTI to search paths for libraries and executables. It is also used to build the RTI examples.
- **PATH** or **LD_LIBRARY_PATH** – The RTI dynamic library must be located in your dynamic library search path. This path is specified in an environment variable: **PATH** is the variable for Windows and **LD_LIBRARY_PATH** is the one for Linux. For more information, see "7.1. Configuring Your System to Use the MAK RTI" above.

- 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 MAK 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.
- RTI_FEDPRO_PORT – The port used to establish connections between the Federate Protocol federates and server. Default: 15164.

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

- RTI_ASSISTANT_PORT
- RTI_ASSISTANT_CLIENT_PORT
- RTI_ASSISTANT_IPV4_ADDRESS
RTI_ASSISTANT_IPV6_ADDRESS
- RTI_ASSISTANT_TTL
- 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 "[Specifying RTI Settings Programmatically in HLA 1516](#)" in the *MAK RTI Reference Manual*.

7.2. Configuring the MAK RTI Version and RTI Settings to Use

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.

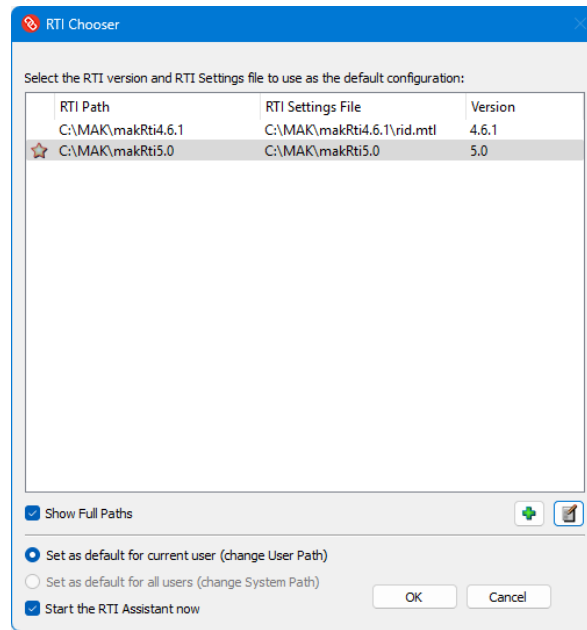
 The RTI Chooser is only available on Windows.

 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. To open the RTI Chooser, on the Windows Start menu, choose **MAK Technologies > MAK RTI 5.0.1 Configuration**. In the Configuration folder, double-click the RTI Chooser shortcut. The RTI Chooser opens (Figure 26). The RTI Chooser window lists the known RTI configurations.

Figure 26. RTI Chooser



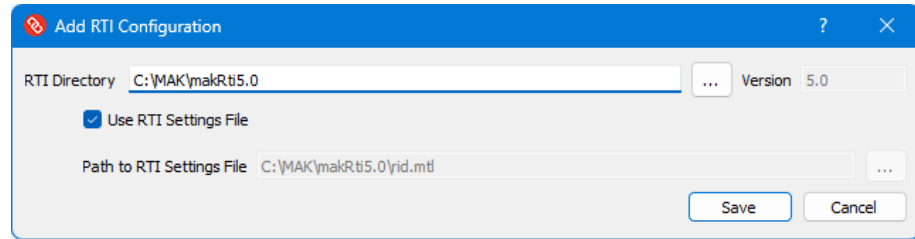
2. If you want to use an existing configuration, select it in the window. (You can add configurations. For details, see "7.2.1 Adding an RTI Configuration" below.)
3. Specify whether you want to change the PATH environment variable for the current user (yourself), or for all users of this computer (system 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. To open the RTI Chooser, on the Windows Start menu, choose **MAK Technologies > MAK RTI 5.0.1 Configuration**. In the Configuration folder, double-click the RTI Chooser shortcut. The RTI Chooser opens.
2. Click the Add button (+). The Add RTI Configuration dialog box opens (Figure 27).

Figure 27. Add RTI Configuration dialog box

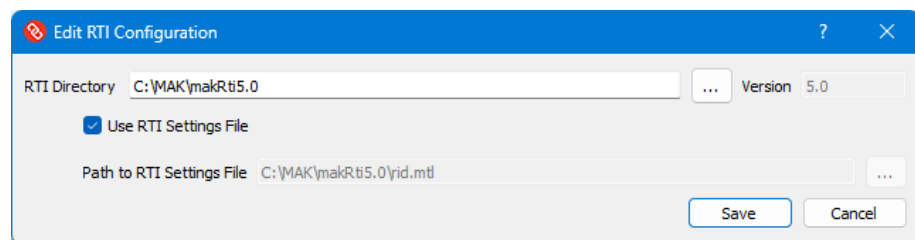
3. Enter the path to the MAK RTI version that you want to configure, or click the browse button and select the installation directory for the MAK RTI version. The RTI Chooser automatically tries to detect the compiler version of the MAK RTI installation.
4. To use the default RID file (*rid.mtl*), select 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. Open the RTI Chooser.
2. Double-click the RTI Chooser shortcut. The RTI Chooser opens (Figure 26 on the previous page).
3. In the window, select the configuration you want to edit.
4. Click the Edit button (✎). The Edit RTI Configuration dialog box opens (Figure 28).

Figure 28. Edit RTI Configuration dialog box

5. Change the MAK RTI version or RID file.
6. 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 uses 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 4 describes the parameters in a connection. Table 5 on the facing page lists parameter combinations that must be unique.

Table 4. 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 lightweight	rtiexec: 4001 lightweight: 4000	Port used for best effort communications.
Best Effort Address	rtiexec lightweight	229.7.7.7	Multicast address used for best effort traffic.
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 5. 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.

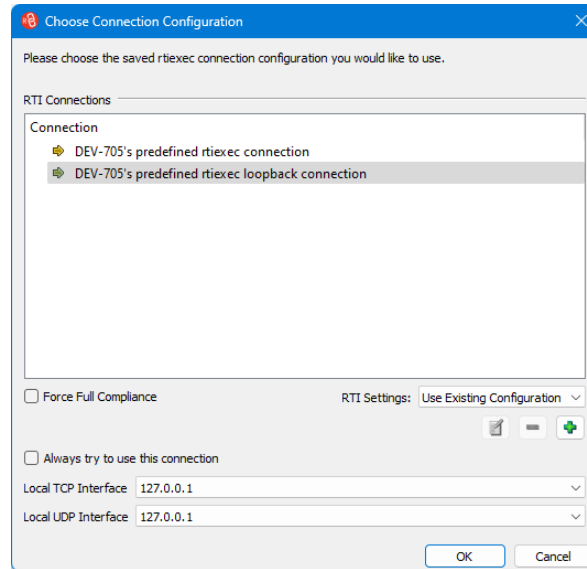
Connection configurations managed from the RTI Assistant override command-line options and RTI settings file parameters unless you force the MAK RTI to use them. The following settings are affected as indicated:

- **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

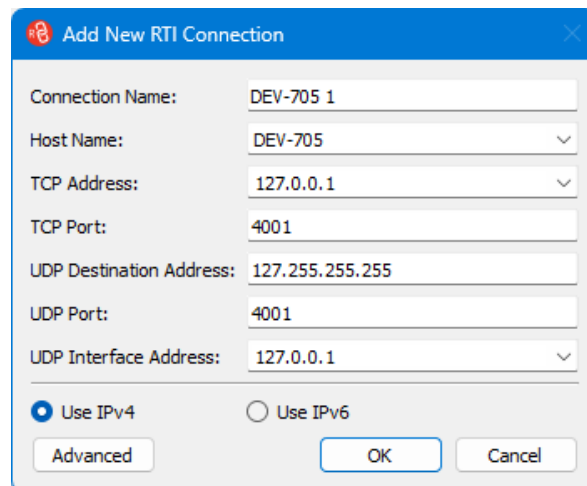
You can add rtiexec connection configurations for your computer in the Choose Connection Configuration dialog box and in the RTI Preferences dialog box, Connections page. If you have specified that the rtiexec always try to use a selected connection, the Choose Connection Configuration dialog box does not open when the rtiexec starts up. In that case you would have to make changes in the RTI Preferences dialog box. (For details, see "7.3.2 Adding RTI Connection Configurations" on page 75.)

- **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 29).

Figure 29. Choose Connection Configuration dialog box

2. Click the Add button (+). The Add New RTI Connection dialog box opens (Figure 30). The dialog box is populated with default parameters and a connection name built from those parameters.
3. Optionally, change the connection name.

i 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 30. Add New Local rtexec Connection dialog box

i You can only add rtexec connection configurations for the local host.

4. Enter the ports you want to use in the TCP Server Port (reliable connection) and UDP Port (best effort connection) text boxes.
5. Enter 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 "[Configuring RTI Forwarders Using the RTI Assistant](#)" in the *MAK RTI Reference Manual*.
8. Click OK.

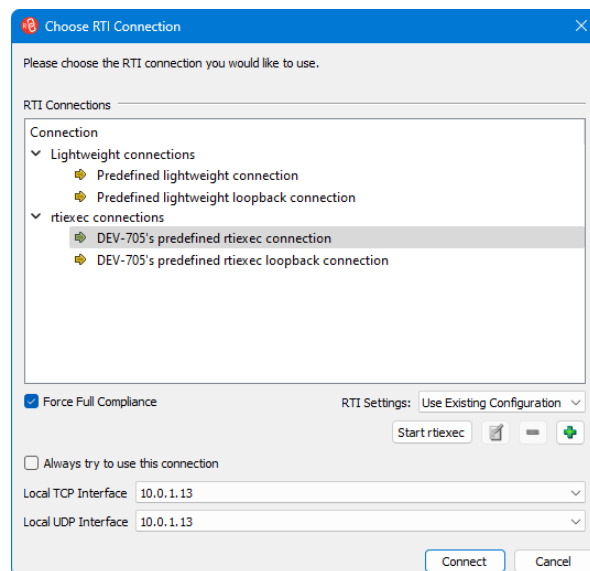
7.3.2 Adding RTI Connection Configurations

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 31), start a federate.

Figure 31. Choose RTI Connection dialog box

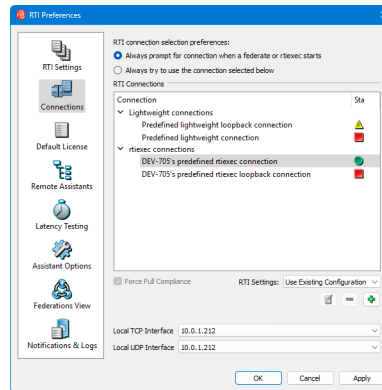


To open the RTI Preferences, Connections page:

- a. In the system tray, right-click the RTI Assistant icon (🔧). A context menu displays.
- b. Choose **Preferences**. The RTI Preferences dialog box opens.
- c. Select the Connections page (Figure 32).

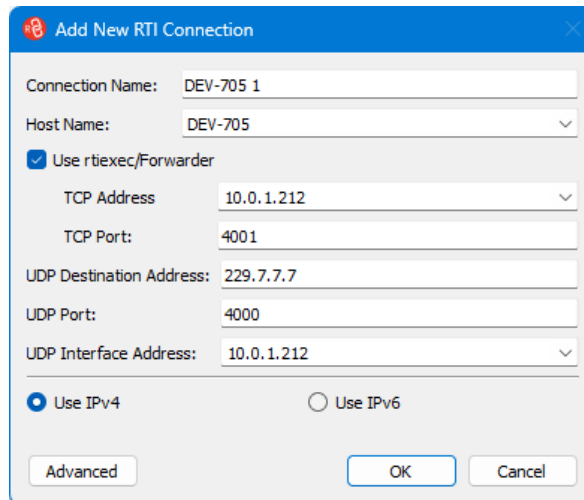
i You can also open the RTI Preferences dialog box to the Connections page from the Start menu by choosing **MAK Technologies > MAK RTI 5.0.1 Configuration**. In the Configuration folder, double-click the Configure Connections shortcut.

Figure 32. RTI Preferences dialog box, Connections page



- Click the Add button (+). The Add New RTI Connection dialog box opens (Figure 33). The dialog box is populated with default parameters and a connection name built from those parameters.

Figure 33. Add RTI Connection dialog box



- Optionally, change the connection name.

i 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.

- If you want to add an rtiexec connection configuration:

- a. Select the Use rtiexec check box.
 - b. Enter the hostname or IP Address of the computer on which the rtiexec will run.
5. Enter values in the TCP Server Port and UDP Port text boxes.
6. Enter 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, enter 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 "[Configuring RTI Forwarders Using the RTI Assistant](#)" in the *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 29 on page 74), Choose RTI Connection dialog box (Figure 31 on page 75), or RTI Preferences dialog box, Connections page (Figure 32 on the previous page), 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 Local rtiexec Connection dialog box (Figure 30 on page 74) or Edit RTI Connection dialog box (Figure 33 on the previous page) 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 32 on the previous page) 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.



- 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 6 lists the default ports for the various MAK RTI components.

Table 6. Default port values

Port	Use	Required	Asst†	RTI Setting
4000	UDP Port	Yes	Yes	RTI_udpPort
4001	TCP Port	No	Yes	RTI_tcpPort
5000	RTI Forwarder Port	No	Yes	RTI_distributedForwarderPort
	Notes:	The port used by RTI Forwarders to communicate with each other.		
6001	Multicast Discovery Port	No	No	RTI_mcastDiscoveryPort
	Notes:	Only needed if multicast discovery is on.		
6003	RTI Assistant Port	No	No	RTI_ASSISTANT_PORT (environment variable)
	Notes:	The port used by RTI Assistants to communicate with each other.		

Table 6. Default port values (continued)

Port	Use	Required	Asst‡	RTI Setting
15164	Federate Protocol	No	No	RTI_FEDPRO_PORT (environment variable)
‡ Overridden by RTI Assistant				



8. Licensing Issues

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

8.1. Configuring a Federate To Allow the rtiexec Check Out a License	82
8.2. Unlicensed Federations	82
8.3. Troubleshooting License Problems	83
8.3.1 Specifying a Default License Server	84
8.3.2 Displaying Default License Server Settings	84
8.3.3 Clearing Default License Server Settings	85

8.1. Configuring a Federate To Allow 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, see "2.9. Managing Licenses for the MAK RTI" on page 28.

 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 these 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 context menu displays.
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 3 on page 29).
7. Select the Disable Unlicensed for Two check box.
8. Click OK.

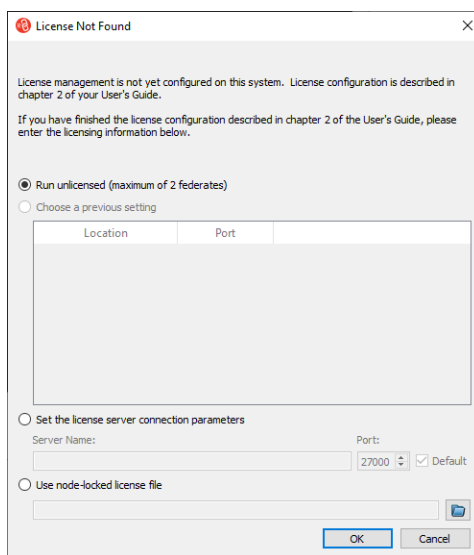
8.3. Troubleshooting License Problems

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 34).

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 34. License Not Found dialog box



► **To resolve license issues:**

1. In the License Not Found dialog box, select one of the 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.

i 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:**

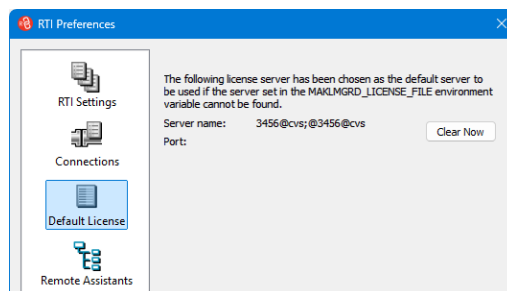
1. In the License Not Found dialog box, select the Use This Setting Whenever a License Cannot Be Found check box.
2. Click OK.

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 context menu displays.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Default License page. The license server settings display (Figure 35).

Figure 35. 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 context menu displays.
2. Choose **Preferences**. The RTI Preferences dialog box opens.
3. Select the Default License page. The license server settings displays (Figure 35 on the previous page).
4. Click Clear Now.
5. Click OK.



MAK Products Glossary

This glossary defines terms used in MAK ONE product documentation.

A**absolute dead reckoning**

A method of dead-reckoning in which an application uses the timestamps contained within state updates if the sender is using absolute timestamping. 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 wheel, turret, or gun.

attached part

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

B**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.

C**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 (does not display) 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.

D**Data Distribution Management**

The set of High-Level Architecture 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. See also absolute dead reckoning and relative dead reckoning.

DDM

See 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

See 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.

E**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 ONE products are VR-Link applications.)

exercise ID

A numeric identification for a DIS simulation exercise.

F**FED file**

Federation Execution Data file. The Federation Execution Data is the subset of FOM data needed and read by the MAK Products Glossary. VR-Link applications also read the FED file.

federate

A connection to the MAK Products Glossary. Typically a single simulation application can be thought of as a federate.

federate protocol

A message-based RTI API utilizing Protocol Buffers encoding that enables federates to invoke RTI services using a simple TCP connection.

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 and HLA 4, defines additional modular data content for a federation execution, usually extensions to an existing Federation Object Model. (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.

 G**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 ONE 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.

H**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 Protocol Data Units are sent to the network regardless of whether 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.

I**IG**

image generator

interaction

A message describing a simulation event. An interaction describes an event, it does not update an objects state.

L**Logger control PDUs**

A set of DIS PDUs or HLA interactions that let you control the MAK Data 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 a Run-Time Infrastructure. A federate communicates with the HLA network through the LRC.

LRC

See Local RTI Component.

M**MÄK Technologies Lisp**

An adaptation of the Lisp language used in configuration files for MAK ONE 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 High-Level Architecture standard MOM. It is a subset of the Federation Object Model 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 Run-Time Infrastructure. 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 MÄK Technologies Lisp.

N**NETN FOM**

The NETN FOM (NATO Education and Training Network Distributed Simulation Federation Object Model) is an extension of the RPR FOM 2.0, allowing other federates using pure RPR FOM to interoperate with applications using the NETN FOM.

O**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 Run-Time Infrastructure 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 Run-Time Infrastructure, 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 ONE 3D applications, the point of view into the scene. (Called the eyepoint in MAK Stealth 6.x and earlier.)

P**PDU**

See Protocol Data Unit.

Protocol Data Unit

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

R**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.

Real-time Platform Reference FOM

An HLA reference Federation Object Model based on the DIS protocol.

recording

A MAK 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 Federation Object Model 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. Contrast with absolute dead reckoning.

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

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

RPR FOM

See Real-time 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.

S**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.

T**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 heartbeat) 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.

U**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.

V

view control messages

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



Index

-

-A command-line option, 45

A

adding

- configuration to RTI Chooser, 70
- remote assistant connection, 63
- RTI connection configuration, 75
- rtiexec connection configuration, 73

address

- broadcast, 43

assistant

- remote
 - connecting, 63

-

--autoExit command-line option, 45

B

broadcast address, 43

C

callstack.log file, 23

checkout, license, 82

Choose RTI Connection dialog box

- starting rtiexec from, 47

choosing RTI connection for federate, 40

CLASSPATH, 26

clearing

- default license server, 85

closing

- RTI Assistant, 65

color

- federation, 59

color coding

- connection, 41

command-line option

- , 47
- A, 45
- autoExit, 45
- D, 45
- d, 46
- destAddrString, 45
- distributedForwarderPort, 45
- doNotUseRoutesFile, 46
- f, 46
- F, 46
- forceFullCompliance, 46
- forwarderToConnectTo, 46
- h, 47
- help, 47
- i, 47
- ignore_rest, 47
- K, 45
- l, 46
- locale, 34
- M, 46
- manual, 46, 73
- n, 46
- N, 47
- notifyLevel, 46

- overriding
 - connection options, 44
 - P, 47
 - p, 47
 - q, 46
 - R, 46
 - r, 47
 - rtiexec, 45
 - setLogFileName, 46
 - setNotifyQuiet, 46
 - setRidFileName, 46
 - setTcpPort, 46
 - setUdpPort, 47
 - T, 46
 - t, 47
 - tcpAddress, 47
 - tcpNetworkInterfaceAddr, 47
 - udpNetworkInterfaceAddr, 47
 - usePreconfiguredRid, 47
 - useReliable, 47
 - v, 47
 - version, 47
 - command-line option, ignore rest, 47
 - compliant mode
 - specifying, 46
 - configuration
 - adding to RTI Chooser, 70
 - connection, 43
 - editing in RTI Chooser, 71
 - configuration file, 38
 - configuring
 - connection, 43
 - Minidumper, 62
 - RTI Assistant, 60
 - startup, 61
 - RTI version and RID file, 69
 - rtiexec connection, 72
 - system, 68
 - connection
 - color coding, 41
 - failed, 42, 47
 - lightweight, 41
 - remote assistant, 63, 65
 - removing, 64
 - rtiexec, configuring, 72
 - connection configuration, 43
 - adding rtiexec, 73
 - lightweight, adding, 75
 - overriding with RID file, 73
 - removing, 78
 - rtiexec, adding, 75
 - crash
 - dump file, 62
 - RTI, 62
 - crash, application, 24
 - custom
 - logical time implementation, 27
- 
- D command-line option, 45
- 
- d command-line option, 46
- 
- data distribution management, See also DDM
 - definition of, 37
 - DDM, 37
 - debugging
 - callstack.log file, 23
 - dmp file, 23
 - file location, 24
 - declaration management
 - definition of, 37
 - default
 - license
 - clearing, 85
 - license server, 84
 - Defense Modeling and Simulation Office, 16
 - deleting
 - connection configuration, 78
 - federate, 50
- 
- destAddrString command-line option, 45
 - destroying
 - federation, 54
 - diagnostic
 - level, 46
 - message, 46
 - dialog box
 - License Not Found, 84
 - directory
 - Windows default install, 22
 - disabling
 - unlicensed mode, 82
 - disconnected federate
 - removing, 55
 - disconnecting
 - remote assistant, 65

distributed forwarder, 17



--distributedForwarderPort command-line option, 45

DLL

search path, 68

DM. See declaration management, 37

dmp file

debugging, 23

location, 24

DMSO RTI

compatibility with, 68

RTI_BUILD_TYPE environment variable, 69

RTI_HOME environment variable, 69

documentation

help, 9



--doNotUseRoutesFile command-line option, 46

dump file, 62

dynamic library search path, 68

dynamic linking, 18



editing

configuration in RTI Chooser, 71

environment variable, 24, 68

MAK_INSTALL_DIR, 22

MAK_LOG_DIR, 24

MAK_RTIDIR, 68

MAKLMGRD_LICENSE_FILE, 30

path, 68

PATH, 70

required, 22

RTI_ASSIS-TANT_PORT, 78

RTI_BUILD_TYPE, 69

RTI_CONFIG, 68

RTI_HOME, 69

RTI_RID_FILE, 68

executable

installation location

Windows, 22

exiting

RTI Assistant, 62, 65

rtiexec, 45



-f command-line option, 46



-F command-line option, 46



failed connection, 42

FDD file, 38

path to, 68

FED file, 38

path to, 68

federate, 58

choosing RTI connection, 40

disconnected, removing, 55

failed connections, 47

icon, specifying, 60

license, 82

resigning, 50

shutting down, 54

stale, 55

starting, 40

synchronizing, 37

troubleshooting connection problems, 42

unlicensed, 20, 82

federation, 58

color, 59

destroying, 54

management, 36

unlicensed, 82

Federation Object Model. See FOM, 36, 38

Federations View, 54, 58

federate icon, 60

federation color, 59

fedtime

installing Java, 26

file, configuration, 38

FLEXlm, 28-29

FOM, 36, 38

FOM Document Data, 38



--forceFullCompliance command-line option, 46

forwarder, starting, 50



--forwarderToConnectTo command-line option, 46
free mode, 20



GUI
localizing, 31



help
HTML5, 9
High Level Architecture, See also HLA
introduction, 16, 36
HLA
Interface Specification, 36
introduction, 16, 36
specification, 19
hostname
license server, 30



-i command-line option, 47



icon
federate, specifying, 60
RTI Assistant, 58
IEEE 1516, 19
installation, 21
default directory, Windows, 22
Linux, 22
Windows, 22
installing
HLA 1516 Java fedtime library, 26
Java bindings, 26
IP address, 45



Java
binding, installing, 26
fedtime library, installing, 26
jvm.dll, 26



-K command-line option, 45



-l command-line option, 46



LAN, connecting to remote assistant, 63
language, specifying, 34
latency testing, 58
LD_LIBRARY_PATH, 26
library, RTI, 17
license
checking out, 82
failure to find, 83
server and port, 83
unlicensed mode, 20
License Manager, 28
license manager, 29
License Not Found dialog box, 84
license server
default, 84
displaying default setting, 84
hostname, 30
lightweight connection, 41
lightweight mode, 18
unlicensed federation, 82
link-compatibility, 18
linking, dynamic, 18
Linux
product installation, 22
Local Component Notification History window, 61
Local RTI Component, 16-17, 38
locale, specifying, 34
localization
MAK RTI GUI, 31
specifying, 34
log file
callstack, 23
location, 24
notification level, 46
logical time
custom implementation, 27
LogicalTimeFactory, 27
LRC, 17, 19, 38
messages, 61



-M command-line option, 46



MAK_INSTALL_DIR environment variable, 22
 MAK_RTIDIR environment variable, 68
 MAKLMGRD_LICENSE_FILE environment variable,
 30



--manual command-line option, 46, 73
 message
 diagnostic, 46
 LRC, 61
 notification, 61
 notification level, 46
 reliable, 47
 Minidumper
 configuring, 62
 Modeling and Simulation Coordination Office, 16
 multicast
 address, 40
 discovery
 port, 78
 multicast group
 address, 45



-N command-line option, 47



network traffic, 37
 notification level
 setting, 46



object management
 definition of, 37
 ownership management
 definition of, 37



-P command-line option, 47



-p command-line option, 47



parameter
 RTI_configureConnectionWithRid, 46, 73
 RTI_destAddrString, 45, 73
 RTI_distributedForwarderPort, 73, 78
 RTI_forceFullCompliance, 73
 RTI_forwarderRoutesFile, 73
 RTI_mcastDiscoveryEnabled, 73
 RTI_mcastDiscoveryPort, 78
 RTI_networkInterfaceAddr, 73
 RTI_tcpForwarderAddr, 73
 RTI_tcpPort, 46, 73, 78
 RTI_udpPort, 47, 73, 78
 RTI_useRtiExec, 73

path
 default Windows install, 22
 environment variable, 68
 FED file, 68
 RID file, 68
 search, 68

PATH, 26
 PATH environment variable, 70
 PDB files

 Linux, 23
 Windows, 23

port, 40, 43
 default values, 78
 license server, 83
 RTI Assistant, 78
 RTI Forwarder, 78
 setting for TCP, 46
 specifying, 47

product installation
 Linux, 22
 Windows, 22



-q command-line option, 46

Q

Qt Linguist, 31

-

-R command-line option, 46

-

-r command-line option, 47

R

reliable transport, 47, See also TCP

remote assistant

- connecting, 63
- connecting and disconnecting, 65
- removing connection, 64

remote rtiexec, 43

removing

- connection configuration, 78
- federate, 50
- remote assistant connection, 64

resigning a federate, 50

RID file, 38, 40

- adding configuration to RTI Chooser, 70
- editing configuration in RTI Chooser, 71
- overriding
 - connection configuration, 73
- overriding connection parameters, 44
- selecting, 68
- specifying, 69
 - path, 68

rid.mtl, 38

routes file, 46

routes.mtl, See also routes file

RTI

- crash, 62
- definition, 16
- executive, 17
- introduction, 36
- library, 17
- services, 36
- settings, 38, 40
- specification, 19
- specifying version to use, 69
- version
 - selecting, 68

RTI Assistant, 17, 19

- adding connection configurations, 75
- clearing
 - default license, 85
- configuring, 60
- connecting remote, 63
- default license server, 84
- default license setting, 28, 82, 84
- exit policy, 62
- icon, 58
- port, 78
- remote connection
 - adding, 63
- shutting down, 54, 65
- starting, 58
- startup
 - configuring, 61

RTI Chooser, 17, 68-69

- adding configuration, 70
- editing configuration, 71

RTI connection

- choosing for federate, 40

RTI Executive, 17

RTI Forwarder, 19

- port, 78

RTI_ASSIS-TANT_PORT environment variable, 78

RTI_BUILD_TYPE environment variable, 69

RTI_CONFIG environment variable, 68

RTI_configureConnectionWithRid parameter, 46, 73

RTI_destAddrString parameter, 45, 73

RTI_distributedForwarderPort parameter, 73, 78

RTI_forceFullCompliance parameter, 73

RTI_forwarderRoutesFile parameter, 73

RTI_HOME environment variable, 69

RTI_JAVA_TIME_CLASS, 27

RTI_JAVA_TIME_INTERVAL_CLASS, 27

RTI_mcastDiscoveryEnabled parameter, 73

RTI_mcastDiscoveryPort parameter, 78

RTI_networkInterfaceAddr parameter, 73

RTI_RID_FILE environment variable, 68

RTI_tcpForwarderAddr parameter, 73

RTI_tcpPort parameter, 46, 73, 78

RTI_udpPort parameter, 47, 73, 78

RTI_useRtiExec parameter, 73

rtiexec, 17, 19

- automatic exit, 45
- checking out license, 82
- command-line options, 45
- connection configuration
 - adding, 73
- destroying federation, 54
- notification messages, 61

- remove disconnected federate, 55
- resigning federate, 50
- shutting down, 54
- starting, 43
 - remote computer, 43
- starting from Choose RTI Connection dialog box, 47
- rtiexec connection, 41
- RTIspy diagnostic GUI, 19, 58
- Run-Time Infrastructure, 36

S

- search order, 68
- selecting
 - RID file, 68
 - rti connection for federate, 40
 - RTI version, 68
- service
 - data distribution management, 37
 - declaration management, 37
 - federation management, 36
 - object management, 37
 - ownership management, 37
 - RTI, 36
 - time management, 37

-

- setLogFileName command-line option, 46

-

- setNotifyQuiet command-line option, 46

-

- setRidFileName command-line option, 46

-

- setTcpPort command-line option, 46
- settings, RTI, 38, 40

-

- setUdpPort command-line option, 47
- Shut Down All, 54
- shutting down
 - RTI Assistant, 65

- RTI components, 54
- SISO DLC HLA API 1516, 18
- SISO-STD-004.1-2004, 18
- specification
 - obtaining HLA and RTI, 19
- specifying
 - compliant mode, 46
 - language, 34
 - notification level, 46
 - path
 - FED and FDD files, 68
 - RID file, 68
 - port, 47
 - port and broadcast address, 43
 - RTI version and RID file, 69
- stack trace, 62
- Start rtiexec button, 48
- starting
 - federate, 40
 - rtiexec, 43
 - remote, 43
- startup, RTI Assistant configuration, 61
- synchronizing federates, 37
- system tray, 58

-

- T command-line option, 46

-

- t command-line option, 47

T

- TCP, See also reliable transport
 - setting port, 46

-

- tcpAddress command-line option, 47

-

- tcpNetworkInterfaceAddr command-line option, 47
- testing latency, 58
- time
 - custom logical, 27

- time management
 - definition of, 37
- timestamp order, 37
- translation, MAK RTI GUI text, 31
- trial mode, 20
- troubleshooting
 - license problems, 83

U

- UDP port, 78

-

- udpNetworkInterfaceAddr command-line option,
47
- unlicensed
 - federation, 82
- unlicensed mode, 20, 83
 - disabling, 82

-

- usePreconfiguredRid command-line option, 47

-

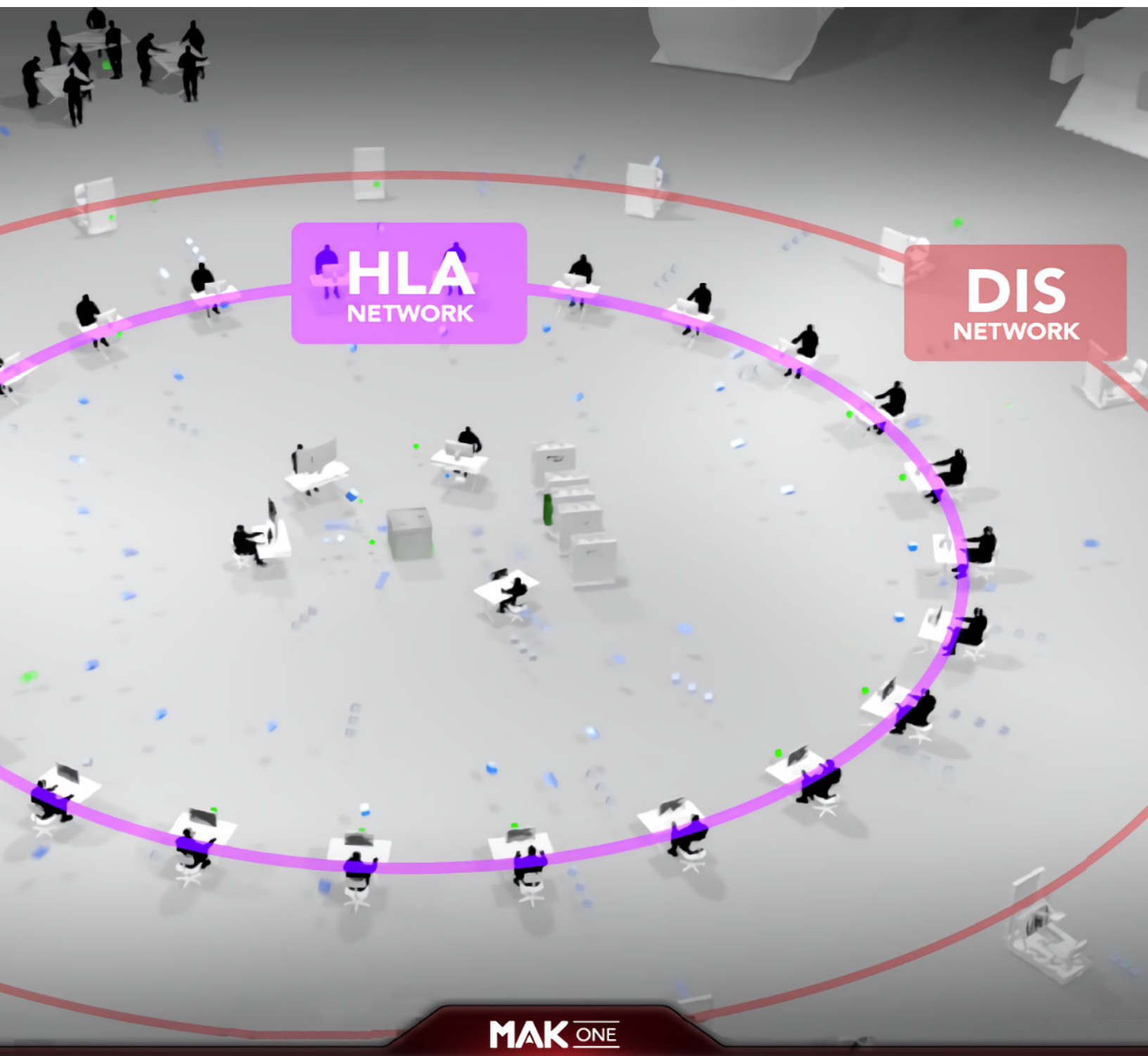
- useReliable command-line option, 47

V

- variable
 - environment, 68
- version
 - displaying, 47
 - specifying RTI, 69

W

- WAN
 - connecting to remote assistant, 63
- WAN connection
 - connecting and disconnecting, 65
- Windows
 - default install directory, 22
 - product installation, 22



RTI-5.0.1-02-251125