



Copyright © 2023 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®, RTIspey®, 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.

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, see "Third-Party Licenses" on page 38 near the end of this booklet.

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 MAK-0-1-230614



Contents

1. MAK Product Interoperability

1.1. The MAK Technologies Product Line	7
1.2. Basic Requirements for Running Any MAK ONE Product	9
1.2.1 Running the License Server	9
1.3. Running HLA Applications	9
1.4. Running DIS Applications	10
1.5. General Issues for Product Compatibility	11
1.5.1 Operating System Compatibility	11
1.5.2 Development Environment Compatibility	11
1.5.3 Third-Party API Compatibility	11
1.5.4 License Management Compatibility	12
1.6. Intra-Product Compatibility	12
1.6.1 VR-Link	12
1.6.2 MAK RTI	12
1.6.3 VR-Forces	12
1.6.4 VR-Engage	13
1.6.5 VR-Vantage	13
1.6.6 MAK Data Logger	13
1.6.7 VR-Exchange	14
1.6.8 VR-TheWorld	14
1.7. Inter-Product Compatibility	14
1.7.1 VR-Engage	15
1.7.2 Recording VR-Forces Simulations with the MAK Data Logger	15
1.7.3 Interoperability Between HLA 1.3 and IEEE 1516 Federates	16
1.7.4 FED File Naming Conventions	16
1.7.5 Compatibility Tables	19
1.8. Synchronizing Terrain for the VR-Vantage and VR-Forces	19
1.8.1 Flat Earth Databases	20

1.8.2 Unsupported Features	20
1.8.3 Database is not WGS84	20
1.8.4 Optimized Databases	20
1.9. Delivering Applications to Users	20

2. Default Ports for MAK Products

2.1. MAK Legion	23
2.2. MAK Data Logger	23
2.3. MAK RTI	23
2.4. VR-Link	23
2.5. VR-Exchange	24
2.6. WebLVC	24
2.7. VR-Forces	24
2.8. VR-Engage	25
2.9. RadarFX	25
2.10. FLEXIm Licensing	25
2.11. Loopback Connections	25

3. Basic Troubleshooting

3.1. Product Does Not Run	27
3.1.1 Linux Applications	27
3.1.2 HLA Applications	27
3.1.3 License Management Checklist	28
3.2. MAK ONE Products Do Not Work Together	30
3.2.1 DIS Applications	30
3.2.2 HLA Applications	32
How to Contact Us	35
Telephone	35
E-mail	35
Internet	35
Post	35

4. Conventions and Third-Party Licenses

Document Conventions	37
DI-Guy Conventions	37
Mouse Button Naming Conventions	38
Third-Party Licenses	38
Third-Party Libraries Supported	38
Library Licensing	39
Boost License	41
libXML and libICONV	42

Lua 42

LizardTech 42

Freefont OpenType Font Set 42

Scintilla and SciTE 43

Autodesk Gameware Navigation 43

NVIDIA 43

Third-Party Licenses for VR-Vantage Applications 44

Index 46



1. MAK Product Interoperability

This handbook provides a high-level introduction to using MAK ONE products in DIS and HLA exercises. MAK ONE products are fully interoperable with other applications that adhere to the DIS or HLA standards. In order to interoperate, however, you must properly configure your applications. This handbook describes the basic requirements to run our products using these simulation standards and discusses compatibility issues among different versions of MAK ONE products. We hope that it will help both new and experienced customers to set up their simulation environments. Chapter 3 provides basic troubleshooting procedures.

For specific installation and configuration instructions, please see the documentation for your particular MAK ONE products. You can also contact support at <https://www.mak.com/mak-one/support/portal>.

1.1. The MAK Technologies Product Line	7
1.2. Basic Requirements for Running Any MAK ONE Product	9
1.2.1 Running the License Server	9
1.3. Running HLA Applications	9
1.4. Running DIS Applications	10
1.5. General Issues for Product Compatibility	11
1.5.1 Operating System Compatibility	11
1.5.2 Development Environment Compatibility	11
1.5.3 Third-Party API Compatibility	11
1.5.4 License Management Compatibility	12
1.6. Intra-Product Compatibility	12
1.6.1 VR-Link	12
1.6.2 MAK RTI	12
1.6.3 VR-Forces	12
1.6.4 VR-Engage	13
1.6.5 VR-Vantage	13
1.6.6 MAK Data Logger	13
1.6.7 VR-Exchange	14
1.6.8 VR-TheWorld	14

1.7. Inter-Product Compatibility	14
1.7.1 VR-Engage	15
1.7.2 Recording VR-Forces Simulations with the MAK Data Logger	15
1.7.3 Interoperability Between HLA 1.3 and IEEE 1516 Federates	16
1.7.4 FED File Naming Conventions	16
1.7.5 Compatibility Tables	19
1.8. Synchronizing Terrain for the VR-Vantage and VR-Forces	19
1.8.1 Flat Earth Databases	20
1.8.2 Unsupported Features	20
1.8.3 Database is not WGS84	20
1.8.4 Optimized Databases	20
1.9. Delivering Applications to Users	20

1.1. The MAK Technologies Product Line

The MAK Technologies line of software products is designed to streamline the process of developing and using networked simulated environments.

The MAK Technologies product line includes the following:

- **VR-Link® Network Toolkit.** VR-Link is an object-oriented library of C++ functions and definitions that implement the High Level Architecture (HLA) and the Distributed Interactive Simulation (DIS) protocol. VR-Link has built-in support for the 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, RTIsPy®, that allows you to extend the RTI using plug-in modules. It also has a graphical user interface (the RTI Assistant) that helps users with configuration tasks and managing federates and federations.
- **VR-Forces®.** VR-Forces is a computer generated forces application and toolkit. It provides an application with a GUI, that gives you 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™ Server. VR-TheWorld Server is a simple, yet powerful, web-based streaming terrain server, developed in conjunction with Pelican Mapping. Delivered with a global base map, you can also easily populate it with your own custom source data through a web-based interface. The server can be deployed on private, classified networks to provide streaming terrain data to a variety of simulation and visualization applications behind your firewall.
- DI-Guy™. 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.

- RadarFX. RadarFX is a client-server application that simulates synthetic-aperture radar (SAR). The server application, which is based on VR-Vantage and SensorFX, loads a terrain database and, optionally, connects to simulations. A client application requests SAR images from the server. RadarFX includes a sample client application.
- VR-Engage. VR-Engage is a multi-role virtual simulator that lets users play the role of a first person human character, a ground vehicle driver, gunner or commander, 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.

1.2. Basic Requirements for Running Any MAK ONE Product

To run any MAK ONE product, you must have a license for that product, you must properly configure all computers on which you are running MAK ONE software, and the license server must be running. Please see the installation instructions for your MAK ONE software for complete details about setting up the license management environment.

1.2.1 Running the License Server

Although product licenses are specific to operating systems, the license server itself can run on any operating system supported by FLEXlm and for which MAK Technologies provides a *maklmgrd* executable (currently Windows and Linux). In other words, if you have a license to run the Linux version of the MAK Data Logger, you could run the license server on a Windows PC.



All computers on which you are running MAK software must have a network connection to the license server, unless you are using a dongle.

1.3. Running HLA Applications

To run the HLA versions of MAK ONE products, you must also install and configure an RTI. Except as noted below, all MAK ONE products are link compatible with the MAK RTI and other RTIs that support the HLA 1.3 specification or the HLA Evolved (IEEE 1516-2010) specification, and which are built with the same compilers as the MAK ONE product you are using.

If you observe the following points, you should have no problem running HLA applications:

- You must install an RTI on each computer that is running an HLA federate. Federates must be able to find the RTI libraries (.dll or .so.) You usually accomplish this on Windows by adding the path to the RTI's ./bin (or ./bin64) directory to the PATH environment variable. On Linux, you add the path to the RTI's lib directory LD-LIBRARY_PATH environment variable.
- If you are using the MAK RTI, you must have an RTI license for each federate that is participating in the federation.
- All federates in a federation must use the same RTI, configured in the same way, and using the same FED file (FDD file in HLA 1516), and each federate must be able to find the FED file. The easiest way to ensure that a MAK ONE product can find the FED file is to put it in the directory from which you are running the application.
- Some RTIs require that you run an *rtiexec*, but only run one instance of it.
- If you are using the MAK RTI, you can run in lightweight mode without an *rtiexec*. If you want to use advanced features such as DDM, Time Management, or MOM, you must run the *rtiexec*. Run only one instance of the *rtiexec* per federation. Always run different federations on different UDP ports.
- Most RTIs rely on an external configuration file (*rid.mtl* for the MAK RTI.) When you set up your federates, you must ensure that each federate has access to the configuration file.



When using HLA Evolved:

- VR-Link 4.0, 4.0.1, and 4.0.2 are only compatible with MAK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.
- VR-Link 4.0.3 and later releases are only compatible with MAK RTI 4.0.4 and later releases.
- VR-Exchange 2.0 and MAK Data Logger 5.0 are only compatible with MAK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.
- VR-Exchange 2.0.1 and later are only compatible with MAK RTI 4.0.4 and later releases.
- VR-Vantage 1.4 and later and VR-Forces 4.0.3 and later are only compatible with MAK RTI 4.0.4 and later releases.

1.4. Running DIS Applications

To run the DIS version of a MAK ONE product out-of-the-box, you do not have to do any special configuration or start any supporting applications (except, of course, the license server.) By default, MAK ONE products listen to UDP broadcast traffic on port 3000. If you do not want your DIS application to interact with other DIS applications on your LAN, change the port you are using. You can specify the port in the application's configuration file or with the `-P` command-line option. (Most product manuals list command-line options in the chapter that explains how to start the product. In addition, VR-Vantage, Data Logger, and VR-Forces have quick reference cards bound inside the back cover that list command-line options.)

1.5. General Issues for Product Compatibility

The Product Specifications & Dependencies page on the MAK website at <https://www.mak.com/mak-one/support/help/specs> lists a variety of compatibility information for each release of a MAK ONE product. In this section, we summarize at a slightly higher level the factors that affect product compatibility. This should help you make sense of the product-specific listings.

1.5.1 Operating System Compatibility

Compatibility of MAK ONE products across different versions of supported operating systems varies. Windows versions of our products work on all supported versions of Windows, as listed in the Product Specifications & Dependencies page.

Due to changes in system libraries, Linux versions of MAK ONE products usually run only on the version of Linux on which they were compiled.

1.5.2 Development Environment Compatibility

Windows-based MAK ONE products and examples are built with Microsoft Visual C++. Depending on the product, there may be versions built with VC++ 14.0 or 15.0 compilers. (Older compilers are dropped over time.) Product documentation assumes that you are familiar with the use of your chosen compiler and development environment.



MAK provides versions of product releases that have been compiled with different versions of Microsoft Visual C++. MAK ONE products built with VC 14 and later are binary compatible and will work together. Products compiled with VC 12 and earlier can experience instability when used in conjunction with products built with different compiler versions. When you run MAK ONE products together on the same computer, for example, the Logger and VR-Vantage or VR-Forces, we strongly recommend that you run versions compiled with the same compiler or binary compatible compilers.

If you are running MAK ONE products on different machines, they do not have to be built with the same compiler. However, remember that if you are using the MAK RTI, you must use the same release on all computers (for example, on computer 1, MAK RTI 4.4 VC 14; on computer 2, MAK RTI 4.4 VC 12).

1.5.3 Third-Party API Compatibility

MAK Technologies uses various third-party APIs to build our products. For example, we use the Qt development toolkit for our graphical user interfaces. The *Preface* in the MAK ONE product's user documentation lists the third-party API requirements for that release. "Third-Party Licenses" on page 38 also lists the requirements for recent releases of MAK ONE products, but you should refer to the product documentation to verify what is supported in the version you are using.

Third-party APIs may be updated more frequently than MAK ONE products are updated. Also, in the interest of product stability and interoperability, MAK may not move to upgraded versions of third-party APIs when they become available.

Although you may be able to build MAK ONE products using more recent versions of third-party libraries, this depends on the third-party developer's efforts to maintain backward compatibility. MAK cannot guarantee that this will be possible. In general, when deciding to move to updated third-party libraries, we try to balance the benefits of new features against the inconvenience to customers that may result from porting to newer software.

1.5.4 License Management Compatibility

When MAK ONE products are built, they are linked to a version of the FLEXlm license management software. To run a MAK ONE product, you must install that either version of FLEXlm or a later release. Check the release notes for each release of a MAK ONE product to find out if there are any changes in the release of FLEXlm being used. You can also check the Product Specifications & Dependencies page on the MAK website, <https://www.mak.com/mak-one/support/help/specs>, to see which version of FLEXlm is required for a given release. As of March 2022, most products use FLEXlm 11.17.1.

License files are forward compatible, so you do not have to change your license files if you upgrade your FLEXlm software.

1.6. Intra-Product Compatibility

This section discusses backward and forward compatibility of MAK products and applications built using them.

1.6.1 VR-Link

Different versions of VR-Link maintain network compatibility as long as you use the same FOM.

VR-Link 5.0 renamed header files to conform to MAK file naming conventions.

If you are using HLA Evolved, see the restrictions in the note in "1.3. Running HLA Applications" on page 9.

1.6.2 MAK RTI

Federates built with the MAK RTI's HLA 1.3 libraries can interoperate with federates built with MAK RTI's HLA 1516 libraries if the federation is not using ownership management, DDM, or MOM services.

The MAK RTI is binary compatible. This means that you can update the MAK RTI without recompiling any MAK ONE products.

If you are using HLA Evolved, see the restrictions in the note in "1.3. Running HLA Applications" on page 9.

1.6.3 VR-Forces

The different versions of VR-Forces® and applications built using the VR-Forces toolkit are generally not compatible. This means, for example, that you cannot control

a VR-Forces 4.10 back-end with a VR-Forces 5.0.3 front-end. Applications built with different versions of VR-Forces can participate in the same exercise. However, they treat each other as remote simulations.

1.6.4 VR-Engage

Due to changes in the simulation model sets between releases, different releases of VR-Engage cannot interact with each other, except that they will see any entities simulated by the other version as remote entities.

1.6.5 VR-Vantage

There are no compatibility issues for using different releases of VR-Vantage applications as viewers in the same exercise, except for standard HLA compatibility issues. However, a VR-Vantage application can only control display engines from the same release. For example, a VR-Vantage 3.0.3 application could not control a VR-Vantage 2.8 display engine. However, applications and display engines built with different Windows compilers can interoperate.

1.6.6 MAK Data Logger

Since the Data Logger only receives data from the network or sends data over it, there are no compatibility issues for coexisting on the network with other versions.

Recordings made with any version of the Logger in the 3.x series can be played on any later version of the Logger. Recordings made with the HLA versions of Logger 3.9 or later cannot be played on versions 3.8.2-ngc or earlier. Recordings made with Logger 4.x or later cannot be played on a 3.x Logger. HLA recordings made using version 3.7 and earlier of the Logger (and possibly some later versions) cannot be played on Logger 4.x or later. If you open these older recordings in Logger 3.12 and save them, they will work in Logger 4.x or later. The Logger file format changed in Logger 5.3 and 5.4. If you open an older file, you are prompted to update to the new format, which is handled automatically.

Recordings are protocol specific. For example, the HLA Logger cannot play a recording made with the DIS Logger. If you need to use a recording made with one protocol with an application from the other, use the MAK Gateway to translate the recording's output to the other protocol.

Use of HLA Time Management during recording does not affect the actual recording process. A recording made with Time Management enabled can be played back with or without Time Management. Similarly, a recording made without Time Management can be played back with or without Time Management.

If you are using HLA Evolved, see the restrictions in the note under "1.3. Running HLA Applications" on page 9.

Recording HLA 1.3 Logger Files and Playing Back in HLA 1516 Federations

In general, the MAK Data Logger uses the same file format for HLA 1.3 and HLA 1516 Logger files. You can record a Logger file from an HLA 1.3 federation, and play it back into an HLA 1516 federation (and vice versa). However, you must use the same FOM representation during record and playback. For example, if you use an HLA 1.3-style FED file during recording, you must use an HLA 1.3-style FED file during playback,

even if you are using the IEEE 1516 version of the Logger and an IEEE 1516 RTI. (VR-Link and the MAK RTI allow you to use HLA 1.3-style FED files with IEEE 1516-based federates, and vice versa).

The main reason that consistency in FOM format is necessary is that IEEE 1516 uses different names for the "Root" classes of the Object and Interaction class hierarchies. A 1.3-style FED file requires a Root class called Object-Root, whereas a 1516-style XML file requires a Root class called HLAObjectRoot. The reason this can be a problem is that if the Logger is playing an HLA 1.3-based Logger file into a federation that is using a 1516-based XML file, it might come across an instance of a class called, for example, "ObjectRoot.Vehicle". If it tries to register an object of this class, the RTI will complain that no such class exists. There might be a class called HLAObjectRoot.Vehicle in the current FOM, but the RTI or Logger does not know that this is actually the same class.

In future versions of the Logger, we intend to rectify this situation and allow playback of FED-based Logger files in federations that use XML-based FOM representations (as long as the FOMs otherwise describe a consistent set of classes, attributes, and parameters).

1.6.7 VR-Exchange

Since VR-Exchange is only an executable, it has no version compatibility issues. Also, it should be less susceptible to library and compiler changes on Linux than other MAK ONE products. On Windows, it has the same compiler compatibility issues as described in "1.5.2 Development Environment Compatibility" on page 11. If you use a FOM Mapper with VR-Exchange, it must be built with the same version of VR-Link as the version of VR-Exchange that you are running it with.

If you are using HLA Evolved, see the restrictions in the note in "1.3. Running HLA Applications" on page 9.

1.6.8 VR-TheWorld

VR-TheWorld has no dependencies on other MAK ONE products. It does not use DIS or HLA. The URL for VR-Vantage users on the web interface requires VR-Vantage 1.2 or later. However, VR-Vantage 1.1, which supports terrain servers, can connect to VR-TheWorld.

1.7. Inter-Product Compatibility

This section discusses compatibility between different MAK ONE products.

In general, to build an application using one of our APIs, you must link with the appropriate version of VR-Link, and (for HLA applications) an RTI. You must be using a supported operating system with the correct compiler. These requirements are listed in the release notes for each product and on the Product Specifications & Dependencies page on the MAK website: <https://www.mak.com/mak-one/support/help/specs>.

On Windows, the correct versions of VR-Link libraries (.dll and .lib) are shipped with each MAK ONE product.

DIS versions of the Logger, VR-Vantage, and VR-Link applications should all be able to interact with each other and with other MAK ONE products without any difficulty. HLA versions should also be able to interact without problems assuming that the HLA environment is set up properly (same RTI version, same FED file.)

VR-Forces should interoperate with different DIS versions of the Logger, VR-Vantage, and VR-Link applications without difficulty. It should also interoperate with HLA versions, subject to the issues previously discussed.

Stealth control from VR-Vantage and VR-Forces uses Stealth Control PDUs. They should work with any version of the Stealth that supports them.



Stealth version 6.2 and earlier supports the full list of Stealth Control PDUs. Stealth 7.0 (part of VR-Vantage) and VR-Vantage Stealth support a limited set of these PDUs. The Stealth Control Toolkit has been superseded by the VR-Vantage Control Toolkit. Please see the *VR-Vantage Users Guide* for details.

Windows versions built with different versions of the C++ compiler may have problems running together on the same computer.

1.7.1 VR-Engage

A VR-Engage release is built against a specific version of VR-Forces. If you want a release of VR-Engage to interact with VR-Forces, that is to take control of entities being simulated by VR-Forces, they must be compatible releases. They must have the same SMSs and the VR-Engage add-ons for VR-Forces must be installed in VR-Forces. See the *VR-Engage Users Guide* documentation to learn which version of VR-Forces is compatible with your version of VR-Engage.

1.7.2 Recording VR-Forces Simulations with the MAK Data Logger

You can use the MAK Data Logger to record VR-Forces simulations and then play them back for review. If you want to view a Logger file in VR-Forces (as opposed to some other MAK viewer, such as VR-Vantage), note the following considerations:

- When you view the Logger file, you must run VR-Forces with an application number that is different from the one it used when you recorded the simulation. If you use the same application number, VR-Forces may crash. (We recommend that when you record the simulation, you run VR-Forces with an application number other than the default (or whatever application number you typically use). This reduces the likelihood of a conflict when you play back the Logger file. You will not have to remember to change the VR-Forces application number every time you start it.)
- Do not load the original scenario in VR-Forces while you view the recording. Just load the terrain database that the simulation requires.
- If you are using HLA 1516, do not record a simulation and then immediately play it back. Exit the applications and start a new federation. Trying to play back the file in the federation from which it was recorded can result in name reservation conflicts and applications may crash.

1.7.3 Interoperability Between HLA 1.3 and IEEE 1516 Federates

In general, the MAK RTI and various other MAK tools support run-time interoperability between HLA 1.3 federates and IEEE 1516 federates. Federates that are using Federation Management, Declaration Management, Object Management, Time Management, and Data Distribution Management services should be able to interoperate across the 1.3-1516 boundary. For example, you can typically run the HLA 1.3 version of the MAK Stealth and the IEEE 1516 version of VR-Forces together in the same federation execution, provided you are using the MAK RTI, or another RTI that supports 1.3-1516 interoperability.

However, note the following restrictions and requirements:

- MOM is not currently supported across the 1.3-1516 boundary and must be disabled. (The 1.3 and 1516 Specifications dictate completely different data representations for MOM attributes and parameters.)
- Ownership management is also problematic and should be avoided. For object names, the 1516 RTI converts all strings from the wide char representation to a narrow character representation. So, the names supplied by the 1516 federates must allow this conversion (that is, only support the ASCII character set). For user supplied tags, the 1516 federates are restricted to the 1.3 string format. The 1.3 RTI cannot represent the complex data types allowed by 1516.
- Although federates might differ in the version of the HLA they use (1.3 versus 1516), they must agree on a common FOM representation to use. All federates must use either an HLA-1.3-style FED file, or an IEEE-1516-style XML file. (VR-Link and the MAK RTI allow you to use HLA-1.3-style FED files with IEEE-1516-based federates, and vice versa).

The main reason that consistency in FOM format is necessary is that HLA 1.3 and IEEE 1516 use different names for the Root classes of the Object and Interaction class hierarchies. A 1.3-style FED file requires a Root class called *Object-Root*, whereas a 1516-style XML files requires a Root class called *HLAObjectRoot*. This is a problem because if a federate uses an HLA-1.3-based FED file, it might subscribe to a class called *ObjectRoot.Vehicle*. On the other hand, a federate that is using an IEEE-1516-style XML file might publish a class that is meant to be the same class, but that is actually named *HLAObjectRoot.Vehicle*. Neither the RTI nor the federates will realize that these classes were intended to be the same, and the subscribing federate will fail to discover any objects that the publishing federate registers.

1.7.4 FED File Naming Conventions

Prior to version 5.2, VR-Link named FED files using the convention VR-Link<RPR FOM version>.fed or VR-Link<RPR FOM version>.xml. For example, the FED file for RPR FOM 2, draft 17 was *VR-Link20017-1.fed*. This naming scheme had two problems. First, it was confusing for end-users of applications that use VR-Link (but who are not themselves VR-Link customers) to be using a FOM named that way. Second, the filename was not versioned, so any updates that we would make to the existing FOMs could cause compatibility problems. In fact, for this latter reason, we had refrained from fixing a few minor bugs that we found over the years until now.

Effective with VR-Link 5.2, VR-Link introduced a new naming convention: MAK-<FOM name><FOM revision><FOM version>.fed (or .xml, or _evolved.xml, as appropriate). For example, *VR-Link20017-1.fed* was renamed *MAK-RPR20017-1-1.fed*.

Additionally, given the power of HLA Evolved FOM Modules, starting with RPR 2.0 we are providing the RPR standard FOM for HLA Evolved, currently named RPR_FOM_v2.0_1516-2010. MAK's extensions are in FOM Modules that our software automatically adds to an exercise. This means that if you have another RPR-based FOM, such as NETN or NASMP, you will have a much easier time using it without needing to modify the FOM.

VR-Link continues to include FED files and XML files using the old convention, so you do not have to change any existing scripts or code. At some point, the old FED files will be deprecated.

The default startup configurations for MAK ONE products built against VR-Link 5.2 or later use the new naming conventions. Therefore, if you want a newer release of a product to interoperate with one using the old naming conventions, you must change the connection configurations so that your applications all use the same FED files and federation names.

Table 1 lists legacy FED files and how they map to new FED files. Many of the FED files delivered in previous releases do not have updated versions, usually because they are rarely used.

Table 1. Old FED file to new FED file mappings

Old FOM	New FOM
HLA 1.3	
DI-Guy-13_o.fed	
DI-Guy.fed	
VR-Link.fed	MAK-RPR1-1-1.fed
VR-Link05.fed	
VR-Link07.fed	
VR-Link08.fed	
VR-Link10.fed	MAK-RPR1-1-1.fed
VR-Link16.fed	
VR-Link20006.fed	
VR-Link20014.fed	
VR-Link20017-1.fed	MAK-RPR20017-1-1.fed
VR-Link20017-6.fed	MAK-RPR20017-6-1.fed
VR-Link20017.fed	
VR-Link20020.fed	MAK-RPR2-1-1.fed
VR-LinkBOML16.fed	
VR-LinkDDM.fed	

Table 1. Old FED file to new FED file mappings (continued)

Old FOM	New FOM
VR-LinkMCo2.fed	MAK-RPR1-MCo2-1-1.fed
VR-LinkTM.fed	MAK-RPR1-TimeManagement-1-1.fed
VRL-20017-1-Link16.fed	
VRL-DIGuy20017.fed	
VRL-DIGuy20017DDM.fed	
1516-2000	
DI-Guy-13_o.xml	
DI-Guy.xml	
VR-Link.xml	MAK-RPR1-1-1.xml
VR-Link20014.xml	
VR-Link20017-1.xml	MAK-RPR20017-1-1.xml
VR-Link20017-6.xml	MAK-RPR20017-6-1.xml
VR-Link20020.xml	MAK-RPR2-1-1.xml
VR-LinkBOML16.xml	
VR-LinkDDM.xml	
VR-LinkTM.xml	MAK-RPR1-TimeManagement-1-1.xml
VRL-DIGuy20017.xml	
VRL-DIGuy20017DDM.xml	
HLA Evolved	
DIGuyModule_1300.xml	MAK-DIGuy-1_evolved.xml
DIGuyModule_1302.xml	
RPR_FOM_v2.o_draft20_1516-2010.xml	RPR_FOM_v2.o_1516-2010.xml
VR-Link20014_base_evolved.xml	
VR-Link20014_evolved.xml	
VR-Link20017-1_base_evolved.xml	MAK-RPR20017-Base-1-1_evolved.xml
VR-Link20017-1_base_evolved_withUpdateRate.xml	
VR-Link20017-1_evolved.xml	MAK-RPR20017-1-1_evolved.xml
VR-Link20017-1_evolved_withUpdateRate.xml	
VR-Link20017-6_base_evolved.xml	MAK-RPR20017-Base-6-1_evolved.xml
VR-Link20017-6_evolved.xml	MAK-RPR20017-6-1_evolved.xml

Table 1. Old FED file to new FED file mappings (continued)

Old FOM	New FOM
VR-LinkDDM_base_evolved.xml	
VR-LinkDDM_evolved.xml	
VR-Link_LgrControl_evolved.xml	MAK-LgrControl-1_evolved.xml
VR-Link_ViewControl_evolved.xml	
VR-Link_VrlExt_evolved.xml	MAK-VRFExt-1_evolved.xml
VR-Link_base_evolved.xml	MAK-RPR1-Base-1-1_evolved.xml
VR-Link_evolved.xml	MAK-RPR1-1-1_evolved.xml
VRL-DIGuy20017DDM_evolved.xml	
VRL-DIGuy20017_evolved.xml	
updateRate.xml	MAK-updateRate-1_evolved.xml
VR-LinkBOML16_module.xml	MAK-BOML16-1_evolved.xml
	MAK-VRFAggregate-1_evolved.xml

1.7.5 Compatibility Tables

To see the compatibility table that cross-references all MAK ONE products, showing which ones are compatible, please visit the Build Compatibility page on the MAK Technologies website: <https://www.mak.com/mak-one/support/help/build-compatibility>.

1.8. Synchronizing Terrain for the VR-Vantage and VR-Forces

Many MAK Technologies customers create scenarios in VR-Forces and view them in one of the VR-Vantage applications. The terrain databases used by the two applications must be synchronized so that the 3D view matches the 2D view and the simulation engine.

VR-Forces 4.0 and later use the same terrain libraries as VR-Vantage. Therefore, if the VR-Forces front-end and back-end use the same terrain as VR-Vantage, simulation and visualization are fully synchronized. However, there may be occasions in which you use different terrains in the front-end and back-end, in which case, the synchronization issues discussed in this section would still apply.

If you are using VR-Forces 3.x, in many cases, you can open an Open Flight (FLT) file in VR-Vantage and VR-Forces and the views are synchronized without any further action on your part. In VR-Forces 3D Front End, version 2.0, you can compose a terrain database and save it in the MTF format for use with MAK 3D applications (3D Front End and VR-Vantage applications) and in GDB format for MAK 2D applications.

Terrain problems may occur under the circumstances described in the following sections.

1.8.1 Flat Earth Databases

When VR-Forces 3.x imports a Flat Earth database, it converts it to UTM; VR-Vantage loads it as-is. If the database is larger than one UTM zone, discrepancies between the VR-Vantage view and VR-Forces will occur in areas outside the UTM zone.

1.8.2 Unsupported Features

VR-Vantage supports virtually all FLT features. VR-Forces 3.x does not support all FLT features. If a database has unsupported features, VR-Forces may not be able to load it.

1.8.3 Database is not WGS84

VR-Vantage assumes that all databases use WGS84 datum. It does not check the database header. VR-Forces 3.x checks the database header and responds appropriately. If you are using a database that does not use WGS84 datum, the best option is to convert it to WGS84. If you cannot convert the database, setting it to WGS84 will provide correlation between VR-Vantage and VR-Forces, however the coordinates sent over the network will be incorrect. For assistance, please contact support at <https://www.mak.com/mak-one/support/portal>.



Any time you use non WGS84 data with VR-Forces in DIS, the coordinates will be incorrect on the network because DIS expects WGS84 data.

1.8.4 Optimized Databases

Databases can be optimized for 3D use or for simulation. If a database is optimized for one, it may work less well in the other. For example, a large, complex database could be optimized for 3D viewing through the use of several levels of detail. While it might work very well in VR-Vantage, it could cause a problem for VR-Forces 3.x performance due to its size.

1.9. Delivering Applications to Users

A developer's license (for example, VR-Link or VR-Forces) allows you to compile applications using the product libraries and run one such application. After you have developed an application, you may need to deliver that application to an internal or external customer. The end user of the application needs a runtime license to use it. The customer does not need the entire toolkit, nor do they need a developer's license (unless they need to recompile the application.) They need only the application, the product libraries (if you link dynamically), the license management files, and a runtime license for each instance of the application that they will run. You should not provide the header files. If you statically link to the MAK libraries, you do not have to deliver them, either.

You can purchase a runtime license for your customer or have the customer purchase the license from MAK, depending on the arrangements you have made.



2. Default Ports for MAK Products

This chapter lists the default ports for MAK Products and their components.

2.1. MAK Legion	23
2.2. MAK Data Logger	23
2.3. MAK RTI	23
2.4. VR-Link	23
2.5. VR-Exchange	24
2.6. WebLVC	24
2.7. VR-Forces	24
2.8. VR-Engage	25
2.9. RadarFX	25
2.10. FLEXlm Licensing	25
2.11. Loopback Connections	25

2.1. MAK Legion

Port 6000 is the default port.

Table 2. MAK Legion Port Value

Default Port	To Use a Different Port...
6000	Change in command line or relevant config file (./bin64/legionMonitor.lua, ./bin64/legionServer.lua)

2.2. MAK Data Logger

Table 3. MAK Data Logger Port Values

Default Port	Protocol / Usage	To Use a Different Port...
3000	DIS	Change in GUI, command line, or config file
4053	UDP: Remote Control	Change in command line or config file
5000	UDP: Video Streaming	Change in GUI or config file

2.3. MAK RTI

The MAK RTI ports are Modified in the RTI GUI or in *rid.mtl*. This information applies only to HLA connection.

Table 4. MAK RTI Port Values

Default Port	Protocol / Usage	To Use a Different Port...
4000	UDP Port	Edit <i>rid.mtl</i> (setqb RTI_udpPort 4000)
4001	TCP Port	Edit <i>rid.mtl</i> (setqb RTI_tcpPort 4001)
5000	TCP: rtiForwarder	Edit <i>rid.mtl</i> (setqb RTI_distributedForwarderPort 5000)
6001	Multicast Discovery Port	RTI_mcastDiscoveryPort
6003	UDP: RTI Assistant Port	RTI_ASSISTANT_PORT (environment variable)

2.4. VR-Link

Port 3000 is the default for all DIS protocols. For HLA, see MAK RTI.

Table 5. VR-Link Port Values

Default Port	Protocol / Usage	To Use a Different Port...
3000	DIS	Change in command line

2.5. VR-Exchange

Table 6. VR-Exchange Port Values

Default Port	Protocol / Usage	To Use a Different Port...
3000	UDP: DIS Broker	Change in GUI, command line, or config file
5111	TCP: Brokers to heartbeat the portal	Change in GUI or config file

2.6. WebLVC

Table 7. WebLVC Port Values

Default Port	Protocol / Usage	To Use a Different Port...
8080	Port for Proxy	Change in GUI or config file
9101	Web Server Port	Change in GUI or config file
9102	TCP Server Port	Change in GUI or config file

2.7. VR-Forces

Table 8. VR-Forces Port Values

Default Port	Protocol / Usage	To Use a Different Port...
3000	DIS	Change in GUI or command line
9000	TCP: Transporting scenario files on load and save	Set <code>--fileTransporterReceivePort</code> on the front-end command line
Random	Port used by the DtCgfDispatcher's file transporter to receive order of battle and plan files from a DtVrfRemoteController. If the port requested cannot be opened, the next open port will be used. A value of zero indicates that VR-Link should choose an ephemeral port, which is automatically assigned by the system.	The only way to change the port is in the back-end. Edit: <code>cgfDispatcherReceivePort</code> in the <code>vrfSim.mtl</code> located in <code>./appData/settings/vrfSim</code> folder (setqb <code>cgfDispatcherReceivePort</code>)

2.8. VR-Engage

Table 9. VR-Engage Port Values

Default Port	Protocol / Usage	To Use a Different Port...
3000	DIS	Change in GUI or command line

2.9. RadarFX

Table 10. RadarFX Port Values

Default Port	Protocol / Usage	To Use a Different Port...
4567	Simple Client	Change in GUI
3000	DIS, RadarFX Server	Change in GUI

2.10. FLEXlm Licensing

Table 11. FLEXlm Licensing Port Value

Default Port	To Use a Different Port...
27000	Edit SERVER hostname MAC_Address 27500 (27500 is the defined port)
Defined @ runtime	Set by adding port=xxxx, for example: VENDOR maklmgrd ./maklmgrd port=1234

2.11. Loopback Connections

When using a loopback connection for DIS or Legion, all executables must have the incoming and outgoing connection open for the loopback to work correctly.

► **To set up an inbound rule for a Windows program:**

1. Run Windows Defender Firewall with Advanced Security.
2. Select Inbound Rules.
3. Choose **Action > New Rule**.
4. Select Program and then click Next.
5. Enter the path to the program executable and then click Next.
6. Select Allow the Connection and then click Next.
7. For the rule, select Domain, Private, and Public. Click Next.
8. Optionally, enter a name for the rule.
9. Click Finish.



3. Basic Troubleshooting

This chapter consists of troubleshooting checklists for some of the most common problems MAK customers encounter.

3.1. Product Does Not Run	27
3.1.1 Linux Applications	27
3.1.2 HLA Applications	27
3.1.3 License Management Checklist	28
3.2. MAK ONE Products Do Not Work Together	30
3.2.1 DIS Applications	30
3.2.2 HLA Applications	32
How to Contact Us	35
Telephone	35
E-mail	35
Internet	35
Post	35

3.1. Product Does Not Run

If you start a MAK ONE product and nothing happens, or a console window flashes, the most likely problem is that license management is not properly configured. See the "3.1.3 License Management Checklist" on the facing page for details.

Most products create a log file in the `./bin64` directory. Check the log for messages. If there is no log, run the executable from the command prompt to check the error messages.

3.1.1 Linux Applications

If you are running an application on Linux, make sure you are running on the same version as that on which the application was compiled (for example, Red Hat Linux Enterprise Workstation 7.0 or 8.0).

3.1.2 HLA Applications

If an HLA application does not start (or the `vpNetHLA` module fails to load), check the following:

- If you get a message that the program cannot find one of the following libraries: *libfedtime*, *libRTI-NG*, *libfedtime1516*, or *librti1516*, you are trying to run an HLA executable and the RTI libraries could not be found. Make sure that an RTI is installed on the machine and that the path to the RTI libraries is in your path:
 - `PATH` on win32
 - `LD_LIBRARY_PATH` on Linux and Solaris.
- Make sure that the application can find the FED file. The FED file should be in the directory from which the application is running or at the location specified by the `RTI_CONFIG` environment variable.
- If the federation requires an *rtiexec*, make sure the *rtiexec* is running and that each federate is capable of connecting to it.

MAK RTI

If you are using the MAK RTI and receive one of the following error messages at startup, the problem is as indicated:

- ****FATAL ERROR**:** Could not create Federation Execution VR-Link:
RTI exception: CouldNotOpenFED ()
The executable could not find the FED file in either its current directory or the location specified by `RTI_CONFIG`.
- ****FATAL ERROR**:** Could not create Federation Execution VR-Link:
RTI exception: RTIinternalError ()
The executable may be looking for an *rtiexec* that is not running.

See "3.2.2 HLA Applications" on page 32 (under MAK ONE Products Do Not Work Together) for more HLA troubleshooting tips.

3.1.3 License Management Checklist

If your MAK ONE product does not start and you believe that it is a license management problem, check the following:

- Did you read and follow the license management installation instructions in your product's users guide?

We do not mean to be facetious when we ask this question. Many customers obtain MAK ONE products via download and do not realize that the installation includes a PDF version of the product documentation, which includes installation instructions. Every MAK ONE product has a `./doc` directory, which contains user documentation. Installation instructions are typically found in the Installing chapter of the users guide.

- Are you running the correct version of the FLEXlm software?
- Is the license server running?

To check whether the license server is running, on the server machine, run the *lmstat* utility in the *flexlm* directory.

- Do you have a valid license for the product and operating system you are trying to start?
- Does the name in the license file match the name of the license server?
- Is the license file an ASCII text file? We provide the license file in text format, but some customers open it in a word processor such as Microsoft Word to look at it and do not save it back to text format.
- Does the license file have the extension `.lic` and is it in the License Management directory?



When looking at files in Windows Explorer, make sure that you have not hidden extensions for known file types. This option can be found in Tools/Folder Options/View, and it is often ON by default. Make sure that "Hide extensions for known file types" is NOT checked.

- Did you set the MAKLMGRD_LICENSE_FILE environment variable on the license server and on the computer on which you are trying to start an application (if the computer is different from the license server)?



For applications built with VR-Link 3.13.1 or later, use of the MAKLMGRD_LICENSE_FILE environment variable is optional. The alternatives for specifying the license server are described in the Installing chapter in the users guide for your MAK ONE product.

- If the computer on which you are running the application is not the same computer as that which is running the license server, can this computer see the license server over the network?

To see if one computer can see another, from a command prompt, run:

```
ping server_name
```

Errors will be reported if the server cannot be found.

- If you have upgraded to a new version of a MAK ONE product, does your maintenance agreement cover this upgrade? (If it does not, you will probably get an error message stating that the license does not support the software.)

If the answer to any of these questions is no, make the needed changes and try again. If the answer to all of these questions is yes, contact MAK support at <https://www.mak.com/mak-one/support/portal>. You may have a corrupt or otherwise invalid license file or some other problem that we can help you with.

License Management Frequently Asked Questions

This section lists some common questions about license management.

- Do I need to start the license server each time I boot the server machine?
The license server must be manually started each time the machine starts up unless it has been configured otherwise.
- How do I configure the MAK license server as a service on Windows?
On Windows, use the *lmtools* utility from the *flexlm* directory. In the Configure Services tab, specify a Service Name, lmgrd path, license file, and log file. Then check Use Services and Start Server at Power Up.
- How can I know that the license server has started successfully on UNIX?
The license server's LOG file (in */usr/local/flexlm*) should list all the components that are in your license file.
- How can I know that the license server has started successfully on Windows?
 - If the license server has been configured as a service, the license server's log file (specified in *lmtools*) should list the components from your license file.
 - If the license server has been started with *runLM*, the console window should list all the components from your license file.
- How can I find out which licenses are currently in use?
On the server machine, run `lmstat -A` from a console.

License Management Error Messages

This section lists common FLEXlm error messages, their causes, and how to respond.

- Cannot find license file [FLEXlm error: -1]
Neither a license file nor a license server could be found. Make sure the `MAKLMGRD_LICENSE_FILE` environment variable is set.
- Cannot connect to license server [FLEXlm error: -15]
The license server machine could not be found. Either the client machine is not networked to the server machine or you need to set the `MAKLMGRD_LICENSE_FILE` environment variable.
- License server does not support this version of this feature [FLEXlm error: -25]

The maintenance for this product is not up-to-date in the license file. Send an email to keys@mak.com with your information.

- Licensed number of users already reached [FLEXlm error: -4]
All available licenses are in use for the product. One of the licenses must be checked in before the program will start.
- No such feature exists [FLEXlm error: -5]
Your license file does not have a license for the product you are trying to run. Check your license file. You can obtain licenses from keys@mak.com.
- Feature has expired [FLEXlm error: -10]
Your product license has expired. Contact keys@mak.com to get a new license.
- The FLEXlm License Finder dialog box always appears at startup. This dialog box asks you to specify either the license server or the license file.

Make sure that the license server is running and that MAKLMGRD_LICENSE_FILE is set.

3.2. MAK ONE Products Do Not Work Together

If you are trying to use several MAK ONE products together, such as the Data Logger and VR-Vantage or VR-Forces and VR-Vantage, and you cannot get data from one to be visible in the other, it is usually a network or simulation protocol configuration problem, or you are running incompatible versions of the applications.

3.2.1 DIS Applications

If DIS applications do not see each other:

- Make sure that you are running the DIS versions of the applications.
- Are all applications using the same port?
- Do all machines share a common netmask and are all applications sending data to the same IP broadcast address?
- Can the computers ping each other?
- If you are playing a Data Logger recording, are you playing a recording of a DIS exercise?

DIS FAQ

- Where are the MAK DIS applications?

In older MAK ONE products, DIS applications are usually in the `./binDIS` directory. On Windows, some DIS applications have DIS as part of the product name in the start menu. For VR-Forces, you choose a DIS connection in the VR-Forces Launcher. For newer versions of MAK ONE products, the DIS applications are in `./bin64`. The file name includes DIS.



VR-Vantage applications are in `./bin64`. They are not protocol-specific. You choose a simulation connection after you start the base executable.

- How can I set the port?

By default, most MAK ONE products use port 3000. (See "Chapter 2. Default Ports for MAK Products" on page 22 for a list.) To change the port, start an application with the `-P` command-line option. The Logger, VR-Forces, and VR-Vantage let you set the port dynamically as part of the connection process.

Troubleshooting the Subnet Mask

To troubleshoot DIS connection problems, try running two VR-Link F18 applications. If you do not have VR-Link, the VR-Link net utilities installer has simple DIS applications that do not require a license.

To run the F18 DIS application, locate the `f18DIS_64` executable in the `./bin64` directory. Launch two F18DIS applications. To specify a destination broadcast address, use the `-A` argument. To specify a port, use `-P`. To specify a device address use `--deviceAddress`.

The f18DIS applications print their connection settings to the console when they are initialized in the format `DESTINATION_ADDRESS:PORT`, for example:

```
DIS socket OK to 10.255.255.255:3000
```

When the F18 example discovers another F18 on the network, it prints a second New Entity line.

If the F18DIS applications both print that they opened a DIS Socket on the same `DESTINATION_ADDRESS:PORT` combination, but do not see each other or are able to shoot each other down, there is a break in the connection. Specifying the subnet mask for your network will usually fix the issue. This is especially likely if your subnet mask does not fall into the IPv4 classful default values listed in Table 12.

Table 12. IPv4 classful default values

Class	Leading bits	Default subnet mask in dot-decimal notation
Class A	0	255.0.0.0
Class B	10	255.255.0.0
Class C	110	255.255.255.0

To set a subnet mask for the F18 DIS application, use the `--subnetMask` command line argument.

Finding or Changing the Subnet Mask

Typically, machines on a LAN have the correct broadcast addresses, so VR-Link applications will be able to communicate with each other immediately.

On UNIX-based systems, you can use the `ifconfig` command to examine and set the configuration of network devices. For example, to set the netmask use:

```
ifconfig device_name netmask 0xFFFF0000
```

In Windows systems, you can use `ipconfig` to examine the configuration of network devices.

If your netmask appears to be wrong in your network settings, please see your system administrator.

VR-Link computes the broadcast address using the device's netmask and host address. Some machines have more than one network device capable of broadcasting. Unless your application is designed to specify a specific device or address (which is possible using VR-Link routines), a VR-Link application uses the broadcast address of the first device listed in the interface table.

3.2.2 HLA Applications

If HLA applications do not see each other:

- Make sure you are running the HLA versions of the applications. If you are using ownership management, DDM, or MOM services, make sure you are running applications built against the same RTI specification (1.3 or 1516).
- Is the same version of the RTI installed on each computer on which an application is running?
- Are you using the correct version of the RTI (1.3 or 1516) for your applications?
- If required, did you start the `rtiexec`?
- Are the RID files configured the same way? For example, if you are using the MAK RTI, is the `udpPort` value the same on each computer?
- Are all federates using the same FED file and can they find the FED file? The FED file should be in the directory from which the application is running or at the location specified by the `RTI_CONFIG` environment variable.
- Are all federate joining the same federation?
- Can the computers ping each other?
- If you are playing a Data Logger recording, are you playing a recording of an HLA exercise?

HLA FAQ

Many of the configuration issues described in this section are RTI implementation-specific. Assume that the information is specific to the MAK RTI unless specified otherwise.

- Where are the MAK HLA applications?

In older version of MAK ONE products, HLA applications are stored in `./binHLA13` or `./binHLA1516`. The `./binHLA13` directory contains executables for use with RTI NG 1.3. The `./binHLA1516` directory contains HLA 1516 executables. On Windows, HLA applications have either HLA 1.3 or HLA 1516 as part of the product name in the Start menu or the connection dialog box.

On more recent versions of MAK ONE products, HLA applications are in `./bin64`. The version of the application is indicated by the presence of 13, 1516, or 1516e in the executable name.



VR-Vantage applications are in `./bin64`. They are not protocol-specific. You choose a simulation connection after you start the base executable.

- How do I use an RTI?

To use an RTI, each HLA application (or federate) must load the RTI libraries. In most cases, the *rtiexec* must also be running.

- How does an HLA application find the RTI libraries?

Typically, the RTI libraries are dynamically linked with the executable. On Windows, this means that an HLA application will look for the libraries in its current directory, and then in the directories specified in the PATH. On IRIX, the executable will look in the directories specified in LD_LIBRARYN32_PATH. On other UNIX-based systems, the executable will look in the directories specified in LD_LIBRARY_PATH.

- Where can I make changes to default RTI settings or view current RTI settings?

RTI configuration settings are in the RTI's RID file. For the MAK RTI, this file is called *rid.mtl* and is located in the RTI's top level directory. You can view, but not edit, RID settings in the RTI Assistant's Federations View. You can also view RID settings in the RTIspy. Some settings can be edited.

- How can I specify which RID file to load?

The easiest way to specify which RID file to load is to use the RTI Chooser. For details, please see the *MAK RTI Users Guide*. Otherwise, the search order for the RID file depends on how you specify it.

If the RTI_RID_FILE environment variable is not defined, the search order is:

1. Try to open *rid.mtl* in the directory from which you run the application (which is not necessarily the directory in which the executable is located.)
2. Try to open the file specified by *RTI_CONFIG/rid.mtl*.
3. If no RID file is identified by one of these methods, the MAK RTI prints a warning and opens using default values.

If the RTI_RID_FILE environment variable is defined, the search order is:

1. Try to open the file specified in RTI_RID_FILE.
2. Try to open the file specified by RTI_CONFIG/RTI_RID_FILE.
3. If a RID file cannot be found in these locations, the MAK RTI stops looking for the RID file and fails.

- How can I specify which FED file to use?

For MAK ONE products, use the *-F* command-line option. For most MAK ONE products, the federation name and FED file name must match. You can specify the FED file through the graphical user interface in the Logger and VR-Vantage. You can also specify a FED file in the VR-Forces Launcher.

- What happens if the RTI cannot find the RID file?

The RTI uses default values.

- When does a federation require *rtiexec*?

Some RTIs require that you always run an *rtiexec*. If you are using the MAK RTI in lightweight mode, you do not need to run *rtiexec*. If you want to use services such as synchronization points, save/restore, time management, and reliable transport you must run *rtiexec*. Please see the *MAK RTI Reference Manual* for details.

- How do I configure the RTI to use *rtiexec*?

In *rid.mtl*, set **useRtiExec** to 1. For the changes to take effect, make sure that each federate reads in a RID file that contains this change.

You can also configure use of the *rtiexec* when you add a connection in the RTI Preferences dialog box. For details, please see "Adding RTI Connection Configurations" in the *MAK RTI Users Guide*.

- How do I specify a federation to join?

For MAK ONE products, use the `-x` command-line option. For MAK ONE products, the federation name and FED file name need to match. For the vpNet module, this can be specified in LynxPrime in the NetHLAConnection.

The VR-Forces, Logger, and VR-Vantage applications let you specify the federation through their simulation connection configuration mechanisms. Please see the relevant users guides for details.

Common error messages:

- Unable to Locate Component: This application failed to start because libRTI-NG.dll was not found. Re-installing the application may fix the problem.

The HLA application cannot find the RTI libraries. Double check that an RTI is installed and that the path to the libraries is in PATH, LD_LIBRARYN32_PATH, or LD_LIBRARY_PATH.

- Error while loading shared libraries: libRTI-NG.so cannot open shared object file: no such file or directory.

The HLA application cannot find the RTI libraries. Double check that an RTI is installed and that the path to the libraries is in PATH, LD_LIBRARYN32_PATH, or LD_LIBRARY_PATH.

- ****FATAL ERROR****: Could not create Federation Execution note -> vuLibrary::Load - failed loading libvpnethla ... warn -> vpModule::initializeModule("vpNetHLA") - cannot load library warn -> valid class-level tag 'NetHLAConnection' encountered, but there is no class registered.

The vpNet module cannot find the RTI libraries. Double check that an RTI is installed and that the path to the libraries is in PATH (win32), LD_LIBRARYN32_PATH (IRIX), or LD_LIBRARY_PATH (Linux/Solaris).

- ****FATAL ERROR****: Could not create Federation Execution VR-Link: RTI exception: CouldNotOpenFED ()

The HLA application could not find the FED file. Double check its location.

How to Contact Us

For MAK ONE product 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:	www.mak.com/support/get-licenses
Technical support:	www.mak.com/mak-one/support/portal
Product version and platform information:	www.mak.com/support/product-versions
For the free, unlicensed MAK RTI:	www.mak.com/support/bonus-material

Post

Send postal correspondence to:	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.



4. Conventions and Third-Party Licenses

This chapter explains the conventions used in this manual and MAK Technologies documentation.

Document Conventions	37
DI-Guy Conventions	37
Mouse Button Naming Conventions	38
Third-Party Licenses	38
Third-Party Libraries Supported	38
Library Licensing	39
Boost License	41
libXML and libICONV	42
Lua	42
LizardTech	42
Freefont OpenType Font Set	42
Scintilla and SciTE	43
Autodesk Gameware Navigation	43
NVIDIA	43
Third-Party Licenses for VR-Vantage Applications	44

Document Conventions

This manual uses the following typographic conventions:

Sans Serif Bold	Indicates a key on the keyboard or menu options. 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, (-h --help).
()	Indicates a directory. Since MAK products run on both Linux and Windows PC platforms, we use the / (slash) for generic discussions of pathnames.
/	Indicates a file name, pathname, or a class name.
<i>Italic</i>	Indicates a procedure.
►	Indicates a menu choice. For example, an instruction to select the Save option from the File menu appears as:
Menu > Option	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 ONE home directory. For example, the directory `./vrlink#.#/doc` appears in the text as `./doc`.

DI-Guy Conventions

The following table lists the conventions used for entity coordinates in DI-Guy documentation.

Convention	Indicates
XYZ	X = forward, Y = to the left Z = up
Yaw, Roll, Pitch	Orientation is applied in the order YRP. Yaw = rz – orientation about the vertical axis Roll = rx – orientation about the fore-aft axis Pitch = ry – orientation nose down, nose up

Mouse Button Naming Conventions

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

Third-Party Licenses

MAK software products may use code from third parties. This section contains summary license information and, where required, specific license documentation required by these third parties.

Third-Party Libraries Supported

MAK ONE 2022 uses the indicated versions of the following libraries:

- VR-Link 5.8
- Qt 5.15.2 (VC 15 and Red Hat 8)
- OpenSceneGraph 3.6.4
- Boost 1.63.0
- FreeGlut 2.4.0 (Windows), 2.6.0 (Linux)
- GL Studio 7.0 (VC 15)
- DI-Guy 13.6 (development version)
- Triton 4.43
- SilverLining 6.20
- SpeedTree 6.3.2
- CDB API 2.2 (Windows), 2.1 (Linux)
- osgEarth 3.2
- CUDA 11.7

Library Licensing

The following libraries are covered by the GNU Lesser General Public License (LGPL) license:

- CIGI
- DevIL
- GEOS
- GStreamer
- LibIconV
- LibWebSockets
- OP CODE
- OpenAL
- OSG
- OsgEarth
- Pthread
- Qt
- Qwt

The following libraries are covered by the ZLib license:

- Bullet Physics
- LibPNG
- LibSDL
- Minizip
- Zlib

The following libraries are covered by the MIT license:

- Expat
- FreeGLUT
- GDAL
- GeoTIFF
- GLM
- HarfBuzz
- LibCURL
- LibSquish
- LibXML
- LUA
- LuaBind
- Proj
- SMHasher
- TCLAP
- Qt Visual Graph Editor
- ZeroBrane Studio

The following libraries are covered by the BSD license:

- FreeType
- GLEW
- JPEG
- LibTiff
- Protobuf
- PCRE
- Ogg Vorbis

The following libraries are covered by the commercial licenses:

- FLEXlm
- Gameware
- GLStudio
- JRM Technologies SenSimRT and SigSimRT
- OpenFlight
- SpeedTree
- Sundog Triton
- Sundog SilverLining
- CDB API

The following libraries are covered by custom licenses:

- NVIDIA CG Toolkit
- FreeImage (FreeImage Public License)
- Boost (Boost Software License)
- Poco (Boost Software License)

COLLADA DOM is covered by the SCEA Shared Source license.

Thread Building Blocks is covered by the GPL license:

SQLite is in the public domain.

SWIG is covered by the GPL license.

Boost License

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

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

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

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

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE

DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

libXML and libICONV

VR-Link, and all MAK ONE products that use VR-Link, links in libXML and libICONV. On some platforms the compiled libraries and header files are distributed with MAK products. MAK has made no modifications to these libraries. For more information about these libraries, see the following websites:

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

Lua

Some MAK products use the Lua programming language (www.lua.org). Its license is as follows:

Copyright © 1994–2012 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

LizardTech

Portions of this computer program are copyright © 1995-2010 Celartem, Inc., doing business as LizardTech. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign Patents Pending.

Freefont OpenType Font Set

VR-Vantage applications and VR-Forces use the Freefont OpenType font set from the Free Software Foundation. It is covered by the General Public License (GPL). For details, see: <http://www.gnu.org/licenses/gpl.html>

Scintilla and SciTE

License for Scintilla and SciTE. Copyright 1998-2002 by Neil Hodgson
<neilh@scintilla.org>. All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Autodesk Gameware Navigation

The following are registered trademarks or trademarks of Autodesk, Inc., in the USA and other countries: 3ds Max, AutoCAD, Autodesk, DWF, DWG, DXF, Kynapse, Maya, MotionBuilder, and Powered with Autodesk Technology.

NVIDIA

The HDR implementation uses the HDR NVIDIA gameworks SDK example that has the following copyright notice:

Multiple Files in folder: es3aep-kepler\HDR/

SDK Version: v2.11

Email: gameworks@nvidia.com

Site: <https://developer.nvidia.com>

Copyright (c) 2014-2015, NVIDIA CORPORATION. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of NVIDIA CORPORATION nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR

CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Third-Party Licenses for VR-Vantage Applications

VR-Vantage applications use a variety of third-party libraries. Developers who want to use these libraries may be required to purchase developer's licenses. Please see “Third-Party Software and Content” in Chapter 1, Introduction to VR-Vantage, in *VR-Vantage Users Guide* for details.



Index

-

-P command-line option, 10

A

application number, 15

B

broadcast address, 31

C

compiler, 11
configuration file
 RTI, 10
configuring
 FLEXlm as a service, 29
connection
 loopback, 25

D

Data Logger, 13
 ports, 23
database
 effect of optimization, 20
DDM, 10
DIS
 loopback connection, 25

DIS application
 incompatibility, 30
 running, 10

E

environment variable
 RTI_CONFIG, 33
 RTI_RID_FILE, 33
error message, 27
 FLEXlm, 29

F

FDD file, 10
FED file, 10, 16, 27, 33
 naming convention, 16
federate, 10
Flat Earth database, 20
FLEXlm, 12
 error messages, 29
flexlm directory, 28
FLEXlm Licensing ports, 25
FLT file
 features supported, 20
FOM
 compatibility, 16

H

HLA
 1.3 and 1516 interoperability, 16
 1516, troubleshooting, 15

- HLA 1.3, 12, 32
 - compatibility with 1516, 16
 - interoperability with IEEE 1516, 13
- HLA 1516, 10, 12, 32
- HLA application, 9
 - incompatibility, 32
 - startup problems, 27

I

- IEEE 1516
 - compatibility with 1.3, 16
 - interoperability with HLA 1.3, 13
- ifconfig command, 31
- installing
 - RTI, 10
- interoperability
 - HLA 1.3 and 1516, 16
 - HLA 1.3 and IEEE 1516 federates, 13
- ipconfig command, 31

L

- LAN, 10, 31
- Legion
 - loopback connection, 25
 - port, 23
- library
 - RTI, 10, 27
- license management
 - troubleshooting, 28
- license server, 9, 28
 - configuring as service, 29
- lightweight mode, 10
- linking to VR-Link and RTI, 14
- Linux, 11, 27
- log file, 27
- Logger, 13
 - interoperability of HLA 1.3 and IEEE 1516, 13
 - ports, 23
 - recording to from VR-Forces, 15
- loopback connection, 25

M

- maintenance agreement, 29
- MAK Data Logger, 13
 - ports, 23
- MAK Legion
 - loopback connection, 25
 - port, 23

- MAK RTI, 9, 12
 - error messages, 27
- MAK RTI port, 23
- maklmgrd executable, 9
- MAKLMGRD_LICENSE_FILE environment variable, 28
- Microsoft Visual C++ 6.0, 11
- MOM, 10

O

- Open Flight database, 20
- operating system
 - compatibility, 11

P

- path, 27
- ping, 28
- port
 - 3000, 10
 - DIS, 10
 - FLEXlm Licensing, 25
 - Legion, 23
 - Logger, 23
 - MAK Data Logger, 23
 - MAK Legion, 23
 - MAK RTI, 23
 - RadarFX, 25
 - RTI, MAK, 23
 - VR-Engage, 25
 - VR-Exchange, 24
 - VR-Forces, 24
 - VR-Link, 23
 - WebLVC, 24

Q

- Qt, 11

R

- RadarFX ports, 25
- recording
 - to Logger, 15
- requirements
 - basic application, 9
- RID file, 33
 - search order, 33
- rid.mtl, 10

RTI
 configuration file, 10
 configuration settings, 33
 installation, 10
 libraries, 27
 linking to, 14
 version, 32
RTI library, 10
RTI port, MAK, 23
RTI.rid, 10
RTI_CONFIG, 27
RTI_CONFIG environment variable, 32-33
RTI_RID_FILE environment variable, 33
rtiexec, 10, 27

S

search order
 RID file, 33
startup problem, 27
Stealth, 13
 synchronizing terrain with VR-Forces, 19
Stealth Control PDU, 15

T

terrain
 synchronizing between Stealth and VR-Forces,
 19
third party API, 11
Time Management, 10
troubleshooting, 27

U

UDP port, 10
udpPort, 32
useRtiExec, 34
UTM, 20

V

viewing
 Logger files, 15
vpNetHLA, 27
VR-Engage port, 25
VR-Exchange ports, 24
VR-Forces, 12, 15
 recording to Logger, 15
 synchronizing terrain with Stealth, 19
VR-Forces ports, 24

VR-Link, 12
 linking to, 14
VR-Link port, 23

W

WebLVC ports, 24
WGS84 datum, 20

X

XML file, 16



MAK ONE

MAK-0-1-230614