



MAK RTI 3.1.2 Release Notes

This document provides the following release-specific information for MAK RTI 3.1.2:

Supported Systems	2
Qt Release Compatibility	2
Libraries and Binaries Built with Visual Studio 2005	3
FLEXlm Support	3
Third Party Library Dependencies	4
RTIsPy Web GUI Configuration	4
Backwards Compatibility	5
Network Compatibility	5
Compatibility with Other RTIs	6
New Features and Updates	6
Bug Fixes	7
Known Problems	7
RTI 1516 Implementation Issues	9

MAK RTI 3.0 has been verified by DMSO as fully compliant with the IEEE 1516 version of the HLA Interface Specification (SISO DLC HLA API 1516 (SISO-STD-004.1-2004).)

Copyright © 2007 MAK Technologies, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MAK Technologies®, VR-Forces®, RTIsPy®, and VR-Link® are registered trademarks of
MAK Technologies, Inc.
Document ID: RTI-3.1.2-2-070730

Supported Systems

Table 1 lists the platforms supported by MÄK RTI 3.1.2. Please contact MÄK if you need it for another platform. Application code must be built with the indicated compilers in order to link to the MÄK RTI libraries.

Table 1: Platforms supported

Operating System	Compiler
SGI IRIX6.5	MipsPro7.4.2m C++ compiler
Sun Sparcstation with Solaris 8	SPARCCompiler C++ 5.2
Red Hat Enterprise Linux Workstation 3	GNU C++ (GCC) 3.2.3; glibc 2.3.2
Red Hat Enterprise Linux Workstation 4	GNU C++ (GCC) 3.4.4; glibc 2.3.4
Red Hat Fedora Core 3	GNU C++ (GCC) 3.4.4; glibc 2.3.6
Windows 2000/XP	Microsoft Visual C++ 7.1, 8.0

Qt Release Compatibility

The MÄK RTIspy GUI uses Trolltech's Qt cross-platform GUI toolkit. (The Qt run-time libraries are distributed with the MÄK RTI). If a federate also uses Qt, it must use the same version of Qt as the MÄK RTIspy GUI, in order for the RTIspy LRC GUI to function properly for that federate. Federates built with other versions of Qt should disable the RTIspy LRC GUI to avoid these incompatibility problems. If your federate has an incompatible Qt GUI, you can still use the RTIspy Web GUI to monitor your federate.

The RTIspy LRC Console GUI Plug-in is built with Qt 3.3.5.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1 and 8.0. When you run MÄK products together on the same computer, for example, the MÄK RTI and the Stealth, we strongly recommend that you run versions compiled with the same compiler. Mixing products compiled with different versions of the compiler can result in program instability.

Libraries and Binaries Built with Visual Studio 2005

All MÄK products built with Microsoft Visual Studio require the C Runtime Library to function. The C runtime libraries have always been available from Microsoft for download, they are also installed on a user's machine when a Microsoft compiler is installed. The C runtime libraries are not part of the normal Windows installation. For customers who plan to use MÄK products on machines that do not have a compiler installed, MÄK has historically distributed a copy of the C Runtime Libraries with MÄK products. These libraries were put in the *bin* directory used by the MÄK products. MÄK products would then use the libraries in the *bin* directory and customers would not have a problem if copies of the libraries were not already installed.

Unfortunately, with the release of the new C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0), the libraries can no longer just be copied into the *bin* directory of an application. The libraries need to be installed correctly into Windows system folders. (The process is actually a little more complicated, a manifest file needs to be created to tell Windows where to find the libraries.)

To accommodate this change, MÄK is distributing the Windows installer for the C runtime libraries with all MÄK products released for MSVC++8.0. The installer is named *vcredist_x86.exe* and is in the base directory of any installed MÄK product.

Running the installer requires Administrator privileges for the machine the installer is run on. MÄK has chosen to not integrate the MÄK installer and the Microsoft installer so as not to require users to have Administrator privileges to install MÄK products. Therefore, if you who do not have a compiler installed, or get error messages like "Software has not been installed correctly, please re-install", you must apply the patch.

For more information see this Microsoft URL:

<http://msdn2.microsoft.com/en-us/library/ms235299.aspx>

FLEXlm Support

MÄK RTI 3.1.2 uses FLEXlm 11.4 for Windows, Sun Solaris, and Linux. SGI IRIX uses FLEXlm 9.2.

Third Party Library Dependencies

In addition to the Qt dependencies mentioned previously, MÄK RTI 3.1.2 has the following dependencies:

- ♦ QT - 3.3.5 - <http://www.trolltech.com/>
- ♦ QWT - 0.4.1 - <http://qwt.sourceforge.net/>
- ♦ EHS - 1.3.1 - <http://xaxxon.slackworks.com/ehs/>
- ♦ PME - 1.0.4 - <http://xaxxon.slackworks.com/pme/>
- ♦ PCRE - 6.4.1 - <http://www.pcre.org/>
- ♦ ZLIB - 1.2.3 - <http://www.zlib.net/>
- ♦ ICONV - 1.8 - <http://www.gnu.org/software/libiconv/>
- ♦ LIBXML2 - 2.6.22 - <http://www.xmlsoft.org/>
- ♦ prototype.js - 1.5.0 - <http://prototype.conio.net/>
- ♦ script.aculo.us - 1.6.1 - <http://script.aculo.us>
- ♦ VR-Link 3.11.

RTIsPy Web GUI Configuration

The RTIsPy Web GUI has the following system considerations:

- ♦ The RTIsPy Web GUI is optimized for screen resolutions of 1024x768 and larger. For smaller screen sizes, additional window scrolling will be required.
- ♦ The RTIsPy Web GUI has been tested and will display correctly with Mozilla Firefox version 1.5 and higher and Internet Explorer versions 6 and higher. Other browsers may be compatible, but they have not been tested and are not specifically supported.
- ♦ The RTIsPy Web GUI requires an RTI Plus license for each LRC. If licenses are available for only a subset of the federation, then those licenses will be consumed in the order that federates are started. Once all licenses are used, additional federates will not be available for monitoring (although they will still be viewable at the federation level).

MÄK RTI Licensing for the RTI Forwarder

You can run the RTI Forwarder without an RTI Plus license if it is used stand-alone (that is, centralized forwarding, in which it is not part of a distribution network and the routes file is empty.) It uses a license if it is run as part of a distributed network with other forwarders. An RTI Plus license allows you to run:

- ♦ One federate with GUI and plug-ins
- ♦ One rtiexec with GUI
- ♦ One distributed forwarder
- ♦ Plug-ins for either the rtiexec or the Distributed Forwarder (but not both with a single license).

Backwards Compatibility

The MÄK RTI 3.1.2 libraries are not run-time compatible with previous versions. (In other words, you cannot have a federation execution in which some federates use MÄK RTI 3.1.2 and some use a previous version of the MÄK RTI.) However, the MÄK RTI 3.1.2 is link compatible with previous versions, so a federate does not need to be recompiled to use this version. We recommend that all federates use the same version of the RTI when operating within the same federation (or using the same *rtiexec*).

Network Compatibility

The MÄK RTI is not network compatible with other RTIs. (In general, no two RTI implementations are network compatible, due to differences in the low-level network mechanisms that they use (which include, but are not limited to packet layout.)) Applications that want to interoperate in the same federation execution must all use the same RTI implementation. A federation can easily switch from one RTI implementation to another between runs, but during each run, all participants must agree on which RTI to use, much as they must also agree on which FOM to use.

Compatibility with Other RTIs

The MÄK RTI 1.3 implements the API dictated by the 1.3 version of the HLA Interface Specification, as amended by DMSO's official HLA 1.3 Interpretations document. All RTIs that are verified as compliant with the HLA 1.3 specification are written to this exact API standard. This insures that the MÄK RTI is link-compatible with other RTIs that have been verified as HLA compliant.

The MÄK RTI 1516 implements the SISO DLC HLA API¹ 1516 (SISO-STD-004.1-2004). This API supports dynamic link compatibility, which was problematic with the original IEEE 1516 API. The IEEE 1516 API only supports compile time compatibility (uses C++ templates and implementation-specific header files). The MÄK RTI is compatible with any RTI written to the SISO DLC HLA API 1516.

Because an RTI is typically accessed by an application through a dynamic library (*.dll* on Windows, *.so* on UNIX), applications do not need to be recompiled or relinked to switch among dynamic-link-compatible RTI implementations that have been built using the same compilers. The only requirement is that the appropriate version of the library be located within the shared library search path. For more information, please see section 4.2, “Compiling and Linking with the MÄK RTI 1.3” in *MÄK RTI Reference Manual*.



Because the DMSO RTI NG v6 was last built with the MSVC++ 6.0 compiler, it is not link compatible with the MÄK RTI 3.1.2. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 3.1.2 would be link compatible.

New Features and Updates

MÄK RTI 3.1.2 is primarily a maintenance release. However it also has the following improvements:

- The dynamic dependency on *libxml2*, which had caused problems on some platforms, has been changed to a static inclusion of the *libxml2* library.

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

Bug Fixes

The following bugs have been fixed in this release:

- ♦ 1516 federates using a 1.3 FED file and Data Distribution Management (DDM) could not associate a region with attributes or register with region.
- ♦ When a federate added a multicast group via `RTI_addDestAddrString`, there was a bug in receiving multicast traffic sent to that group.
- ♦ The RTI would crash when multiple requests to release attribute ownership were received for a given attribute.

Known Problems

This release has the following known problems:

- ♦ Distributed forwarder multicast subscription does not work for late-joining forwarders. With the introduction of the unified Distributed Forwarder a need arose to communicate between federates on a LAN and forwarders on remote LANs when a DM or DDM multicast group is joined or dropped, such that the forwarders can monitor or drop those groups on its LAN and forward messages back to the federates joined to such multicast groups. However the scheme to accomplish this does not work if the forwarder is not connected to the forwarder network when the federate joins the multicast group. The workaround is to ensure that the forwarder network is completely established prior to launching federates, or at the least, prior to allowing federates to attempt to join DM or DDM multicast groups.
- ♦ The RID consistency feature relies on the ability to establish a connection between a federate and the RID tester (presumably the `rtiexec`), however if the federate's `RTI_enableTcpCompression` parameter and the RID tester's parameter are different, the two cannot communicate, and consequently the RID consistency test request and response messages cannot be exchanged. The workaround is to ensure that the `RTI_enableTcpCompression` parameter is consistent among all participants prior to launch.
- ♦ Bundling (`RTI_enablePacketBundling`) cannot be used when also using big messages (`RTI_enableBigMessage`). The format of messages when big messages is enabled conflicts with how messages are bundled.
- ♦ The XML processing of the FDD DTD attribute differs between Linux and Windows. In Linux, the DTD is found relative to the federate's execution directory. In Windows, the file is found relative to the FDD location.
- ♦ In 1516, the HLA DTD is not distributed with the FDD file. In order for the LRC to process the FDD file, the HLA DTD must be accessible as specified in the FDD DTD attribute.

- ♦ The MÄK Technologies Lisp (MTL) files used by MÄK products to specify configuration settings are parsed at startup by the RTI. When a configuration file has certain errors, such as unmatched parentheses, the parser will not necessarily return an error code. This can cause the RTI to reach a partially initialized state or cause the RTI to initialize itself with default RID values, causing unexpected behavior. This applies to lisp expressions in the RID file as well as the expressions passed to the 1516 createRTIambassador function.
- ♦ The RTI can be configured so that when a federate terminates abnormally, the RTI deletes the objects for which the federate owns the privilegeToDelete attribute. If a federate has divested this attribute, the RTI does not recognize the ownership transfer with regard to the delete orphan behavior. As a result, when a federate terminates, abnormally or through normal resignation, the delete orphan behavior is enforced for any object it had created, whether it owns the privilegeToDelete or not. The exception is when the federate still owns the privilegeToDelete attribute and resigns normally with an action of UNCONDITIONALLY_DIVEST_ATTRIBUTES.
- ♦ A workaround has been added to deal with situations where a host running the RTI Forwarder application has multiple network interface adapters (NICs) or has multiple IP addresses assigned to an interface, which may be the case if the RTI Forwarder host is connected to the WAN via a VPN tunnel. Under these circumstances the RTI Forwarder may fail to initiate connections to other forwarders if the IP address corresponding to its host in the Forwarder List section of the *routes.mtl* file does not match the IP address the RTI Forwarder application receives when it queries the host itself for its IP address. To correct this the IP addresses of the interfaces on which it will connect to other RTI Forwarders must be entered into the *routes.mtl* file Interface Address List section. At a minimum one such entry should correspond to the entry identifying the host in the Forwarder List section of the *routes.mtl* file. Please refer to section 6.3, “[Configuring Networking on a WAN](#)” in *MÄK RTI Reference Manual* for details about the Interface Address List entry format.
- ♦ The HLA 1.3 version of the RTI can sometimes fail to throw an “ObjectAlreadyRegistered” exception if a federate tries to register an object with a name that is already being used by another object. The federate can register an object using a name that belongs to another object that is unknown to the local RTI component (LRC). The LRC only maintains the names of objects that are registered or discovered by the federate. While we plan to fix this problem, a good preventive measure is to always subscribe and tick the RTI before registering objects.
- ♦ The rtiexec does not connect the output streams to a console until after reading the RID file in Windows. As a result, errors opening or parsing the RID file are not displayed. If RTI_enablePopUpErrorMsgs is enabled then these errors are displayed in a popup box, however this solution works only under Windows.
- ♦ To use shared memory on operating systems other than Windows, you must start the Shared Memory Queue Manager manually. Automatic launching of the shared memory queue manager has been disabled because the current implementation resulted in a conflict for console I/O.

- ♦ If you are using the Linux version of the RTIspy with versions of MÄK products earlier than the most recent releases, the MÄK products may crash. If this happens, contact MÄK technical support and we will provide you with a fix for your application.
- ♦ The asynchronous callback feature is not available for HLA 1.3 federates. It is only available for 1516 federates. Enabling this option will have no effect on an HLA 1.3 federate.

RTI 1516 Implementation Issues

The MÄK RTI 1516 has the following known issues:

- ♦ While the 1516 API uses the `std::wstring` to exchange string information, internally, the MÄK RTI 1516 operates on narrow strings only (that is, representable by `std::string`). Therefore, the contents of any wide string passed through the RTI API must be convertible to a narrow string.
- ♦ The publish and subscribe services in the 1516 interface specification are additive when successive calls are made, versus replacement for the 1.3 interface specification. The MÄK RTI 1516 implementation follows the additive policy; however, the MÄK RTI 1.3 follows the replacement policy. As a result, the advisories and discoveries may conflict when a federation is composed of both 1.3 and 1516 federates. The `RTI_processUnknownUpdatesForDiscovery` RID parameter should be enabled as a work around for discoveries.

