



MÄK RTI 4.2 Release Notes

This document provides the following release-specific information for MÄK RTI 4.2:

Supported Systems	2
Compiler Compatibility on Windows	2
License Manager.....	2
Third Party Library Dependencies.....	3
Backwards Compatibility	3
Network Compatibility.....	3
Compatibility with Other RTIs.....	4
New Features and Updates	5
Documentation Updates	6
Bug Fixes	6
Known Problems	9

The MÄK RTI has been verified by DMSO as fully compliant with the IEEE 1516-2000 version of the HLA Interface Specification (SISO DLC HLA API 1516 (SISO-STD-004.1-2004).)
--

Copyright © 2013 VT MÄK, 150 Cambridge Park Drive, 3rd Floor, Cambridge, MA 02140 All rights reserved. VR-Exchange™, VR-TheWorld™, and VR-Vantage™ are trademarks of VT MÄK. MÄK Technologies®, VR-Forces®, RTIs Spy®, B-HAVE®, and VR-Link® are registered trademarks of VT MÄK.
Document ID: RTI-4.2-2-130507

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 4.2. 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: MÄK RTI: Platforms supported

Operating System	Compiler
Red Hat Enterprise Linux Workstation 5, 6. (32 bit and 64 bit libraries)	Default compiler.
SUSE 11 (32 bit and 64 bit libraries)	Default compiler.
Windows XP	Microsoft Visual C++ 8.0, 9.0**
Windows Vista*/Windows 7	Microsoft Visual C++ 8.0, 9.0**, 10.0**
*You must have administrator privileges to install VT MÄK products on Windows Vista.	
**The MSVC++ 9.0 and 10.0 versions have 32 bit and 64 bit libraries.	

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1, 8.0, 9.0, and 10.0 (some products are not available on all compilers). When you run MÄK products together, for example, the Logger and a VR-Vantage application, we strongly recommend that you run versions compiled with the same compiler. Mixing products compiled with different versions of the compiler on the same computer can result in program instability.

License Manager

To run the licensed version of MÄK RTI, you must install license management software. MÄK RTI 4.2 uses FLEXlm 11.11. If you are upgrading from a version of the MÄK RTI that used an older version of FLEXlm, you must upgrade your license management files. You do not need a new license. Licenses are forward compatible.

The License Manager files are not part of the MÄK RTI installer. You can download them at:

- Windows:
<ftp://ftp.mak.com/out/MAKLicenseManager-win-setup.exe> (for 32-bit systems)
<ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe>
- Linux: <ftp://ftp.mak.com/out/MAKLicenseManager-linux-setup.tar.gz>

A license manager FAQ is available at:

<http://www.mak.com/license-support.html>

Third Party Library Dependencies

MÄK RTI 4.2 has the following dependencies:

- ♦ QT 4.7.4.
- ♦ Qwt 6.0.1.
- ♦ FreeGlut 2.4.0
- ♦ Boost - 1.37 on all platforms except VC10, which uses 1.45.
- ♦ 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/>

Backwards Compatibility

The MÄK RTI 4.2 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 4.2 and some use a previous version of the MÄK RTI.) However, except as noted below, the MÄK RTI 4.2 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*).



HLA Evolved federates compiled with MÄK RTI 4.0.4 or later are not link compatible with HLA Evolved federates compiled with MÄK RTI 4.0 through 4.0.3.

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.) 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 1516-2000 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.

The MÄK RTI HLA Evolved implements the IEEE HLA 1516-2010 API. This API supports dynamic link compatibility, and the MÄK RTI is link-compatible with any RTI written to this API specification. There was an error in the HLA Evolved implementation in MÄK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3. As a result, those versions are not dynamic link compatible with RTI 4.0.4, RTI 4.1, or any other HLA 1516-2010 compatible RTIs. It is recommended that any HLA Evolved federates compiled against any of those versions of the MÄK RTI be recompiled against the latest version.

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 3.2, “[Compiling and Linking with the MÄK RTI 1.3](#)” in *MÄK RTI Reference Manual*.

i

Because the DMSO RTI NG v6 was last built with the MSVC++ 6.0 compiler, it is not link compatible with the MÄK RTI 4.2. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 4.2 would be link compatible.

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

New Features and Updates

MÄK RTI 4.2 has the following new features:

- ♦ The web-based RTIspy is no longer provided. The RTIspy networking monitoring function is now integrated into the RTI Assistant. This is documented in various chapters in *MÄK RTI Reference Manual*.
- ♦ The TTL settings for RTI Assistant communications can now be specified with the environment variable `RTI_ASSISTANT_TTL`.
- ♦ Various changes to RID parameters. Please see the next section for details.

Changes to RID Parameters

The following parameters have been added:

- ♦ **RTI_enableLRCNetworkStatisticsMonitoring**. When enabled (set to 1), a federate's network statistics can be viewed in the RTI Assistant's RTIspy window. This capability is provided by a plug-in. There are limits on how many plug-ins can be active at the same time. When **RTI_enableLRCNetworkStatisticsMonitoring** is set to 0, all plug-ins should work as expected. When it is set to 1, the following constraints are in place (meaning plug-ins will not work or will be completely ignored).
 - Plug-ins cannot override the `DtRtiAmbassadorImplementor`.
 - Plug-ins cannot override the `DtConnectionMgr`.
 - Plug-ins cannot override the `DtFedAmbWrapper`.Default: 1.
- ♦ **RTI_tickWaitPeriod**. **RTI_tickWaitPeriod** specifies a wait period during a `tick()` call between any callbacks or IO thread related operations. The `tick()` method may still last the same min and max amount, but it can potentially wait/sleep in smaller increments. Default is 0, which means it sleeps for the minimum time specified in the call to `tick()`.

The following parameters have been removed:

- ♦ **RTI_enableNetworkMonitoring**
- ♦ **RTI_logNetworkMonitorStatistics**
- ♦ **RTI_enableRtiexecWebservice**
- ♦ **RTI_enableLrcWebservice**
- ♦ **RTI_enableLrcWebserviceEventLog**
- ♦ **RTI_webserviceHttpPort**
- ♦ **RTI_webserviceRtiPort**
- ♦ **RTI_webserviceEnableServerAutoStart**
- ♦ **RTI_webserviceEnableForwarding**
- ♦ **RTI_webserviceNotifyLevel**
- ♦ **RTI_connectionTestInterval**.

`RTI_mcastDevAddrString` was previously deprecated and is no longer supported.

`RTI_pluginDirectory` and `RTI-addPluginName` have been replaced with settings in a new configuration file, *plugins.xml*. For details, please see RTIspy API documentation.

Documentation Updates

RTI documentation has been updated to cover all new features.

Bug Fixes

The following bugs (with defect tracking numbers) have been fixed in this release:

- When using HLA_IMMEDIATE callbacks and prior to joining a federation, the RTI will now provide the `reportFederationExecutions` callback. 49652, 49475
- In HLA Evolved, the `federationRestoreStatusResponse` now reports the correct pre and post restore handles. 49684
- Fixed a bug which prevented a single federate from resigning from one federation and then creating a new federation. 44704
- Fixed an issue in merging FOM modules with duplicate `synchronizationPoint` definitions. 48133
- When viewing a federation with an `rtiexec` on a remote machine, the RTI Assistant would sometimes display federates that had already shut down. 47765
- Fixed several issues with time management service calls when using the Java bindings with HLA 1516-2000 and 1516-2010. 47881
- Fixed an issue with the Java bindings HLA 1516-2010 encoder and decoder for the `HLAfixedArray` encoding type. 47900
- Corrected some issues with the HLA 1516-2010 Java example federate code that used MÄK-specific implementations. 47927
- When using the MÄK RTI on Linux, a library conflict prevented the online help from opening in some other MÄK products. 45716
- Fixed a latency issue with messages using Asynchronous IO on Windows. 45811
- The `getObjectInstanceHandle` service call in HLA 1516-2010 was throwing the exception `ObjectInstanceNotKnown` even when a correct object instance name was provided. 46800
- The default IPv6 addresses used by the RTI incorrectly had the transient flag bit set to 0. 47522
- Fixed issues with displaying synchronization points in the RTI Assistant. 47571
- Corrected an error in *rtiSimpl13.cxx*. 48211
- The `simpleTime` example federate did not always shut down gracefully. 48357
- The RTI was not processing the `CallbackModel` parameter that was specified in the `connect()` service call of HLA Evolved federates. 48425

- ♦ If an HLA Evolved federate attempted a `joinFederationExecution()` without itself calling `createFederationExecution()` it would be unable to find *HLAstandardMIM.xml*. 48484
- ♦ Fixed an error that occurred when a full path was specified for an HLA 1516-2010 FOM module. 48509
- ♦ Fixed an exception that could occur when using the `getTransportationTypeHandleFactory()` service call in the HLA 1516-2010 Java bindings. 48534
- ♦ Fixed the encoding and decoding of the `TransportationTypeHandle` in the Java bindings for HLA 1516-2010. 48552
- ♦ Fixed an issue with processing HLA 1516-2010 FOM modules with blank semantics fields. 48557
- ♦ The RTI incorrectly required dimensions to be defined in every HLA 1516-2010 FOM module in which they were used. 48557
- ♦ The RTI crashed when no `dataTypes` section was included in an HLA 1516-2010 FOM module. 48557
- ♦ Fixed a "run-time check failure" that occurred with VC++ 10 when joining a federation. 48565
- ♦ The RTI did not properly handle an empty switches table in HLA 1516-2010 FOM modules. 48569
- ♦ The example federates would not build on Windows when there was a space in the RTI installation directory. 48582
- ♦ The HLA 1516-2010 service call `enableTimeRegulation` would throw the exception `InvalidLogicalTime` when it should have been throwing `InvalidLookahead`. 48600
- ♦ In HLA Evolved, `HLAfloat64Interval` and `HLAinteger64Interval` were throwing `InvalidLogicalTime` when they should have been throwing `InvalidLogicalTimeInterval`. 48601
- ♦ The RTI was not always successful in differentiating an HLA 1516-2000 format FDD file and a 1516-2010 format FDD file. 48666
- ♦ The HLA Bounce example no longer has a dependency on Qt libraries with non-standard names, allowing you to build it against any compatible standard installation of Qt. 48683
- ♦ The operator `<<` for HLA 1516-2010 handle classes did not actually output the handle value. 48713
- ♦ The HLA 1516-2010 `initiateFederateRestore` `FederateAmbassador` callback was always specifying a blank `FederateHandle`. 48714
- ♦ The Java bindings did not handle FDD files that were specified as relative paths with spaces in them. 48734
- ♦ The Java bindings did not handle FDD files that were specified as relative paths with spaces in them. 48735

- ♦ Fixed a crash that would occur when using the reserveMultipleObjectName HLA 1516-2010 RTIAmbassador service with a large number of names. 48745
- ♦ Fixed a crash that occurred at startup on Linux machines when the network cable was unplugged. 48822
- ♦ A plug-in that subclasses DtRtiAmbassadorImplementor did not work because RID parameters are parsed after the logic to createRtiAmbassadorImplementor is done. 49381
- ♦ The rtiexec would fail to perform FED file distribution on Windows if the TMP environment variable was not set. 49442
- ♦ The getKnownObjectClassHandle service now returns the correct known class (for example, the class the object was discovered as) instead of the actual class (for example, the class the object was registered as) if an object was discovered via a subscription that was not the actual class (for example, a parent class). 49471
- ♦ Federate handles from previously joined federates can now be submitted to the normalizeFederateHandle service call. 49472
- ♦ The getFederateHandle and getFederateName methods will now return valid information for the federate's own name and handle. 49474
- ♦ Appropriate MOM interactions are now allowed after a save/restore has been initiated (for example, HLAinteractionRoot.HLAmanager.HLAfederate.HLAservice.HLAfederateSaveBegun). 49476
- ♦ Multiple object name reservation will now allow unreserved names to succeed even when already reserved names exist in list. The presence of any illegal names (beginning with HLA) still causes exception and failure of entire set. 49477
- ♦ The RTI Assistant did not properly save its settings when the machine was shut down. 49490
- ♦ The RTI would not process HLA 1516-2010 FOM modules that included transportation types other than HLAReliable or HLABestEffort. 49030
- ♦ Fixed an issue that occurred when merging *HLAstandardMIM.xml* with HLA 1516-2010 FOM modules that included MOM dimensions. 49102
- ♦ Fixed an error that occurred with DTD validation when using HLA 1516-2000 with VC++ 9. 49191
- ♦ Fixed an rtiexec crash that would occur on some systems with active VPN connections. 47665
- ♦ Fixed a forwarder crash that sometimes occurred when a connected forwarder unexpectedly disconnected. 49546
- ♦ Federates that subscribed to the base class of a subclass that was added in a module after they joined would not always discover objects or receive interactions of that subclass. 49661

Known Problems

This release has the following known problems:

- ♦ On Windows, the MÄK RTI Java Bindings do not work with Eclipse or Netbeans IDEs when the Java 1.7 JRE, JDK, or updates are installed. When Java 1.7 is installed, it results in the following error message:

```
Exception in thread "main" java.lang.UnsatisfiedLinkError:  
C:\MAK\makRti4.2\bin\makRtiJava1516e.dll: The specified procedure could  
not be found
```

In order to use an IDE with the MÄK RTI Java Bindings, use Java 1.6. 49638

- ♦ If multicast is disabled on a network, the RTI Assistant network map and federations table may not display properly. This problem does not affect the functioning of the RTI, just the display in the RTI Assistant.
- ♦ The RTI Assistant's RTIspy window can display network and performance statistics for any federate in your federation that has **RTI_enableLRCNetworkStatisticsMonitoring** enabled in its RID file. This monitoring capability is implemented by a plugin to the LRC. The RTIspy can only have one plug-in loaded at a time. Therefore, the RTIspy window will not work if you are using your own LRC plugin.
- ♦ Shared memory does not work with 64-bit Linux federates.
- ♦ Some computers running Windows Vista may experience problems receiving IPv6 multicast messages. This can affect communication between RTI Assistants, which makes it impossible to detect active connections on your network, and best-effort communications between federates.
- ♦ 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 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.

- ♦ 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 10.6, “[Configuring RTI Forwarders for Distributed Forwarding](#)” 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.