



MÄK RTI 4.1 Release Notes

This document provides the following release-specific information for MÄK RTI 4.1:

Supported Systems	2
Using Libraries and Binaries Built with Visual Studio 2005 and Later	2
Compiler Compatibility on Windows	3
License Manager	3
Third Party Library Dependencies	3
Patch Required for AMD Dual-processor Windows PCs	4
Backwards Compatibility	4
Network Compatibility	4
Compatibility with Other RTIs	5
New Features and Updates	6
Support for IPv6	6
Changes to RID Parameters	6
Bug Fixes	8
Known Problems	9
RTIspy Diagnostic GUI Scalability	10

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 © 2011 VT MÄK, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, B-HAVE®, RTIspy®, and VR-Link® are registered trademarks of VT MÄK. Document ID: RTI-4.1-2-120116

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 4.1. 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 4	Default compiler.
Red Hat Enterprise Linux Workstation 5. (32 bit and 64 bit libraries)	Default compiler.
SUSE 11 (32 bit and 64 bit libraries)	Default compiler.
Windows XP	Microsoft Visual C + + 7.1, 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.	

Using Libraries and Binaries Built with Visual Studio 2005 and Later

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, the C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later cannot 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 distributes the Windows installer for the C runtime libraries with all MÄK products released for MSVC++8.0 and later. The 32-bit installer is named *vcredist_x86.exe*; the 64-bit installer (if supported) is named *vcredist_x64.exe*. They are in the base directory of any installed MÄK product that requires them.

Running the installer requires Administrator privileges for the machine the installer is run on.

For more information see this Microsoft URL:

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



You must ensure that the preprocessor defines `_SECURE_SCL=0`, and `_HAS_ITERATOR_DEBUGGING=0` are set for MSVC++8.0 and MSVC++9.0 builds. If these are not set, random crashes and assertions may be encountered during runtime. The MSVC++ 10.0 version uses the default values for these flags.

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 4.1 uses FLEXlm 11.8 for all versions except the Windows VC++ 7.1 version, which continues to use FLEXlm 11.6. 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>
- 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

In addition to the Qt dependencies mentioned previously, the MÄK RTI 4.1 has the following dependencies:

- QT - MSVC++ 10 version uses QT 4.7.4. All other versions use QT 4.5.
<http://qt.nokia.com/>
- 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/>

Patch Required for AMD Dual-processor Windows PCs

VR-Link-based products use a high resolution counter for time calculations on Windows PCs. Customers who are running Windows on PCs with multiple AMD Athlon 64-bit processors may notice clock jitter, which may cause time in MÄK products to run backwards. This occurs when the Windows scheduler changes the CPU the MÄK process is using. If the high resolution counters on each processor are not synchronized, the application may witness a decrease in the high resolution counter value stored in the processor causing an incorrect time calculation. To fix this problem customers, apply the AMD Dual-Core Optimizer patch provided by AMD. You can get the patch at:

http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_871_9706,00.html



If you get an error when you try to access this URL, reload the page.

Backwards Compatibility

The MÄK RTI 4.1 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 4.1 and some use a previous version of the MÄK RTI.) However, except as noted below, the MÄK RTI 4.1 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 MAK 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 MAK 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 4.1. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 4.1 would be link compatible.

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

New Features and Updates

MÄK RTI 4.1 has the following new features and changes:

- ♦ Added support for IPv6.
- ♦ New example Java federate.
- ♦ Bundling and compression can now be configured separately for individual links in an RTI network.
- ♦ Multiple load-balanced forwarders can run on the same computer.
- ♦ Support for SUSE 11.
- ♦ Ability to configure different ports for best effort and reliable transport.
- ♦ Changes to RID file configuration parameters.

Support for IPv6

The RTI now supports the latest version of the Internet Protocol, IPv6. When using IPv6, you can configure your connection using either the RTI Assistant or the RID file. Simply specify an IPv6 address instead of an IPv4 address for any parameter taking an IP address. However, you cannot mix IPv4 and IPv6 in a single federation. For example, you cannot set your `RTI_tcpForwarderAddr` to an IPv6 address and your `RTI_destAddrString` to an IPv4 address in order to send reliable messages via IPv6 and best-effort messages via IPv4. The exception to this rule is the `RTI_webServiceAddr` parameter. The RTIspy web server only supports IPv4, so this parameter must always be configured using an IPv4 address.

Changes to RID Parameters

The following parameters have been removed:

- ♦ `RTI_momExceptionReports` and `RTI_momServiceReporting`. They were replaced with `RTI_exceptionReportingSwitch` and `RTI_serviceReportingSwitch` in a previous release.
- ♦ `RTI_fomDataReliableWhenUsingRtiexec` and `RTI_forceFomDataReliable`. They have been replaced with `RTI_fomDataTransportTypeControl`.

The following parameters have been added:

- ♦ `RTI_tcpPort`. Specifies the port number to be used for TCP messages. This parameter is ignored unless `RTI_configureConnectionWithRid` is set to 1. This is a federation-wide parameter. Default: 4000.
- ♦ `RTI_logFileDirectory`. Specifies a directory where the log file should be output. This is an LRC-specific parameter. Default: "" (the working directory).
- ♦ `RTI_packetBundlingSize`. When packet bundling is enabled, specifies the size of the message bundles for TCP packets and UDP datagrams. The value should be $\leq$ MTU of the network. This is an LRC-specific parameter. Default: 0.

- ♦ **RTI_fomDataTransportTypeControl.** This parameter sets the FOM Data Transport Type Controller as follows:
 - 0 – FOM data is sent either best effort or reliable based on FOM settings. This is the default. It is forced when **forceFullCompliance** is enabled.
 - 1 – All FOM data is sent best effort. It is the forced setting when using light-weight configurations.
 - 2 – All FOM data is sent reliable. This is permitted only when **RTI_configureConnectionWithRid** is 1.
- ♦ **RTI_throwExceptionOnConcurrentAccess.** In HLA 1.3, controls whether the ConcurrentAccessAttempted exception is thrown when an RTIambassador service call is called while concurrently processing another service call. This usually happens when a FederateAmbassador callback calls another RTIambassador service call during a tick().
 - 0 – Do not throw exception.
 - 1 – Throw exception.

This is an LRC-specific parameter. Default: 0.
- ♦ **RTI_allProvideUpdateRequestsDelayed.** Specifies that **RTI_autoProvideDelay** should apply to all update requests, not just those generated due to the autoProvide switch. This is an LRC-specific parameter. Default: 0.
- ♦ **RTI_tickWaitPeriod.** Determines the smallest increment of time that a tick will cause a federate to wait for messages from the I/O thread when using asynchronous I/O. If set to 0.0, the minimum time specified in the tick service invocation is used instead. This is an LRC-specific parameter. Default: 0.0.

The following parameters have changed:

- ♦ **RTI_udpPort.** Only affects UDP, not TCP.
- ♦ **RTI_enableTcpCompression.** Now LRC-specific.
- ♦ **RTI_tcpCompressionLevel.** Removes 10 as a valid compression level. LRC-specific.
- ♦ **RTI_enableUdpCompression.** LRC-specific.
- ♦ **RTI_udpCompressionLevel.** Removes 10 as a valid compression level. LRC-specific.
- ♦ **RTI_enablePacketBundling.** This parameter now just enables bundling. The bundle size is set by **RTI_packetBundlingSize**. LRC-specific.
- ♦ **RTI_minChannelAddr.** Default: 224.0.1.0 if using IPv4; ff02::3:1 if using IPv6.
- ♦ **RTI_maxChannelAddr.** Default: 239.255.255.255 if using IPv4; ff02::3:ffff if using IPv6.

Bug Fixes

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

- ♦ Fixed an issue that sometimes caused a federate to fail to connect to an RTI Assistant that it had just started. 42579
- ♦ Improved error messages when the RTI encountered an error processing a FED file. 42715
- ♦ HLA Evolved exceptions were not always thrown in proper order of precedence. 42744
- ♦ The RTI installer and RTI Chooser would sometimes hang when updating from older RTI versions. 42768
- ♦ Connecting distributed RTI Forwarders through the RTI Assistant connection configuration dialog boxes did not work. 43357
- ♦ Distributed RTI Forwarders reported warnings when federates were using DDM. 43358
- ♦ In the HLA Bounce example, the region box was displayed even after the federate had resigned from the federation. 43360
- ♦ The RTIspy web service was not working for MSVC++ 10.0. 43814
- ♦ RTI Assistants on computers with multiple NIC cards tried to start multiple rtiexecs when a remote RTI Assistant requested rtiexec start up. 44261
- ♦ When the configured RTI Assistant port is already in use by another application, the federate will now fallback to not using an RTI Assistant and reporting a warning. 44525
- ♦ When a federate is force to resign from the RTI Assistant Federations View, it will now throw an exception the next time the federate ticks the RTIambassador. 45227
- ♦ The RTI Assistant no longer flashes a console window when starting up on Windows. 45335
- ♦ The built in RTI latency tests timed out for HLA Evolved federates. 45525
- ♦ The HLA 1.3 version of the HLA Bounce example did not allow multiple region extents. 45541
- ♦ The processing of the RTI_RID_FILE token to specify an alternative RID file is now supported in the HLA Evolved local settings designator passed to the RTI Ambassador connect method. The RTI_RID_FILE designator must appear first in local designator string and the filename must be terminated with a '#' character. 46477
- ♦ The HLA Bounce example was not encoding the attributes for the Ball class as specified in the FED file. 46685
- ♦ The rtiexec would sometimes crash when over 170 federates joined the federation. 46962
- ♦ Fixed the simpleDDM GUI to fix some sizing issues on Linux platforms. 42581
- ♦ When running a Java federate, if an rtiexec was launched from the RTI Assistant Choose RTI Connection dialog, the rtiexec would crash. 45545

Known Problems

This release has the following known problems:

- ♦ The RTIspy web service does not work with non-English web browsers. To work around this issue, you can install a locale switcher plug-in for your browser which will tell the RTIspy page that the browser is an English locale.
- ♦ 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 11.6, “[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.
- ♦ The RTIs Spy browser-based GUI does not work well with Internet Explorer. While you will be able to see and use the RTIs Spy GUI with IE7, the page layout may be problematic, and some information may not be displayed correctly. The network map does not work in the default version of IE8. If you are using IE8, you may be able to view the network map in compatibility mode. For optimal use, MÄK recommends using Firefox (2.0 or later).

RTIs Spy Diagnostic GUI Scalability

The RTIs Spy diagnostic GUI provides helpful information and statistics that can be useful when debugging issues in federations with low federate counts, object counts, and update rates. It does not scale well to larger federations or federations with high frequency updates of many objects. Please contact MÄK if you require tools for debugging larger federations.