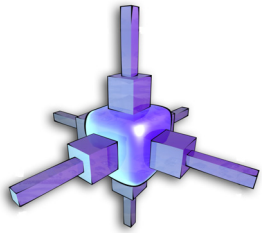

MÄK RTI 3.4 Release Notes

This document provides the following release-specific information for MÄK RTI 3.4:

MÄK 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).)

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 3.4. 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	Default compiler.
Windows XP	Microsoft Visual C++ 7.1, 8.0, 9.0**
Windows Vista*	Microsoft Visual C++ 8.0, 9.0**
*You must have administrator privileges to install VT MÄK products on Windows Vista.	
**The MSVC++ 9.0 version has 32 bit and 64 bit libraries.	

The RTI LS supports federates that run on the Solaris operating system. This version of the MÄK RTI only includes the LRC libraries. It does not include the rtiexec, RTIspy, or RTI Forwarder. You can link to the libraries if you are creating new federates on Solaris.

If you need to run an rtiexec, you must do so on Windows or Linux.

Table 2: RTI LS: Platforms supported

Operating System	Compiler
Sun Sparc Solaris 10	Sun C++ 5.8

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, with the release of the new C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later, 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 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. 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

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.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1, 8.0, and 9.0. 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 can result in program instability.

FLEXlm Support

MÄK RTI 3.4 uses FLEXlm 11.6. If you are upgrading from a version of the MÄK RTI that used an older version of FLEXlm, you must replace the files in the license server's FLEXlm directory with the files provided with this release. You do not need a new license. Licenses are forward compatible.

Third Party Library Dependencies

In addition to the Qt dependencies mentioned previously, the MÄK RTI 3.4 has the following dependencies:

- 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 3.4 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 3.4 and some use a previous version of the MÄK RTI.) However, the MÄK RTI 3.4 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.) 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 API[1] 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 3.2, "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 3.4. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 3.4 would be link compatible.

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

New Features and Updates

MÄK RTI 3.4 is a feature release. It has the following new features and updates:

- Improved ease-of-use
  - Connections now have names
  - Connections can be edited
  - Lightweight connections are known
  - Lightweight connections can conflict
  - RTI Forwarders can be configured in the RTI Assistant
  - The RTI Assistant provides status information about connections.
- The rtiexec GUI has been removed. The rtiexec always runs as a daemon. The information previously provided by the rtiexec GUI is now displayed in two windows that are accessible from the RTI Assistant menu, the RTI Federations View and the RTI Network Components view. For details, please see Chapter 6, *The RTI Assistant*, in *MÄK RTI Users Guide* and throughout the updated procedures in *MÄK RTI Users Guide* and *MÄK RTI Reference Manual*.
- Remote startup of the rtiexec. You can start rtiexecs on remote computers. For details, please see "Starting the rtiexec", in *MÄK RTI Users Guide*.
- The rtiexec no longer has a built in RTI Forwarder. The RTI Forwarder always runs as a separate application. For details, please see Section 5.3, "The RTI Forwarder Application", in *MÄK RTI Reference Manual*.
- The RTI Assistant is persistent. You can configure the RTI Assistant to start automatically when you log in. For details, please see "Configuring the RTI Assistant's Startup Behavior", in *MÄK RTI Users Guide*.
- RTI Command Line Tool. The Command Line Tool is a non-interactive way to get information from the MÄK RTI and send commands to it. For details, please see Section 5.5, "The RTI Command Line Tool", in *MÄK RTI Reference Manual*.
- New and removed RID parameters
- A new manual, *MÄK RTI Users Guide*, contains basic user-level instructions for running federates with the MÄK RTI. Advanced configuration and developer-level documentation is still in *MÄK RTI Reference Manual*.
- The rtiexec `-C` command-line option has been replaced with `[-M | --manual]`. This option works exactly the same as `-C` did.

Ease-of-Use Features

The ease-of-use improvements begun in MÄK RTI 3.3 have been extended. The RTI Assistant is now the primary user interface for the MÄK RTI. The rtiexec GUI has been removed. Its functions are now implemented in the RTI Federations View and the RTI Network Components View.

- Connections now have names. When you create a connection, you can give it a name for easy reference. For details, please see "Managing Connection Configurations", in *MÄK RTI Users Guide*.
- Connections can now be edited. In MÄK RTI 3.3, you could create and remove connection, but could not edit them. For details, please see "Editing Connections", in *MÄK RTI Users Guide*.
- The RTI Assistant tracks the status of lightweight connections.
- Lightweight connections can conflict with other connections. If a lightweight connection uses a specific port and a specific multicast channel, other connections cannot use that port and address, even if the rtiexec is used.
- The RTI Forwarder uses connections. In MÄK RTI 3.3, the RTI Forwarder ran as a thread in the rtiexec or and independent application started from the command line. It did not use the RTI Assistant. In release 3.4, when you start an independent RTI Forwarder, it uses the same startup process as the rtiexec or any federate.
- The RTI Assistant provides status information about connections. The choose connection dialog boxes show the status of connections using colored arrows and a "Do Not Enter" icon. The Connections page in the RTI Preferences dialog box shows status with colored icons. Tooltips provide additional information about connections status. For details, please see "Connecting a Federate to the MÄK RTI" and "Viewing Connection Status", in *MÄK RTI Users Guide*.

New, Changed, and Obsolete RID Parameters

The following parameters have been added to *rid.mtl*:

RTI_rtiForwarderLogFileName specifies the RTI Forwarder log file name.

RTI_distributedForwarderPort specifies the port used to communicate between RTI Forwarders. Default: 5000.

RTI_dumpRid specifies whether or not to dump the RID parameters used to the log file. Default: 0 (do not dump parameters).

RTI_enableSaveRestoreWhenUsingRtiexec enables (1) or disables (0) the save and restore services when an rtiexec is being used. Default: 1. This parameter is forced to 1 when **RTI_forceFullCompliance** is set.

The following parameters have been removed from *rid.mtl*:

RTI_rtiExecHasTcpForwarderThread. All RTI Forwarder instances are run as separate applications.

RTI_enableRtiexecGUI. The rtiexec no longer has its own GUI.

The default values for the following parameters have changed:

- **RTI_mcastDiscoveryPort** — 6001
- **RTI_enableRtiexecWebservice** — 0
- **RTI_webserviceRtiPort** — 6002.

Changes to the Routes File

The **RTI_distributedForwarderPort** parameter has been removed from the routes file. It is now specified in *rid.mtl*. **RTI_addInterfaceAddr** was removed. All local interfaces are automatically added.

Documentation Changes

A new manual, *MÄK RTI Users Guide*, has been added to the MÄK RTI documentation set. This manual focuses on users who just want to run their federates with the MÄK RTI with a minimum of attention to configuration and network topology issues. *MÄK RTI Reference Manual* has been extensively revised.

- Section 11.4 in the print version of *MÄK RTI Reference Manual* has incorrect cross references to figures. The correct text is:

  "(Figure 11-3) shows a distributed forwarding network. This figure is essentially identical to Figure 11-2."

- The way that **RTI_conveyRegions** works has changed. The updated description is:

  Configures the RTI to provide associated region information in the reflect attribute values and receive interaction services. In HLA 1516, enabling this parameter overrides the setting of the of the Convey Region Designator Sets Switch in the FDD file. If this parameter is disabled the values specified by the FDD switch are used. Since the FDD parameter affects the entire federation, this RID parameter allows you to selectively enable this switch at federates of your choosing.

  If this parameter is disabled in HLA 1.3, no region information is populated in the Attribute Handle Value Pair Sets or Parameter Handle Value Pair Sets included in the reflect attribute values and receive interaction services, respectively. This can improve performance slightly if region information is not required by your federate.

  The possible values for this parameter are:

  0 — Do not convey regions.

  1 — Convey Regions.

  Default: 0.

Bug Fixes

The following bugs have been fixed in this release:

- The RTI sometimes crashed when subscribing or unsubscribing to a DDM region.
- The RTI Forwarder crashed when using a hierarchical forwarder configuration.
- The RTI Forwarder sent unnecessary warnings about an "Unrecognized node address".
- Fixed a bug in ownership transfer that occurred when a federate invoked unconditionalAttributeOwnershipDivestiture and attributeOwnershipAcquisition without ticking the RTI between the calls.
- When using DDM and some of the attributes of a registered object went out of scope, the turnUpdatesOnForObjectInstance and turnUpdatesOffForObjectInstance federate ambassador callbacks were being called for all published attributes of the object, not just those attributes that changed scope.
- Federates sometimes received and processed messages from other federations when running multiple federations in a forwarder group configuration.
- The RTI Assistant tray icon context menu sometimes did not appear when right-clicking on the icon.
- The RTI crashed when processing DDM region association messages.
- The RTI crashed if two RTI ambassadors were created in the same application and the first used a configuration with DDM enabled and the second did not.

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 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 RTIspy browser-based GUI does not work well with Internet Explorer 6 or 7. While you will be able to see and use the RTIspy GUI with IE7, the page layout may be problematic, and some information may not be displayed correctly. For optimal use, MÄK recommends using Firefox (2.0 or later).
- Using the MÄK RTI on Fedora 7 machines that have Xen Virtualization software installed causes a problem with multicast discovery. This is because INADDR_ANY maps to the virtual interface rather than the default, eth0. To resolve this issue, set **RTI_networkInterfaceAddr** to the IP address of the real ethernet card.