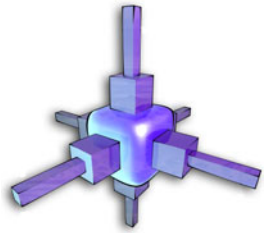

MÄK RTI 3.3 Release Notes

This document provides the following release-specific information for MÄK RTI 3.3:

Document ID: RTI-3.3-2-081015

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 3.3. 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.
Red Hat Fedora 7	Default compiler.
Windows XP	Microsoft Visual C++ 7.1, 8.0
Windows Vista	Microsoft Visual C++ 8.0

The RTI LS supports federates that run on the Solaris and IRIX operating systems. (Please see *MÄK RTI Release Notes* for specific versions supported.) This version of the MÄK RTI only includes the LRC libraries. It does not include the rtiexec, RTIspy, or distributed forwarder. You can link to the libraries if you are creating new federates on IRIX or 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
SGI IRIX6.5	MipsPro7.4.2m C++ compiler
Sun Sparc Solaris 10	Sun C++ 5.8

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.

Unfortunatly, 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.

Therefore, if you 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 3.3 uses FLEXlm 11.4 for Windows, Linux, and SGI IRIX. Sun Solaris uses FLEXlm 9.2.

Third Party Library Dependencies

In addition to the Qt dependencies mentioned previously, the MÄK RTI 3.3 has the following dependencies:

- QT - 4.4.1 - http://www.trolltech.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/
- VR-Link 3.12
- EHS - 1.3.1 (modified) http://xaxxon.slackworks.com/ehs/
- PCRE - 6.4 (modified)) http://www.pcre.org/
- PME - 1.0.4 (modified) http://xaxxon.slackworks.com/pme/index.html
- PTHREADS - 2.7.0 http://sourceware.org/pthreads-win32/.

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.

RTIspy Web-based GUI Configuration

The RTIspy has the following system considerations:

- The RTIspy is optimized for screen resolutions of 1024x768 and larger. For smaller screen sizes, additional window scrolling will be required.
- The RTIspy 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.

Backwards Compatibility

The MÄK RTI 3.3 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 3.3 and some use a previous version of the MÄK RTI.) However, the MÄK RTI 3.3 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 4.2, "Compiling and Linking with the 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 3.3. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 3.3 would be link compatible.

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

New Features and Updates

MÄK RTI 3.3 has the following new features and updates:

- New user assistance features
- Ability to choose the RTI connection to use at rtiexec and federate startup
- New rtiexec graphical user interface
- Centralized license checkout using the rtiexec
- Latency testing
- Elimination of the RTI Plus licensing
- RTI Chooser lets you configure the RTI version and RID file to use (Windows only)
- Ability to enable rtiexec and LRC logging at runtime
- Changes to unlicensed mode (trial mode)
- New, changed, and deleted configuration parameters
- Updated implementation of logical time for HLA 1516
- Updated command line arguments for the rtiexec and RTI Forwarder.

New User Assistance Features

The usability of the MÄK RTI has been improved in a variety of ways. Many of these improvements are incorporated in the redesigned rtiexec GUI. Others have been encapsulated in utility applications called the RTI Assistant and in the RTI Chooser.

The RTI now monitors connection and license checkout and provides helpful dialog boxes when problems are detected. You can start the RTIspy web-based GUI from the rtiexec. You can enable message logging from the RTI Assistant for the rtiexec and individual federates. You can even configure the rtiexec to manage license checkouts. This means you do not have to configure license management at individual computers. These new features are described in more detail in these release notes and in *MÄK RTI Reference Manual*.

New rtiexec Graphical User Interface

The rtiexec GUI is the default interface for the rtiexec. It has been completely updated. In addition to all the features it previously had, such as viewing federation information and time management information for federates, it now displays a console for rtiexec messages and allows you to launch the RTIspy and latency testing. For complete details, please see *MÄK RTI Reference Manual*.

Ability to Configure RTI Connections

MÄK RTI 3.3 provides a new way to specify basic connection information. You can now do this through the GUI, instead of (and in preference to) using the RID file. For most RTI configurations you can configure the RTI connection settings (port and multicast address) when you start a federate. The Choose RTI Connection dialog box allows you to start an rtiexec, connect to an rtiexec that is already running, or use a lightweight mode connection. For complete details about connection configurations, please see Section 5.3, "Managing Connection Configurations" in *MÄK RTI Reference Manual*.

Centralized License Checkout Using the rtiexec

The MÄK RTI rtiexec can check out licenses for federates. When this feature is enabled, the federates do not need to have access to a license server or otherwise configure license management. This can be quite helpful when a federation has federates located on multiple LANs whose security configuration makes it difficult for a federate to connect to a license server. For details, please see Section 2.3.12, "Managing Licenses for the MÄK RTI" in *MÄK RTI Reference Manual*.

Latency Testing

The new latency testing feature lets you test the amount of time it takes to send an interaction to a federate and receive a response. For details, please see Section 9.2, "Testing Latency Between Federates" in *MÄK RTI Reference Manual*.

RTI Chooser

If you have more than one version of the MÄK RTI installed, you can use the RTI Chooser to quickly change which one to use without editing your path. The environment is changed by the RTI Chooser. You can also specify which RID file to use with a particular RTI version. The RTI Chooser is an off-line utility. It is only available on Windows. For complete details about connection configurations, please see Section 5.2, "Configuring the RTI Version and RID File to Use" in *MÄK RTI Reference Manual*.

Elimination of the RTI Plus Licensing

All RTI features, including the rtiexec GUI, RTI Forwarder, and RTIspy are now included in the standard MÄK RTI license. Additionally, features such as the rtiexec and RTI Forwarder do not require a license to run and, therefore, can be run on any platform supported by the RTI.

Ability to Enable rtiexec and LRC Logging at Runtime

Previous versions of the MÄK RTI required you to enable rtiexec and federate logging in the RID file. This option is still available. However, you can now enable logging on a per-federate basis at runtime. This allows you to better manage the additional overhead the logging messages creates. You can specify the notification level and can save messages to a file. For details, please see Section 6.4, "Viewing Notification Messages from the rtiexec" and Section 8.7, "Viewing LRC Diagnostic Messages" in *MÄK RTI Reference Manual*.

Changes to Unlicensed Mode

Trial mode has been renamed unlicensed mode. Unlicensed mode is now available for Linux.

In unlicensed mode, the RTI will now prevent federates from joining after the federate count limit (two federates) has been reached. In the past if you ran more than two federates in unlicensed mode, their behavior was unpredictable (for example, federates receiving updates from one federate but not another). Now, the RTI will cause the joins to fail and generate a notification.

If you use the rtiexec, the third federate will fail to join a federation with two joined federates. In lightweight mode, the join service call will succeed but the third federate will be forced to resign. It is now possible to resign one federate from a 2 federate federation and have another federate take its place.

Unlicensed mode federates and licensed federates will no longer be able to communicate with each other.

HLA 1516 Logical Time

The integer implementation of 1516 logical time has been changed from a 32-bit integer to 64-bit integer to better comply with the IEEE 1516 interface specification. The preferred name for this implementation is "RTI_Integer64Time" as specified in the 1516 specification. The original "unsigned integer" name can still be used for this implementation; however, the name is deprecated and will be phased out eventually.

New, Changed, and Deleted Configuration Parameters

The following parameters have been added:

- RTI_sharedMemExecLogFileName enables logging shared memory exec output to a file with the specified name. Default = *shmExec.log*.
- RTI_enableFederateHearbeat enables the federate heart-beating service in the RTI, which is configured via the RID parameters RTI_federateHeartbeatInterval and RTI_federateTimeoutInterval. Default 0 (disabled).
- RTI_enableBestEffortSendRetry enables the automatic resending of failed attempts to send best effort messages. Configured via RTI_bestEffortSendRetryWaitUsec and RTI_bestEffortSendRetries. Default 0 (disabled).
- RTI_enableMessageThrottling enables the throttling of transmission and receipt of internal messages as well as the buffering of provideAttributeValueUpdate callbacks in HLA 1516. Configured via RTI_transmitDelay, RTI_transmitRate, and RTI_autoProvideDelay. Default 0 (disabled).
- RTI_detachNotifyLevelFromStdOut detaches all notification messages at the given level and above from the standard output. This setting has no affect on logging notifications to a file. Default 5 (no notification levels detached).
- RTI_rtiExecPerformsLicensing enables license checkout from the rtiexec. The rtiexec will checkout a license for each joining federate. Default 0 (disabled).
- RTI_processingModel sets the synchronous/asynchronous IO operating mode. (It replaces individual processing mode parameters - RTI_asynchronousIO, RTI_asynchronousProcessMessage, and RTI_asynchronousCallbacks). Asynchronous IO configuration (e.g., RTI_IOPeriod) is used for the asynchronous processing models.
  - 0 – Synchronous mode - no asynchronous processing, default
  - 1 – Asynchronous IO - Perform network reads and writes in an asynchronous thread
  - 2 – Asynchronous Message Processing - In addition to performing network reads, the asynchronous thread will process messages. However, it will not perform any asynchronous federate ambassador callbacks. The callbacks will be cached and delivered when the federate calls tick or evokeCallbacks.
  - 3 – Asynchronous Callbacks - Process messages and invoke callbacks in an asynchronous thread.

  Default: 1 (asynchronous IO).
- RTI_lrcMcastDiscoveryDelay - Specifies the interval that the LRC will wait for a multicast discovery response (replaces RTI_mcastDiscoveryInterval). Default 3.0.
- RTI_execMcastDiscoveryDelay - Specifies the interval that the LRC will wait for a multicast discovery response (replaces RTI_mcastDiscoveryInterval). Default 1.0.

The following parameters have been renamed:

- **RTI_internalMsgReliable**. This parameter has been renamed **RTI_internalMsgReliableWhenUsingRtiexec** and is ignored unless **RTI_useRtiExec** is set to 1. Default 1 (enabled).
- **RTI_fomDataReliable**. This parameter has been renamed **RTI_fomDataReliableWhenUsingRtiexec** and is ignored unless **RTI_useRtiExec** is set to 1. Default 1 (enabled).
- **RTI_forceTrialVersion** is now **RTI_forceUnlicensedForTwo**.
- **RTI_disableTrialVersion** is now **RTI_disableUnlicensedForTwo**.

The following parameters have been removed:

- **RTI_enablePopUpErrorMsgs**. The messages are now sent to the RTI Assistant and cannot be disabled.
- **RTI_mcastDiscoveryInterval** has been replaced by **RTI_lrcMcastDiscoveryDelay** and **RTI_execMcastDiscoveryDelay**
- **RTI_asynchronousIO**, **RTI_asynchronousProcessMessage**, and **RTI_asynchronousCallbacks** have been replaced with **RTI_processingModel**, which is described earlier in this section.
- **RTI_enableRtiexecGUIConsoleLog**.

Updated Command Line Arguments

The following RTI executables have new or changed command line arguments:

- rtiexec: for details, please see Section 6.2, "Starting the rtiexec" in *MÄK RTI Reference Manual*
- RTI Forwarder: Section 10.3, "Running the RTI Forwarder Application"
- Shared memory manager: Section 17.3.1, "Shared Memory Queue Manager Command-Line Options"
- RTIspy web server: "Running the Web Service in Stand-Alone Mode (Central Server)", in Section 14.2.3, "Local vs. Centralized HTTP Servers".

Documentation Updates

MÄK RTI Reference Manual has been updated and reprinted for this release. Online help for the rtiexec has been updated. The following section is a supplement to Section 10.2, "Configuring the RTI Assistant", in *MÄK RTI Reference Manual*.

Starting the RTI Assistant

The RTI Assistant starts automatically when you start the rtiexec or the first federate, so ordinarily, you do not need to start it directly. However you might want to start it before you start a federate in order to change configuration settings.

➤ To start the RTI Assistant, choose **Start** → **Programs** → **MÄK Technologies** → **MÄK RTI** *x.x.x* → **Utilities** → **RTI Assistant**.

To start the RTI Assistant from the command line:

1. Change directory to the *./bin* directory.
2. Run:

   `rtiAssistant` *`options`*

   Table 3 describes the command-line options.

Table 3: RTI Assistant command-line options

Option	Description
`(--` | `--ignore_rest)`	Ignore all command-line options that follow this one.
`(-C` | `--Connection)`	Causes the RTI Assistant to immediately opens the RTI Preferences dialog box to the Connections page.
`(-h` | `--help)`	Displays help.
`(-K` | `--autoexit)`	Specifies that the RTI Assistant will automatically exit after 10 seconds without an active connection to an RTI component. This option is ignored if the RTI Assistant has been configured to be kept alive in the RTI Preferences dialog box.
`(-v` | `--version)`	Displays the MÄK RTI version.

Bug Fixes

The following bugs have been fixed in this release:

- A bug in the Java bindings caused an exception when using time management in a multi-threaded federate.
- Using smart forwarding, updates sent immediately after registering an object or transferring attributes are no longer missed by subscribers.
- Update advisories are now properly called when a region is modified such that attributes go in or out of scope.
- The MOM manager.federation federatesInFederation attribute is updated correctly.
- rtiexec daemon mode no longer fails to start when the RTI_CONFIG environment variable is set.
- If you are using one of the asynchronous processing modes, best effort messages that fail to be sent follow the send retry configuration (**RTI_enableBestEffortSendRetry**) and are dropped if they still cannot be sent. This behavior change prevents the send queue from being jammed by a message that fails to be sent.
- Reconnect did not work when there were multiple federations.
- When using local FED files during join (**RTI_distributeFedFile** disabled), the RTI will now use the same search procedure that it uses for create federation (search in the execution directory and then the directory found in RTI_CONFIG environment variable).

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.
- 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 13.7, "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.
- 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.
- The RTI Dump utility does not work directly with shared memory only mode. It can be used with shared memory if the shared memory mode relays messages to the network and the RTI Dump utility is connected via the network communications.
- Smart Forwarding does not work when using fixed grid DDM. The smart forwarding feature relies on the exchange of DM and DDM information in order to establish a correct destination list. Because fixed grid DDM does not exchange DM or DDM information, enabling smart forwarding causes no messages to be sent because there are no destinations.
- When merging FOM modules, the dimensions are taken from the create FOM module and no others are merged.
- 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 Core 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_mcastDevAddrString** to the IP address of the real ethernet card.
- If you use shared memory, you cannot use the RTI Assistant to configure a connection. To use shared memory, you must set the RTI_configureConnectionWithRid to 1.
- The Search feature in online help does not work. This is due to a bug in the Qt Assistant browser.

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.