MÄK RTI 4.4 Release Notes

This document provides the following release-specific information for MÄK RTI 4.4:

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).)

Document ID: RTI-4.4-2-150313

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 4.4. 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 5, 6. (32 bit and 64 bit libraries)	Default compiler.
CentOS - versions that are equivalent to supported versions of Red Hat Linux.	Default compiler.
Windows Vista*/Windows 7, WIndows 8	Microsoft Visual C++ 8.0, 10.0**, 11.0**

*You must have administrator privileges to install VT MÄK products on Windows Vista.

**The MSVC++ 10.0, and 11.0 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++ 8.0, 9.0, 10.0 and 11.0 (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 4.4 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 (for 32-bit systems)
  ftp://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz

A license manager FAQ is available at:

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

Third Party Library Dependencies

MÄK RTI 4.4 has the following dependencies:

- QT 4.7.4, except VC 11, which uses Qt 4.8.5.
- Qwt 6.0.1.
- FreeGlut 2.4.0
- Boost - 1.37 on all platforms except VC 10 and VC 11, which use 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 4.4 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 4.4 and some use a previous version of the MÄK RTI.) However, except as noted below, the MÄK RTI 4.4 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*).

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 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.

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*.

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

New Features and Updates

MÄK RTI 4.4 has the following new features:

- Significant improvement in performance.
- Support for translation files for the RTI Assistant. For details about using translation files, please see Section 2.6, "Localizing the Graphical User Interface", in *MÄK RTI Users Guide*.
- Support for merging FOM modules in lightweight mode.
- Support for removing the Shut Down All option on the RTI Assistant menu. TO do so, add the RTI_ASSISTANT_REMOVE_SHUT_DOWN_ALL environment variable.
- Executables such as the rtiAssistant, rtiexec, rtiForwarder, rtiChooser, and rti now support generating .DMP files on Windows and generating stack traces on Linux platforms if an unknown error occurs.
- RID file parameters have been remove, added, and updated.

RID File Parameter Changes

There is one new parameter – RTI_strictHLA1516eAttributeOwnershipUnavailableCallback. Enabling this parameter may cause attributeOwnershipUnavailable to not occur in federates with pending acquisitions in response to the cancelNegotiatedAttributeOwnershipDivestiture and confirmDivestiture service calls when the federate does not receive the requested attributes.

The following parameters have been removed:

- RTI_variableLengthDataUsesNull
- RTI_IOLockQueue.

The following parameters have new default values:

- RTI_tcpBufferSize
- RTI_dualTransmitFirstInteractionRegions
- RTI_socketReceiveBufferSize
- RTI_socketSendBufferSize
- RTI_enableBestEffortSendRetry
- RTI_bestEffortSendRetries
- RTI_bestEffortSendRetryWaitUsec
- RTI_enablePacketBundling
- RTI_packetBundlingSize
- RTI_IOPeriod
- RTI_maxIOQueue
- RTI_maxIOCount.

Documentation Updates

Documentation has been updated for this release.

Bug Fixes

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

- The RTI would throw an exception that was not allowed via the HLA standard's C++ API exception specification, which could cause an abort on Linux. 52060
- The rtiForwarder would use 100% CPU when 20+ federates disconnected at the same time. 54332
- Under some circumsnaces, evokeCallback would return before receving a federateAmbassador callback. 54091
- Two federates creating and joining the same federation at nearly the same time could cause an rtiAssistant message that would say "Federate could not join federation: FederationExecutionDoesNotExist" after successfully creating the federation. 53950
- Fixed an issue where calling getAttributeHandleFactory() from the HLA1516-2000 Java API would cause an Exception in thread "Thread-6" java.lang.UnsatisfiedLinkError: C:\MAK\makRti4.3\bin\makRtiJava1516_64.dll: Can't load AMD 64-bit .dll on a IA 32-bit platform. 53946
- When a federate using DDM services unsubscribed to attributes, a crash could occur. 53726
- The rtiAssistant could use 100% CPU in Redhat 5 and Redhat 6 even when doing nothing. 53467
- If the rtiexec was terminated unexpectedly, the federate could hang indefinitely using 100% CPU when the rtiAmbassador was destroyed. 53436
- The rtiForwarder will now look in the same directory as the RTI_RID_FILE for the routes file. If it doesn't find it it will look in the working directory. Previous versions of the forwarder only looked in the working directory. 53371
- The rtiForwarder could have a segmentation fault when logging was on with Smart Forwarding. 53356
- Creation of the best effort connection could fail when specifying a device other than 0.0.0.0. 53354
- Setting `(setqb RTI_use32BitsForValueSize 1)` in *rid.mtl* would cause interactions to be received with 0 parameters. 53240
- Performing a requestRestore then immediately resigning could cause a crash. 52954
- Setting **RTI_preferRidToFedSwitchTable** to 1 could cause FOM merging to raise rti1516e::InconsistentFDD where switches do not match. 52906

- The exception thrown for an HLA1516-2010 FOM merging error could result in a cryptic "Failed merging switches. Number of switches between FOM modules are not equal, 10 and 10" in the rtiAsssistant and the exception's what(). 52583
- When using Time Management, federates would incorrectly receive an update as RO instead of TSO. 52464
- In the C++ HLA1516-2010 encoding helpers, the assignment operator did not perform a deep copy and its implementation could go out of scope causing a crash. 52268
- Under some circumstances the MAK RTI would drop some best effort messages when being overwhelmed. The MAK RTI now correctly holds the message until the sending queue clears and then sends it. 50028
- The RTI Assistant default warning tick interval is now 60 seconds instead of 0 seconds. 0 seconds was an invalid value, since ticking is never instantaneous. 49951
- Fixed a failed join error if the RTIExec gets a join message before it has finished creating the federation. Federates will now wait a half second and try again until the RTIExec has finished creating the federation. 49676
- In the previous release, the MÄK RTI Java Bindings did not work on Windows with Eclipse or Netbeans IDEs when the Java 1.7 JRE, JDK, or updates are installed. The RTI now supports these IDEs with Java 1.8 and above. 49638

Known Problems

This release has the following known problems:

- The rtiForwarder may crash when using multiple rtiForwarders and when using routers, VPNs, or both, if the *rid.mtl* file specifies the same value for the **RTI_udpPort** and **RTI_tcpPort** parameters. 54545
- 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.