MÄK RTI 4.3 Release Notes

This document provides the following release-specific information for MÄK RTI 4.3:

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-4.3-2-140312

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 4.3. 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 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, WIndows 8	Microsoft Visual C++ 8.0, 9.0**, 10.0**, 11.0**
*You must have administrator privileges to install VT MÄK products on Windows Vista.	
**The MSVC++ 9.0, 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 4.3 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 4.3 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 4.3 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 4.3 and some use a previous version of the MÄK RTI.) However, except as noted below, the MÄK RTI 4.3 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*).

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*.

Because the DMSO RTI NG v6 was last built with the MSVC++ 6.0 compiler, it is not link compatible with the MÄK RTI 4.3. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 4.3 would be link compatible.

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

New Features and Updates

MÄK RTI 4.3 has the following new features:

- Performance improvements. Many performance improvements have been made to the MÄK RTI. Most of these are internal, so all improvements are gained when simply upgrading to MÄK RTI 4.3. Some improvements are done in the Local RTI Component (LRC) and this includes reducing the number of copies of objects when making service calls, making internal searches more efficient, and other improvements in the code base.

  In MÄK RTI 4.3 the number of messages during startup is significantly reduced in large federations. Previously federations could have late joiners, in which subscriptions from newly joined federates could result in the resending of discovery messages to all joined federates. A new messaging scheme for late joiners is controlled by the RID parameter **RTI_internalMsgSmartForwardingWhenUsingRtiexec**. When this is set to 1 (default) it greatly reduces the number of messages sent over the network, which in turn can also decrease CPU usage by the rtiForwarder. In addition to the new RID setting, we have made optimizations in messaging such that when the Data Distribution Management service is turned off in the RID file (**RTI_dataDistMgmt** 0), messages sent for discovery callbacks are smaller and bundled when possible, decreasing the amount of traffic on the network and reducing rtiForwarder CPU usage.

  In the RTI Assistant, we have made performance improvements in the communication between the LRC and the RTI Assistant such that larger entity counts are handled and displayed better when looking at the RTIspy Federate Statistics.

- Support for Microsoft Visual C++ 11.0.
- New RID parameter: **RTI_internalMsgSmartForwardingWhenUsingRtiexec**. When enabled, this parameter allows internal messages, such as subscribe and register, intended for individual federates, to be directed at those federates instead of being sent directionless. This dramatically decreases the number of internal messages required to register new objects or late joining federates. This parameter can be independent of the **RTI_smartForwardingLevel**, which handles update messages only. It is recommended that it remain on, although it can be turned off in order for the RTI to work as it had previously.
- The RTI now always byte aligns data returned in reflectAttributesValues and receiveInteraction, which allows casting to integral types from getValuePointer() in VariableLengthData (HLA1516-2000 and HLA1516-2010 only). 64 bit values can now be safely cast when using getValuePointer() instead of having to do a memcpy.

Documentation Updates

The documentation is up-to-date, but screen captures have not been updated to show the new RTI iconography for dialog box and window title bars and the RTI Assistant.

Bug Fixes

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

- Having a syntax error in *rid.mtl* could cause a MAKRti::DtTemplatedException<1> to be thrown when instantiating an RTIAmbassador. 51856
- HLA1516-2000 and HLA1516-2010 rtiVersion() and rtiName() functions now return the same values in the C++ and Java API. 49278
- The RTI's method of converting enumerations was incorrect and it caused a number of status messages to return false negative statuses. For example, the status message for "RestoreStatus" was always FEDERATE_WAITING_FOR_FEDERATION_TO_RESTORE even if it ran correctly. The method of generating enumerations has been changed and all status messages now show the correct value. 51568
- Fixed a bug in the rtiexec where a save federation execution service call could cause an unhandled exception. 50209
- Fixed a bug where the plugins.xml file for the RTISPY Plugin API could not be located on Linux platforms. 50267
- If the rtiAssistant was launched implicitly by a federate in Windows, the rtiAssistant inherited handles from the federate, causing unexpected behavior when the federate exited (such as federate TCP connections remaining active for the lifetime of the rtiAssistant rather than the federate). 50670
- Fixed a bug in the Java HLA1516-2010 HLAvariableArray encoding helper's resize() and toByteArray() methods. 51019
- The Java HLA1516-2010 HLAfixedArray encoding helper incorrectly threw an ArrayIndexOutOfBoundsException. 51011
- When using HLA_IMMEDIATE callback model in HLA1516-2010, Java federates would sometimes hang on exit due to an internal non-daemon thread running in the RTI. 50723
- Fixed a bug with the RTI_CONFIG environment variable where Java federates could not resolve FOM locations. 50923
- RTI_ridConsistencyChecking 1 could cause a crash with HLA_IMMEDIATE callback model or RTI_processingModel 3 (Asynchronous Callbacks). 50945
- In the Java HLA1516-2010 Encoder helpers, EncoderFactory.createHLAfixedArray(DataElementFactory<T> factory, int size); did not properly initialize with the size and the factory. 50956
- Fixed intermittent crash at startup when using processing model 2 - asynchronous process message. 50627
- The reportFederationExecutions callback will be called when in processing model 2 - asynchronous process messages. 50649
- When using 64 bit libraries and using RTI_smartForwardingLevel 1 or 2, updateAttributeValues could cause a crash due to byte alignment. 50606

- The RTI did not properly parse HLA1516-2010 FOMs and misidentified the HLA specification as HLA1516-2000. 49871

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.