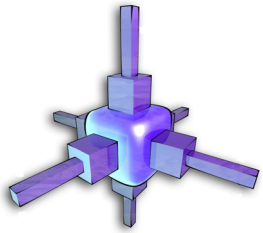

MÄK RTI 4.1.1 Release Notes

This document provides the following release-specific information for MÄK RTI 4.1.1:

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).)

Supported Systems

Table 1 lists the platforms supported by the standard version of MÄK RTI 4.1.1. 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 4.1.1 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 4.1.1 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/

Backwards Compatibility

The MÄK RTI 4.1.1 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 4.1.1 and some use a previous version of the MÄK RTI.) However, except as noted below, the MÄK RTI 4.1.1 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 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 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*.

Because the DMSO RTI NG v6 was last built with the MSVC++ 6.0 compiler, it is not link compatible with the MÄK RTI 4.1.1. However, updated versions of the DMSO RTI provided by other vendors using compilers supported by MÄK RTI 4.1.1 would be link compatible.

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

New Features and Updates

MÄK RTI 4.1.1 is primarily a maintenance release. It has the following new features:

- The efficiency of FED file distribution and FOM module merging has been improved in this release. In addition to optimizing the code to improve performance, the RTI now features more intelligent FED file distribution. In RTI 4.0, **RTI_distributeFedFile** became one of the parameters which was forced on when the RTI was configured for Full Compliance mode. This was because HLA Evolved FOM module merging required the rtiexec to have copies of each of the modules that was merged into the FOM of the federation execution. The downside to this approach was that while the federation was starting up, the rtiexec spent a large amount of time distributing files to and from the federates. This led to longer start up times than were present in previous versions of the RTI where FED file distribution was disabled by default.

  In order to alleviate any unnecessary FED file distribution, both the LRC and the rtiexec will attempt to load a local copy of a file before requesting distribution. If the file can be found in either the working directory or the directory specified by the RTI_CONFIG environment variable, distribution of that file will not be necessary. This can significantly reduce the startup time for your federation.

  However, one of the benefits of FED file distribution was that it ensured that all federates were using the same version of the file. Otherwise one federate might inadvertently load an old version of the file with the same name, which could result in miscommunications between federates. So in order preserve this capability, we introduced a new RID parameter called **RTI_fullFedFileDistribution**. It is disabled by default, but if set to 1 it will cause the RTI to fully distribute FED files as it did in previous releases. Another option is to enable **RTI_crcCheckFedFile**, which now works with FOM modules as well. With this option enabled, the RTI will simply distribute the file's CRC instead of its entire contents. This is usually enough to detect an error which can then be corrected before trying to launch your federation again.

- New RID parameters:
  - **RTI_fullFedFileDistribution**. Described in the previous paragraphs.
  - **RTI_throwExceptionCallNotAllowedFromWithinCallback**. If enabled, the HLA Evolved LRC will throw the CallNotAllowedFromWithinCallback exception when certain RTIambassador service calls are made within FederateAmbassador callbacks. This parameter is disabled by default, but is forced on when **RTI_forceFullCompliance** is enabled. This is a federate specific parameter.

Documentation Updates

The RTIspy API documentation has been added to the HTML class documentation (now API Documentation). It will be removed from *MÄK RTI Reference Manual* in the next release.

Bug Fixes

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

- Fixed and improved the text in some of the tool tips in the RTI Assistant Federations View. 47515
- Synchronization points were not always displayed in the RTI Assistant Federations View. 47568
- The HLA Bounce example federate now properly encodes and decodes attribute values. 47626
- Fixed issues with sending best-effort messages on loopback and VPN interfaces. 47636
- Fixed a crash that occurred when the specified FDD file could not be found. 47652
- The RTI_RID_FILE specification in the HLA Evolved localSettingsDesignator or HLA 1516 RTIambassador creation arguments would not support filenames or paths with spaces in them. 47706
- Federates would sometimes crash if another federate in the federation crashed or shut down unexpectedly. 47714
- Fixed a crash that occurred when HLA Evolved FDD files contained a blank semantics field. 47952
- The RTI was unable to load HLA Evolved FOM modules in which the dimension of an attribute or interaction included white space. 48007
- Fixed a crash that occurred when an HLA 1516 FDD file contained duplicate enumeratedDatatypes. 48019
- In HLA Evolved, the RTI was not throwing the NotConnected exception when service calls were made prior to successfully calling the connect service call. 48055
- Using the RTI within an application that uses Boost io_service would result in hangs or crashes. 48063
- RTI was not able to load FDD files which were located in the directory specified by the RTI_CONFIG environment variable. 48065
- The RTI now displays an appropriate error message when a dimension is incorrectly defined in an HLA 1516 or HLA Evolved FDD file. 48084
- The RTI now accepts HLA 1516 FDD files in which the upperBound is not specified for the "Federate" dimension. 48084
- The RTI would throw an exception if a single federate tried to create multiple federation executions. 48087
- The RTI will now correctly throw the HLA Evolved CallNotAllowedFromWithinCallback exception if **RTI_throwExceptionCallNotAllowedFromWithinCallback** is enabled in the RID file. 48188
- The RTI was sometimes throwing the HLA 1.3 ConcurrentAccessAttempted exception when it should not have been when **RTI_throwExceptionOnConcurrentAccess** was enabled. 48188

- The rtiSimple13 example federate had a logic error in the code that performed a federation save. 47876
- The RTI would sometimes crash if a federate tried to join a federation which had not already been created. 48208
- Examples did not build out of the box. 46637
- Fixed a logic error that caused the RTI to send interactions via reliable transport even if the FOM specified they should be sent best-effort. 48330
- The Java bindings did not work with FOM file names created using the File.toURI().toURL() method. 48355
- The RTI Assistant crashed when an invalid handle was specified to the RTI command line tool's "delete" or "kill" commands. 48362
- The RTI was not processing the CallbackModel parameter that was specified in the connect() service call of HLA Evolved federates. 48417

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 RTIspy browser-based GUI does not work well with Internet Explorer. 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. 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).

RTIspy Diagnostic GUI Scalability

The RTIspy 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.