



MÄK RTI 2.0.2-ngc Release Notes

This document provides the following release-specific information for MÄK RTI 2.0.2-ngc:

Overview	2
Supported Systems	2
Qt Release Compatibility	2
Backwards Compatibility	2
Network Compatibility	3
Compatibility with the DMSO RTI	3
New Features and Updates	3
RTIspy API Changes	4
Fixed Bugs	5
Known Problems	6

MÄK RTI 2.0-ngc has been verified by DMSO as fully compliant with the HLA Interface Specification, version 1.3.

Overview

MÄK RTI 2.0-ngc has been verified by DMSO as fully compliant with the HLA Interface Specification, version 1.3. All services are implemented.

Supported Systems

Table 1 lists the platforms supported by the MÄK RTI 2.0.2-ngc. 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: Platforms supported

Platform	Compiler
SGI IRIX6.5	MipsPro7.3 C++ compiler (available from SGI)
Sun Sparcstation with Solaris2.7 or later	SPARCompiler C++ version 5.2 (available from Sun)
PC with Red Hat Linux 8.0	GNU C++ (g++) distributed with Red Hat distribution. gcc version 3.2
PC with Windows NT 4.0, Windows 98/2000/XP	Microsoft Visual C++ 6.0.

MÄK RTI 2.0.2-ngc was built with VR-Link 3.8.1-ngc.

Qt Release Compatibility

The MÄK RTIspy GUI uses Trolltech's Qt cross-platform GUI toolkit. In particular, the GUI included with MÄK RTI 2.0.2 relies on Qt version 3.1.2. (The Qt run-time libraries are distributed with the MÄK RTI). If a federate also uses Qt, it must be sure to use the same version of Qt as the MÄK RTIspy GUI, in order for the RTIspy GUI to function properly for that federate. Federates built with other versions of Qt should disable the RTIspy GUI to avoid these incompatibility problems.

Backwards Compatibility

The MÄK RTI 2.0.2-ngc 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 2.0.2-ngc and some use a previous version of the MÄK RTI.) However, the MÄK RTI 2.0.2-ngc 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 the DMSO RTI. (In general, no two RTI implementations are network compatible, due to differences in the low-level network mechanisms that they use (which include, but are not limited to packet layout.)) 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 the DMSO RTI

With the exception of several additional functions (such as, `DtReadFileDescriptor()`, described in MÄK RTI documentation), the MÄK RTI 2.0.2-ngc has the exact same API as the DMSO RTI 1.3 NG (except v5). Unless you are using the `RtReadFileDescriptor` function, an application built against the DMSO RTI NG does not need to be recompiled in order to be used with the MÄK RTI 2.0.2-ngc, or vice versa (except that the Linux version is not compatible with the DMSO RTI).

MÄK RTI 2.0.2-ngc uses the same FED file format as the DMSO RTI, but a different RID file format. (RID file formats depend on the RTI-implementation.)



MÄK products, including the MÄK RTI 2.0.2-ngc, are not run-time compatible with the DMSO RTI NG v5 due to changes in the how the DMSO RTI handles namespaces.

New Features and Updates

MÄK RTI 2.0.2-ngc has the following new features:

- ♦ The RTI Executive has been optimized to no longer receive best effort messages (FOM data) when internal RTI messages are sent reliably (`RTI_internalMsgReliable` is enabled).
- ♦ The RTISpy API has been updated.

RTIsPy API Changes

- The use of the DDM, DtTime, and DtMom pre-compile macro definitions were changed in many places from “#if” to “#ifdef”.
- The collection of destination addresses in DtRIDParameters::myDestAddr was changed from an array of strings to a list of strings.
- The limits on FOM classes, attributes, and parameters have been modified. The limits on the number of class are now solely based on the FED file. The limits on the number of attributes and parameters are fixed at 300 to fit within the fixed bit array used to store handles. The attribute and parameter limits have been moved from the FOM Manager (*DtFomClassMgr*) to the object and interaction class information classes (*DtObjectClassInfo* and *DtInteractionClassInfo*).
- The error value for the join response message was added to the combined DtJoin-ResponseMsg::set(...) member function.
- Some methods were added to the *DtConnection* class to allow actions to be performed on the connection without having to get access to the internal socket. In addition, the VR-Link socket class was modified to generically support some of the methods that were only defined in specific socket derivations (for example, flush()).
- The *DtConnectionMgr* was changed to better encapsulate how it creates and uses connections, particularly best effort. Previously, the methods to create and use the connections connoted the default underlying implementation (for example, tcpConn() and udpConn()). These names and the member data were changed to the more generic reliableConn and bestEffortConn. Also, the creation of the best effort connection exposed specific details of how the default UDP connection was set up. The initialization was not clear and it was not consistent with how the reliable connection was created. A single createBestEffortConn was added to replace several helper methods used to create and initialize the best effort connection.

As a result of these changes, the preferred method for replacing the connections is by overriding the DtConnectionMgr::createReliableConnection() and DtConnectionMgr::createBestEffortConnection() as opposed to the DtConnectionMgr::init() (the create calls are made in the init() method). Even if the DtConnectionMgr::init() method is overridden (such that the code to initialize the connections is totally contained in init), the two create connection methods should also be overridden. These calls may be (and in fact are) called outside of DtConnectionMgr::init().

- ♦ The RTI Executive (rtiexec) and federate local RTI components (LRC) use the same connection component for network communication. This component supports both best effort and reliable communication. Previously, this component would always initialize both the communication connections. However, the rtiexec has no need to process best effort messages if internal RTI messages are sent reliably. In order to support an optimization for the rtiexec to not create a best effort connection if internal messages are sent reliably, the connection manager may delay creating the best effort connection until the connectFederation() method. In that case, the createBestEffortConnection() method would be called at that time if the bestEffortConn() is NULL. If the createBestEffortConnection() method was not overridden, it could replace any initialization of the best effort connection performed during init().

Fixed Bugs

This release fixes the following problems in previous releases:

- ♦ Several bugs which may have crashed federates have been removed from the RtiSpy GUI.
- ♦ The rtiexec GUI has been optimized so it should now be much more usable with large federations. Previously, critical messages were more likely to be dropped due to blocking in the rtiexec; this could lead to fatal errors in the federates.
- ♦ Problems with RTIspy licensing have been resolved. If a user started the GUI for the rtiexec, it consumed a plug-in license. Customers with only one runtime license were unable to run the rtiexec GUI and a federate.
- ♦ A problem dumping FED files on Solaris has been fixed.
- ♦ When setting `RTI_notifyLevel` to 3 or above in combination with enabling `RTI_asynchronousIO`, the LRC will now display sent and received messages in correct sequential order as they occur relative to the federate. Previously, the asynchronous thread generated the output for sent messages and the output could be displayed some time after it occurred in the federate.
- ♦ Previously, the processing of the Next Event Request (NER) services could result in a deadlock. The conditions for the deadlock were that TSO messages with time stamps less than any NER time were sent such that no federate was subscribed to the data. This deadlock condition has been removed and the RTI now advances time correctly.
- ♦ The order for finding RID and FED files is now consistent. For the FED file, the RTI searches in the execution directory and then in the directory found in the `RTI_CONFIG` environment variable. For the RID file, the RTI will use the file specified in the `RTI_RID_FILE` environment variable if it exists. If not then the RTI will search the execution directory followed by the `RTI_CONFIG` directory.
- ♦ While processing the create, destroy and join services, the RTI will now yield the processor while it waits for message exchanges with the rtiexec.
- ♦ LRC GUI window can be closed without crashing federate.

Known Problems

This release has the following known problems:

- ♦ 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.
- ♦ If you are using the Linux version of the MÄK RTIs with versions of MÄK products earlier than the most recent releases, the MÄK products may crash. If this happens, contact MÄK technical support and we will provide you with a fix for your application.