



MÄK RTI 2.4.2 Release Notes

This document provides the following release-specific information for MÄK RTI 2.4.2:

Supported Systems	2
Qt Release Compatibility	2
Patching the Microsoft Visual C++ 6.0 Compiler	3
Backwards Compatibility	3
Network Compatibility	4
Compatibility with the DMSO RTI	4
New Features and Updates	4
TCP Reconnect	5
RTI Exec Logging	5
MOM Service Reporting	5
Delayed Auto-Provides (1516 only)	5
Improved Support for HLA 1516	6
Customized RID Files	6
Documentation Updates	7
Bug Fixes	7
Known Problems	8
RTI 1516 Implementation Issues	8

Copyright © 2005 MÄK Technologies, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, RTIsby®, and VR-Link® are registered trademarks of
MÄK Technologies, Inc.
Document ID: RTI-2.4.2-2-051222

Supported Systems

Table 1 lists the platforms supported by the MÄK RTI 2.4.2. 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 Solaris 8 or later	SPARCompiler C++ version 5.2 (available from Sun)
PC with Red Hat Linux 9.0	gcc version 3.2.2
PC with Red Hat Enterprise 3	gcc version 3.2.2
PC with Red Hat Fedora Core 3	gcc version 3.4.4
PC with Red Hat Enterprise 4	gcc version 3.4.4
PC with Windows 2000/XP	Microsoft Visual C++ 6.0 and 7.1 (.NET)

MÄK RTI 2.4.2 was built with VR-Link 3.9.6.

Qt Release Compatibility

The MÄK RTIspy GUI uses Trolltech's Qt cross-platform GUI toolkit. (The Qt runtime 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.

The default RTIspy LRC Console GUI Plugin is built with Qt 3.3.5. For federates such as recent versions of the MÄK Logger, MÄK Stealth, and VR-Forces 3.8, which require Qt 3.3.1, an alternate build of the RTIspy GUI Plugin is included with MÄK RTI 2.4.2. The Qt 3.3.1 versions of the plugin for Windows are:

- ♦ LRCconsole-qt3.3.1.dll
- ♦ LRCconsoleqt-qt3.3.1.dll
- ♦ LRCconsole1516-qt3.3.1.dll
- ♦ LRCconsole1516d-qt3.3.1.dll.

The Qt 3.3.1 versions of the plugin for UNIX are:

- ♦ LRCconsole-qt3.3.1.so
- ♦ LRCconsole1516-qt3.3.1.so.

To select the Qt 3.3.1 version of the RTIspy LRC Console GUI instead of the Qt 3.3.5 version, use the `RTI-addPluginName` RID file parameter. For example, to select the Debug build of the Qt 3.3.1 plugin for HLA 1.3 under, Windows set the following RID parameters:

```
(setqb RTI_enableLrcGUI 0)
(RTI-addPluginName "LRCconsole-qt3.3.1.dll")
```

Patching the Microsoft Visual C++ 6.0 Compiler

If you use Microsoft Visual C++ version 6.0, you must install a patch to a standard header file, in order to successfully build applications for HLA 1.3 or 1516. The patch comes from Dinkumware, Ltd (the company that developed the Standard C++ Library header files for Microsoft), but for convenience, we are distributing it with MÄK RTI releases.

The patch fixes a bug that would cause applications to crash if they pass `std::sets` or `std::maps` from an application to a DLL or vice-versa. For details about the patch go to http://www.dinkumware.com/vc_fixes.html.

To install the patch:

1. In the top level MÄK RTI directory, is a file called *XTREE-msvc++patch*. Rename this file *XTREE*.
2. Make a back-up version of the original *XTREE*.
3. In the Visual Studio *include* directory (for example, *C:\Program Files\Microsoft Visual Studio\VC98\Include*), replace the original version of *XTREE* with the patched version.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 6.0 and 7.1. When you run MÄK products together, 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.

Backwards Compatibility

The MÄK RTI 2.4.2 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 2.4.2 and some use a previous version of the MÄK RTI.) However, the MÄK RTI 2.4.2 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 (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 2.4.2 for HLA 1.3 has the exact same API as the DMSO RTI 1.3 NG. 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 2.4.2, or vice versa (except that the Linux version is not compatible with the DMSO RTI).

MÄK RTI 2.4.2 uses the same FED file format as the DMSO RTI, but a different RID file format. (RID file formats depend on the RTI-implementation.)

New Features and Updates

MAK RTI 2.4.2 has the following new features and updates:

- ♦ TCP Reconnect
- ♦ RTI Exec Logging
- ♦ MOM Service Reporting
- ♦ Delayed Auto-Provides (1516 only)
- ♦ Improved support for HLA 1516
- ♦ Examples of customized RID files.

TCP Reconnect

You can now request that the LRC attempt to re-establish its connection to the TCP Forwarder if the connection breaks. This feature may allow the federate to continue running in the face of a large class of network layer errors. To use this feature, set the `RTI_federateReconnectEnabled` parameter to 1, and specify a value for `RTI_federateReconnectPause`. If TCP reconnect is enabled, when a broken connection is detected, the federate waits for the time specified by `RTI_federateReconnectPause` to allow the network to stabilize and then attempts to re-establish the broken connection. If the federate reconnects, it usually continues to operate in the federation with no visible effects to the application. If the federate cannot reconnect, the LRC throws an `RTIinternalError` exception to indicate that the federate has lost its connection to the federation (as it does when `RTI_federateReconnectEnabled` is disabled). During the reconnect period some reliable messages may be lost, since the federate is temporarily disconnected from the RTI.

RTI Exec Logging

You can now log `rtiexec` output to file. The `RTI_rtiExecLogFileName` parameter specifies the name of the log file for the `rtiexec` process. A unique identifier is inserted in the file name when `RTI_reuseLogFile` is enabled. This parameter is overridden by the `rtiexec -l` option.

MOM Service Reporting

You can now instruct the RTI to enable MOM service reports even if the federate does not enable them via the RTI Ambassador service call. This feature may be useful in the management of federations with federates which do not automatically enable service reports. To enable MOM service reporting, set the `RTI_momServiceReporting` parameter to 1.

Delayed Auto-Provides (1516 only)

HLA 1516 federates can now instruct the RTI to buffer automatic update requests temporarily, in order to accumulate automatic requests from multiple federates. This can reduce the number of automatic `provideAttributeUpdate` callbacks required for changes in publication and subscription. To disable this feature, set the `RTI_autoProvideDelay` parameter to -1. To enable it, specify the amount of time to wait before reading update requests, in seconds.

Improved Support for HLA 1516

This release includes improvements to the following services:

- ♦ DDM
- ♦ Time Management
- ♦ Ownership Management
- ♦ MOM.

Customized RID Files

The MÄK RTI includes three sample RID files that are customized for different objectives. You can use these files as alternatives to the default RID file. The sample RID files are subsets of the default RID file and show you how to customize the default settings for the following cases:

- ♦ High performance (*highPerf-rid.mtl*). This file shows the parameters you need to customize to maximize performance and provides suggested values.
- ♦ Lightweight mode (*lightWeight-rid.mtl*). This file includes the parameters supported by lightweight mode. It excludes parameters for the services that lightweight mode does not support.
- ♦ Typical configurations (*basic-rid.mtl*). This file includes the parameters that most installations will want to or need to customize regardless of performance objectives.



If you use one of the sample RID files instead of the default RID file, you must either specify the file with the RTI_RID_FILE environment variable or rename it *rid.mtl*.

Documentation Updates

The description of the search order for RID files has been updated. The correct text is:

The search order for RID files is:

If `RTI_RID_FILE` is defined:

1. Try to open the file specified in the `RTI_RID_FILE` environment variable.
2. Try to open the file specified by `RTI_CONFIG/RTI_RID_FILE`.
3. If *rid.mtl* cannot be found in these locations, the MÄK RTI stops looking for the RID file and fails.

If `RTI_RID_FILE` is not defined:

1. Try to open *rid.mtl* file in the directory from which you run the application (which is not necessarily the directory in which the executable is located.)
2. Try to open the file specified by `RTI_CONFIG/rid.mtl`.
3. If no RID file is identified by one of these methods, the MÄK RTI prints a warning and opens using default values.

Bug Fixes

The following bugs have been fixed in this release:

- ♦ HLA 1516 performance has been improved significantly for all 1516 federates.
- ♦ Both the MSVC++ 6.0 and MSVC++ 7.1 builds of the RTI are now built with the Intel C++ compiler, which has been shown to provide a significant performance improvement in RTI processing.
- ♦ The `rtiexec` will now honor the `RTI_notifyLevel` parameter specified in the RID file as well as the `-n` startup option.
- ♦ Several issues that may cause crashes for some federates using DDM in have been fixed.
- ♦ A memory leak in RTI 2.4, which applied to sending interactions with Smart Forwarding enabled, has been fixed.
- ♦ An invalid memory access, which could result in federates reporting a crash on exit, has been fixed.

Known Problems

This release has the following known problems:

- 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.
- If you are using the Linux version of the MÄK RTIspy 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.
- The rtiexec does not connect the output streams to a console until after reading the RID file in Windows. As a result, errors opening or parsing the RID file are not displayed. If `RTI_enablePopUpErrorMsgs` is enabled then these errors are displayed in a popup box, however this solution works only under Windows.
- The asynchronous callback feature is not yet available for HLA 1.3 federates. It is only available for 1516 federates. Enabling this option will have no effect on an HLA 1.3 federate.

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.