



MÄK Data Logger 5.1 Release Notes

This document provides the following release-specific information 5.1:

Systems Supported.....	2
Logger API	2
Qt Toolkit Compatibility	2
Compiler Compatibility on Windows	2
License Manager.....	3
ODBC Driver Support.....	3
Using Libraries and Binaries Built with Visual Studio 2005 and Later	3
Patch Required for AMD Dual-processor Windows PCs	4
Network Compatibility.....	4
RTI Support	5
Support for HLA Evolved.....	5
FOM Support.....	6
New Features	7
Documentation Updates	7
Bug Fixes	8
Known Problems	8
Playing HLA 1.3 Logger Files in HLA 1516 Federations	9

Copyright © 2011 VT MÄK, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, RTIspy®, B-HAVE®, and VR-Link® are registered trademarks of VT
MÄK. Document ID: LOG-5.1-2-111005

Systems Supported

Table 1 lists the platforms currently supported by MÄK Logger 5.1. Application code must be built with the indicated compilers in order to link to Logger libraries.

Table 1: Platforms supported

Platform	Compiler
Red Hat Enterprise Linux Workstation 5.0	Default compiler
PC with Windows XP/Vista/Windows 7	Microsoft Visual C++ 7.1
	Microsoft Visual C++ 8.0
	Microsoft Visual C++ 9.0 (32-bit)
	Microsoft Visual C++ 10.0 (64 bit)



You must be an administrator to install MÄK products on Windows Vista.

Logger API

You must have VR-Link 4.0.3 and its required compiler versions to link the Logger libraries, but you do not need it to run the executables. Please see VR-Link documentation for information about required compilers.

You must have a VR-Link developer's license to compile plug-in or applications using the Logger API.

Qt Toolkit Compatibility

The Logger GUI is built using the Qt Toolkit, a cross-platform GUI toolkit. Logger 5.1 uses Qt 4.6.3. You do not need Qt to do development that does not affect the GUI.

Compiler Compatibility on Windows

Current and prior releases of MÄK products have been compiled using different versions of the Visual C++ compiler. When you run MÄK products together, for example, the Logger and the Stealth, we strongly recommend that you run versions created with the same compiler. Mixing products compiled with different compilers can result in program instability.

License Manager

To run the licensed version of Logger, you must install license management software. Logger 5.1 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 Data Logger 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 Logger 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>

License management support is available at <http://www.mak.com/support/get-licenses/license-support.html>.

ODBC Driver Support

The Logger database export feature supports the following ODBC drivers:

- MySQL ODBC 3.51 and 5.1.
- Microsoft Access.

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, with the release of the new C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later, the libraries can no longer 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 is distributing 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. MÄK has chosen to not integrate the MÄK installer and the Microsoft installer so as not to require users to have Administrator privileges to install MÄK products. Therefore, if you who do not have a compiler installed, or get error messages like “Software has not been installed correctly, please re-install”, you must apply the patch.

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.

Patch Required for AMD Dual-processor Windows PCs

VR-Link-based products use a high resolution counter for time calculations on Windows PCs. Customers who are running Windows on PCs with multiple AMD Athlon 64-bit processors may notice clock jitter, which may cause time in MÄK products to run backwards. This occurs when the Windows scheduler changes the CPU the MÄK process is using. If the high resolution counters on each processor are not synchronized, the application may witness a decrease in the high resolution counter value stored in the processor causing an incorrect time calculation. To fix this problem customers, apply the AMD Dual-Core Optimizer patch provided by AMD. You can get the patch at:

http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_871_9706,00.html



If you get an error when you try to access this URL, reload the page.

Network Compatibility

HLA only

Logger 5.1 was built against VR-Link 4.0.3 and is compliant with:

- ♦ RPR-FOM 1.0 and a subset of 2.0 (draft 6, 14, and 17)
- ♦ MÄK RTI 4.x (please see “[Known Problems](#),” on page 1-8.)

DIS only

Logger 5.1 supports DIS 4, 5, and 6.

RTI Support

We recommend using the latest version of the MÄK RTI. However, the Logger should work with other RTIs that conform to the HLA 1.3 specification, the IEEE 1516 SISO DLC API, or HLA Evolved and are built using the same compiler as your version of the Logger.

When using the MÄK RTI, remember to make sure that the Logger can find your FED file, and optional *rid.mtl*, file by either putting them in the directory from which you are running, or by setting the environment variable the directory that contains them.

Support for HLA Evolved

Logger 5.1 updates the HLA Evolved version to be compatible with MÄK RTI 4.0.4, VR-Link 4.0.3, and the HLA Evolved specification. A full discussion of this issue is as follows:

The HLA Evolved libraries included with the MÄK RTI 4.0 through 4.0.3 were unintentionally built against one of the last draft versions of the HLA 1516-2010 specification header files. Between the version of the header files used by the MÄK RTI and the final version of the specification there was one change. The HLA Evolved API has three functions for `createFederationExecution`. In the version originally used by the MÄK RTI there was an ambiguity. Due to optional parameters, two of these functions could be invoked in such a way that it was not clear which one was being used. Here are the functions as they used to be:

```
// variation 1
virtual void createFederationExecution (
std::wstring const & federationExecutionName,
std::wstring const & fomModule,
std::wstring const & logicalTimeImplementationName = L"");

// variation 2
virtual void createFederationExecution (
std::wstring const & federationExecutionName,
std::vector<std::wstring> const & fomModules,
std::wstring const & logicalTimeImplementationName = L"");

// variation 3
virtual void createFederationExecution (
std::wstring const & federationExecutionName,
std::vector<std::wstring> const & fomModules,
std::wstring const & mimModule,
std::wstring const & logicalTimeImplementationName = L"");
```

Variation one is unambiguous. However, consider the following code:

```
createFederationExecution( L"FedExName", vectorOfFomModules, L" );
```

It is not clear whether you are using variation two or three, and the compiler would throw an error.

To fix this ambiguity, the final version of the HLA Evolved API renamed the third version of the function to `createFederationExecutionWithMIM`. However this revision did not make it into the header files used by the MÄK RTI. RTI 4.0.4 corrects this and uses the final version of the HLA Evolved header files. As a result, we recommend that anyone using the MÄK RTI with HLA Evolved upgrade to this version. Because the API has changed between RTI 4.0.3 and 4.0.4, any HLA Evolved federates built against previous MÄK RTI versions will need to be recompiled against the new version.

Because VR-Link's HLA libraries are built against the MÄK RTI, this issue also exists in previous versions of VR-Link that supported HLA Evolved. We recommend that anyone using VR-Link with HLA Evolved upgrade to version 4.0.3, which is built against the latest RTI and uses the renamed function.

This problem affects MÄK products built against the MÄK RTI and VR-Link that support HLA Evolved, such as Logger 5.0. Logger 5.1 has been built against VR-Link 4.0.3 and is now compliant with the final version of the standard.

As a general rule, any version of the Logger is compatible with any version of the RTI that supports the same platform. However, due to this header file change, previous versions of the Logger and the MÄK RTI will not be compatible when using HLA Evolved. This change does not affect compatibility between the Logger and the MÄK RTI when using HLA 1.3 or 1516-2000.

When using HLA Evolved:

- Logger 5.0 is only compatible with MÄK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.
- Logger 5.1 and later releases are only compatible with MÄK RTI 4.0.4 and later releases.

FOM Support

Logger 5.1 has built-in support for versions 0.5, 0.7, 0.8, 1.0, and 2.0, drafts 6, 14, and 17, of the RPR FOM. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, Logger 5.1 uses RPR FOM 1.0.

For information about FOM mapping and selecting the correct FOM Configuring the Logger for HLA in Chapter 2, "Installing and Configuring the Logger", in *MÄK Data Logger Users Guide*.

New Features

Logger 5.1 has the following new features:

- ♦ A new remote control API. The old remote control API is still supported, but has been deprecated. In addition to supporting all of the Logger commands, as you would expect, the new API allows you to control multiple Loggers from one application. Use of multiple Loggers can be beneficial for scenarios with high entity counts that require load balancing and redundancy. For details about the new API, please see Chapter 13, *Remote Control API*, in *MÄK Data Logger Users Guide*.
- ♦ High speed playback. The high speed playback feature skips from checkpoint to checkpoint, resulting in dramatically faster playback. For details, please see Section 6.6.7, “*Using High Speed Playback*”, in *MÄK Data Logger Users Guide*.
- ♦ Support for MS VC++ 10 (64 bit).
- ♦ Support for MySQL ODBC Connector 5.1.
- ♦ The lgrControl example has been updated to use the new remote control API.
- ♦ The loggerApp example has been added. It can be used as a template to modify and recompile the Logger application.
- ♦ Text dump of Logger file includes wallclock time for when the recording was made and when each packet is received. The text dump also now includes the meta data packets (for example, Logger configuration data, FOM file, index table, and so on).
- ♦ The DIS Connections dialog box has been updated to allow setting a subnet mask.

Documentation Updates

MÄK Data Logger Users Guide has been updated to describe the new remote control API.

Bug Fixes

This release fixes the following problems:

- ♦ The Logger sent total state and other remote control PDUs in DIS without including the exercise ID. 45285
- ♦ When recording to multiple files, the last file segment was empty. 45084
- ♦ The logger incorrectly used the current time as the timestamp of received messages (updates and interactions). 45001
- ♦ The Logger did not load files if the file name had a dash. 44199
- ♦ When using DDM, a jump in time did not incorporate DDM region changes. 45203
- ♦ Not all regions were output when writing the subscription list to a file due to incorrect relational operators in region descriptors. 44085
- ♦ MySQL 5.1 was not supported. 37694
- ♦ The Logger crashed when performing a time advance due to an incorrect calculation between old and new time. 44046
- ♦ The Annotation view did not open when an annotation was created or when a file was loaded that had an annotation file. 42346

Known Problems

This section lists known problems with Logger functionality:

- ♦ Logger remote control messages cause previous versions of the Logger to crash. Therefore, you cannot use them in an exercise that is running mixed versions of the Logger. To disable remote control messages, set the following parameters in *lgrConfig.xml*:

```
<var name="sendResponses" type="bool" value="false"/>
<var name="connectSendInterval" type="double" value="-1.0"/>
<var name="heartbeatSendInterval" type="double" value="-1.0"/>
```
- ♦ The export to database feature is not fully supported on the VC10 platform. The VC10 platform requires 64-bit ODBC drivers. The 64-bit ODBC MySQL driver has been tested when added to the User DSN tab (versus the system DSN tab). However, the 64-bit ODBC MS Access driver has not been tested. It requires installation of 64-bit MS Office suite. 45979
- ♦ The HLA 1516 Logger crashes on exit if you use the MÄK RTI 2.4.2 through 3.1.1. This is due to a problem in the RTI.
- ♦ Logger files that include the VR-Forces embarkation feature in DIS exercises, such as *embarkdemoDIS.lgr*, do not reset embarkation status after time jumps. To correct the embarkation state of entities, stop playback and start playing the file from the beginning.

- If you are running the Logger in HLA 1516 with the MÄK RTI, and the RTI parameter `RTI_enableLrcWebService` is enabled, the Logger might pause when starting or stopping recording. This pause can be 30 seconds or longer and the Logger might not be responsive during this time.
- On Linux, online help may not work if the Logger lib directory is not in the `LD-LIBRARY_PATH` or the RTI *lib* directory is before it.

Playing HLA 1.3 Logger Files in HLA 1516 Federations

In general, the Logger uses the same file format for HLA 1.3 and HLA 1516 Logger files. You can record a Logger file from an HLA 1.3 federation, and play it back into an HLA 1516 federation (and vice versa). However, you must use the same FOM representation during record and playback. For example, if you use an HLA-1.3-style FED file during recording, you must use an HLA-1.3 style FED file during playback, even if you are using the IEEE 1516 version of the Logger and an IEEE 1516 RTI. (VR-Link and the MÄK RTI allow you to use HLA-1.3-style FED files with IEEE-1516-based federates, and vice versa).

The main reason that consistency in FOM format is necessary is that IEEE 1516 uses different names for the "Root" classes of the Object class hierarchies. A 1.3-style FED file requires a Root class a 1516-style XML files requires a Root class called *HLAObjectRoot*. The reason this can be a problem is that if the Logger is playing an HLA-1.3-based Logger file into a federation that is using a 1516-based XML file it might come across an instance of a class called, for example, "ObjectRoot.Vehicle". If it tries to register an object of this class, the RTI will complain that no such class exists. There might be a class called *HLAObjectRoot.Vehicle* in the current FOM, but the RTI or Logger does not know that this is actually the same class.

