



MÄK Data Logger 5.3.2 Release Notes

This document provides the following release-specific information:

Systems Supported.....	2
Logger API	2
Qt Toolkit Compatibility	2
Compiler Compatibility on Windows	2
License Manager.....	3
ODBC Driver Support.....	3
Network Compatibility.....	3
RTI Support	4
Support for HLA Evolved	4
FOM Support.....	5
New Features	6
Bug Fixes	6
Known Problems	6
Playing HLA 1.3 Logger Files in HLA 1516 Federations.....	6

Copyright © 2016 VT MÄK, 150 Cambridge Park Drive, 3rd Floor, Cambridge, MA 02140 All rights reserved. VR-Exchange™, VR-TheWorld™, VR-Vantage™, DI-Guy™, and DI-Guy Scenario™ are trademarks of VT MÄK. MÄK Technologies®, VR-Forces®, RTIspey®, B-HAVE®, and VR-Link® are registered trademarks of VT MÄK.

Document ID: LOG-5.3.2-2-160613

Systems Supported

[Table 1](#) lists the platforms currently supported by MÄK Logger 5.3.2. 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 6.0	Default compiler (64-bit)
PC with Windows XP, Vista, Windows 7, Windows 8	Microsoft Visual C++ 10.0 (32-bit and 64 bit)

Logger API

You must have VR-Link 5.1.3 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 5.3.2 uses Qt 4.7.4 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 5.3.2 uses FLEXlm 11.13.0.2. 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> (for 32-bit systems)
<ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe>
- ♦ Linux: <ftp://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz>

A license manager FAQ is available at:

<http://www.mak.com/support/faq.php#license>

ODBC Driver Support

The Logger database export feature supports the following ODBC drivers:

- ♦ MySQL ODBC 2.50 Driver
- ♦ MySQL ODBC 5.351 Driver
- ♦ MySQL ODBC 5.1 Driver
- ♦ MySQL ODBC 5.2w Driver.
- ♦ Microsoft Access.

For the VC10 64-bit Logger, a 64 bit connector is required.

The MySQL 5.1 server and connector can be downloaded at:

- ♦ server: <http://dev.mysql.com/downloads/mirror.php?id=411970>
- ♦ connector: <http://dev.mysql.com/downloads/mirror.php?id=411736>

Network Compatibility

HLA only

Logger 5.3.2 was built against VR-Link 5.1.3 and is compliant with:

- ♦ RPR-FOM 1.0, RPR FOM 2.0 drafts 6, 14, 17, and 20, and RPR FOM 2.0 (final version).
- ♦ MÄK RTI 4.1 or later.

DIS only

Logger 5.3.2 supports DIS 4, 5, 6, and 7.

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 5.3.2 updates the HLA Evolved version to be compatible with MÄK RTI 4.0.4, VR-Link 5.1.3, 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 or later, 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 5.2 has been built against VR-Link 5.1.3 and is 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 may 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 5.3.2 has built-in support for RPR FOM versions 0.5, 0.7, 0.8, 1.0, 2.0 drafts 6, 14, 17, and 20, and RPR FOM 2.0 final version. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, Logger 5.3.2 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 5.3.2 is primarily a maintenance release. It has the following new features:

- ♦ Support for VR-Link 5.1.3.
- ♦ Support for FLEXlm 11.13.0.2.

Bug Fixes

This release fixes the following problems:

- ♦ Pausing the Logger while recording occasionally caused a crash. 53957
- ♦ The HLA versions of the Logger did not always return a license when shut down. 52422

Known Problems

This section lists known problems with Logger functionality:

- ♦ 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.
- ♦ 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.

