



MÄK Data Logger 4.1 Release Notes

This document provides the following release-specific information 4.1:

Systems Supported	3
Logger API	3
Qt Toolkit Compatibility	3
Compiler Compatibility on Windows	3
Network Compatibility	4
RTI Support	4
FOM Support	4
New Features	5
New Recording Behaviors	5
Editing Logger Recordings	5
New Conceptual Approach to Logger Files	6
Packet Count Histogram	6
Support for Subscription Lists	6
Support for Time Management	6
New Commands	6
New Annotation Features	7
Exporting Files to Other Formats	7
Filtering VR-Forces Messages	7
Converting a File to the Current Logger Format	7
Documentation	7
Fixed Bugs	8
Known Problems	8

Copyright © 2007 MÄK Technologies, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, RTIspey®, and VR-Link® are registered trademarks of MÄK Technologies, Inc.

Document ID: LOG-4.1-2-070606

Playing HLA 1.3 Logger Files in HLA 1516 Federations.....	8
---	---

Systems Supported

Table 1 lists the platforms currently supported by MÄK Logger 4.1. 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 3.0	GNU C++ (GCC) 3.2
Red Hat Fedora Core 3	GNU C++ (GCC) 3.4.4
Red Hat Enterprise Linux Workstation 4.0	GNU C++ (GCC) 3.4.4
PC with Windows 2000/XP	Microsoft Visual C++ 7.1 and 8.0
SGI IRIX 6.5	MipsPro7.3 C++ compiler (available from SGI)
Sun Sparcstation with Solaris 2.8 or later	SPARCCompiler C++ version 5.2 (available from Sun)

Logger API

You must have VR-Link 3.11 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 4.1 uses Qt 3.3.5. 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.

Network Compatibility

HLA only

Logger 4.1 was built against VR-Link 3.11 and is compliant with:

- ♦ RPR-FOM 1.0 and a subset of 2.0 (draft 6, 14, and 17)
- ♦ MÄK RTI 3.1.1 (please see “[Known Problems](#),” on page 8.)

DIS only

Logger 4.1 supports DIS 2.0.4, IEEE 1278.1, 2.1.4, and IEEE 1278.1a, and can therefore interoperate with DIS applications of any of these

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 or the IEEE 1516 SISO DLC API and are built using the same compiler as your version of the Logger.



As of this release, we no longer provide an MSVC ++ 6.0 version of the Logger. Therefore, it is no longer compatible with the version of the DMSO RTI-NG that was built against MSVC++ 6.0.

To use Logger 4.1 with the RTIspy® LRC GUI, you must use a version of the MÄK RTI built with Qt 3.3.5.

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.

FOM Support

Logger 4.1 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 4.1 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 User’s Guide*.

New Features

This release has the following new and updated features:

- ♦ Redesigned default recording behavior and the ability to pause while recording.
- ♦ Ability to edit Logger recordings, including offsetting packet timestamps, merging recordings, deleting packets, and trimming dead space at the start of a recording.
- ♦ New conceptual approach to Logger files.
- ♦ Display of packet count in a histogram.
- ♦ Ability to specify HLA subscription lists.
- ♦ Support for HLA Time Management.
- ♦ New commands and dialog boxes to support added features.
- ♦ New annotation capabilities.
- ♦ Ability to export Logger files or segments of files to the Logger format or other formats.
- ♦ Ability to filter out VR-Forces messages.
- ♦ Ability to convert Logger files to the current format.

New Recording Behaviors

When you record a simulation, you can begin recording immediately, without specifying a name for the recording, by clicking the Record button. You can also specify a name for the recording prior to the start of recording, by choosing Record → Record As. Previously, if you chose Records As, the Logger went into Pause mode until you explicitly started the recording process. Now, the Logger begins recording immediately unless you choose Options → Pause on Record As. For details, please see Section 5.3, “[Recording Exercises](#)”, in *MÄK Data Logger User’s Guide*.

In past releases of the Logger you could not pause the recording process and then resume recording to the same file. Logger 4.1 supports a pause mode while recording.

Editing Logger Recordings

Logger 4.1 has added extensive editing capabilities through the GUI. You can concatenate files, delete sections of files, offset packets, and trim dead space. For details, please see Chapter 7, [Editing Logger Files](#), in *MÄK Data Logger User’s Guide*.

New Conceptual Approach to Logger Files

In previous releases of the Logger, file interaction through the GUI was static. A file was opened and closed. Now that editing capabilities have been added, the Logger treats files similarly to other programs that can edit files. Once you make a change to a file, it is in an unsaved state. You can record to, play, and edit unsaved files. The Logger prompts you to save unsaved files before you execute a command that would cause loss of data. This change is reflected in *MÄK Data Logger User's Guide* by now referring to Logger files rather than Logger recordings.

Packet Count Histogram

Logger 4.1 can display a histogram showing the packet count for objects and interactions. This display also shows the location of checkpoints along the timeline. For details, please see Section 5.11, “[Viewing a File's Packet Count Histogram](#)” in *MÄK Data Logger User's Guide*.

Support for Subscription Lists

By default, the Logger subscribes to all HLA classes. You can specify which classes to subscribe to by creating a subscription list. You can do this in the GUI or by using Logger commands. For details, please see Section 4.5.9, “[Specifying the Logger Subscription List](#)”, in *MÄK Data Logger User's Guide*.

Support for Time Management

The Logger now supports HLA Time Management. It supports time-constrained recording and time-regulating playback. For details, please see Section 4.5.6, “[Configuring Time Management](#)”, in *MÄK Data Logger User's Guide*.

New Commands

The Logger has new commands to support the new features in this release. For details about commands and how to use them on the command-line, the text interface, and in the configuration file, please see Chapter 8, [Using Logger Commands](#), in *MÄK Data Logger User's Guide*.

The text interface now supports string lists.

The Logger has new dialog boxes for setting the start and end times for playback segments and for setting the zoom.

New Annotation Features

In Logger 4.0, you had to set the time manually for annotations. Now you can add annotations that automatically record the time when they are created. You can also dynamically update the time of an annotation to the current time as a recording progresses. For details, please see Section 6.1.1, “[Adding an Annotation](#)”, in *MÄK Data Logger User’s Guide*.

You can now convert an annotation from a point annotation to a range annotation and vice versa.

Exporting Files to Other Formats

You can now export Logger files to other supported formats. Logger 4.1 supports export to plain text and to the Logger format. You can export a segment of a file to be its own independent file.

You can use the Logger API to write plug-ins for conversion to other formats. For details, please see Section 5.7, “[Exporting Data](#)”, in *MÄK Data Logger User’s Guide*.

Filtering VR-Forces Messages

If you are recording simulations created with MÄK’s computer generated forces product VR-Forces, you may want to filter out the messages that VR-Forces users for internal communication. Logger 4.1 supports filtering of these messages. For details, please see Section 5.10.3, “[Filtering VR-Forces Messages](#)”, in *MÄK Data Logger User’s Guide*.

Converting a File to the Current Logger Format

If you load a Logger file created with an older version of the Logger, you are prompted to convert it to the current format. You can decline to do so and still load the file. Thereafter, you can still convert the file from the Logger GUI.

Documentation

MÄK Data Logger User’s Guide has been reorganized, updated, and reprinted for this release.

Fixed Bugs

Logger 4.1 fixes the following problems from previous releases:

- DIS asynchronous I/O is fixed.
- The text interface now parses commands with list type parameters.
- A disconnect and reconnect bug in HLA 1516 has been fixed.
- The last message in a tape is now played.
- Multicast addresses were missing in the Connection dialog box.
- The Logger no longer crashes when sending transmitter PDUs in DIS.
- Sending MOM data in 1516 caused crashes. This data is now filtered out by default.
- Specifying a filename on the command line for DIS now works correctly.
- There was a problem when a user changed the playback speed while playback was paused.
- The DIS entity type filtering dialog box did not work properly.
- The Logger no longer freezes when loading some files longer than one hour.
- The Logger no longer exits improperly in Linux.

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.
- In the IRIX version of Logger 4.1, unloading plug-ins causes the Logger to crash. Therefore, the ability to unload plug-ins has been disabled.

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.

