



MÄK Data Logger 3.11.3 Release Notes

This document provides the following release-specific information for MÄK Data Logger 3.11.3:

Systems Supported.....	2
Logger API and Logger File API	2
Qt Toolkit Compatibility	2
RAM and Memory Requirements	2
Compiler Compatibility on Windows	2
Network Compatibility.....	3
RTI Support	3
FOM Support.....	3
New Features and Changes	4
Grouping (DIS Only).....	4
Resuming Playback After Stopping.....	4
New Configuration Parameters.....	5
Fixed Bugs	5
Known Problems	5

Copyright © 2005 MÄK Technologies, 10 Fawcett St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, RTIsPY®, and VR-Link® are registered trademarks of
MÄK Technologies, Inc.
Document ID: LOG-3.11.3-2-050823

Systems Supported

Logger 3.11.3 is being released for the following platforms:

- ♦ PC with Windows 2000/XP, Visual C++ 7.1
- ♦ Red Hat Linux 9.0, Workstation Enterprise 3, Fedora Core 3
- ♦ SGI IRIX 6.5 n32
- ♦ Sun Solaris version 2.8.

Logger API and Logger File API

You must have VR-Link 3.9.3 and its required compiler versions to use the Logger file library or the Logger API, but you do not need it to run the executables. See VR-Link documentation for information about required compiler versions.

You must have a VR-Link developer's license to compile plug-ins or applications that use the Logger API or Logger file API.

Qt Toolkit Compatibility

The Logger GUI is built using the Qt Toolkit, a cross-platform GUI toolkit. If you use the Logger API to change the GUI, you need to get a Qt license and install Qt. Logger 3.11.3 was built with Qt 3.3.1.

RAM and Memory Requirements

To install and run the Logger on a PC, we recommend the following minimum configuration:

- ♦ Pentium class PC 400 MHz or above
- ♦ 150 MB disk space
- ♦ 64 MB RAM.

On UNIX systems, the Logger requires the following amount of disk space:

- ♦ Linux - 150 MB
- ♦ Sun Solaris - 185 MB
- ♦ SGI IRIX - 325 MB.

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 Logger 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.

Network Compatibility

HLA only

Logger 3.11.3 was built against VR-Link 3.9.3 and is compliant with:

- ♦ RPR-FOM 1.0 and a subset of 2.0 (draft 6, 14, and 17)
- ♦ MÄK RTI 2.4.

DIS only

Logger 3.11.3 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 versions.

RTI Support

Logger 3.11.3 works with the MÄK RTI 2.x, and DMSO RTI NG v6.

Logger 3.11.3 works with older versions of the MÄK RTI, but we recommend using it with release 2.4.

When using the MÄK Real-time 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 RTI_CONFIG to the directory that contains them.

FOM Support

Logger 3.11.3 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 3.11.3 uses RPR FOM 1.0.

The keywords for specifying different versions of the RPR FOM with the `-f` startup option are:

- ♦ RPR FOM 0.5: `v1RPR05`
- ♦ RPR FOM 0.7: `v1RPR07`
- ♦ RPR FOM 0.8: `v1RPR08`
- ♦ RPR FOM 1.0: `v1RPR10`
- ♦ RPR FOM 2.0 draft 6: `v1RPR20006`
- ♦ RPR FOM 2.0 draft 14: `v1RPR20014`
- ♦ RPR FOM 2.0 draft 17: `v1RPR20017`.

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

New Features and Changes

Logger 3.11.3 has the following new features:

- ♦ Environmental processes send a heartbeat when a recording is paused
- ♦ Updates are sent when you do a time jump while a recording is paused
- ♦ Support for grouping in DIS
- ♦ Changes to behavior when resuming play after stopping playback
- ♦ Logger dialog boxes are now modal
- ♦ New configuration parameters.

Grouping (DIS Only)

When the Logger forces an update, it sends PDUs in groups, with a delay between each group. This prevents flooding of the network. To configure grouping, you specify **groupCount** and **groupDelay** in *logger.mtl*. The default **groupCount** is `INT_MAX`, which essentially is no grouping (thus replicating the behavior of the Logger in previous versions). The default **groupDelay** is .5 seconds.

Resuming Playback After Stopping

When you stop playback of a recording, the Logger deletes the objects in the simulation, but does not reset the recording time. If you resume play without rewinding the recording, the Logger behaves as follows:

- ♦ If a recording has checkpoints, the Logger sets the time to the current time in the recording, does a time jump, reads the closest checkpoint, then starts sending updates.
- ♦ If a recording does not have checkpoints, the Logger sets the time to the current time, reads the file to this time, then starts sending updates.
- ♦ DIS only: By default, the DIS Logger behaves as described above. However, if you set the **jumpOnStart** parameter to 0, the Logger does not determine the state of the exercise. It just starts sending updates from the point in the recording at which you stopped.

New Configuration Parameters

- ♦ **LGR_DEBUG** specifies use of the debug or release version of the Logger, for correct loading of plug-ins.
- ♦ **LGR_OS** specifies the operating system, for correct loading of plug-ins.
- ♦ **readAllEntities** specifies whether or not to parse the entire logger recording at the start of playback to read all entity names and IDs. For large files this can take considerable time. The default value is 1. If you set **readAllEntities** to 0, you will not be able to filter by Entity ID/Name.
- ♦ **sendSizeInPdu** (DIS only) specifies the send size of a PDU based on the size defined in the PDU header, rather than the packet size. PDUs that are not aligned on 4 byte boundaries, will have different sizes. It is thus necessary to send the size defined in the PDU rather in the Packet. The default value is 1 (enabled). If **sendSizeInPdu** is set to 0, there will be unbundling messages in VR-Link-based programs, however everything should work as normal.

Fixed Bugs

The following problems have been fixed:

- ♦ Large file support now works correctly in UNIX systems.
- ♦ Dead entities no longer send heartbeats.
- ♦ You can now skip to the end or start of a recording multiple times while the Logger is stopped without any problems.
- ♦ Time jumps to the middle of a recording now work.
- ♦ Reverse time jumps in DIS now work correctly.
- ♦ Updates are now sent correctly during time jumps when playing and paused.
- ♦ Use of subscription control in HLA 1516 no longer causes a crash.
- ♦ Data is no longer scaled multiple times for HLA 13 and HLA 1516.

Known Problems

Logger 3.11.3 has the following known problems:

- ♦ The lgrTrim utility does not work with HLA recordings.

