



MÄK Data Logger 3.11 Release Notes

This document provides the following release-specific information for MÄK Data Logger 3.11:

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
Faster Timeskip During Playback	4
Improved Console Functionality in Windows	6
API Changes	6
Close File Option	6
Updated Parameters for lgrTrim	6
Fixed Bugs	7
Known Problems	7

MÄK Technologies®, VR-Forces®, RTIsby®, and VR-Link® are registered trademarks of MÄK Technologies, Inc.

Copyright © 2005 MÄK Technologies, 10 Fawcett St., Cambridge, MA 02138
All rights reserved.
Document ID: LOG-3.11-2-050210

Systems Supported

Logger 3.11 is being released for the following platforms:

- ♦ PC with Windows 2000/XP
- ♦ Red Hat Linux 8.0 and 9.0
- ♦ 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 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
- ♦ 110 MB disk space
- ♦ 64 MB RAM.

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

- ♦ Linux - 110 MB
- ♦ Sun Solaris - 144 MB
- ♦ SGI IRIX - 285 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 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 2.3.x
- ♦ DMSO RTI NG v6.

Logger 3.11 is compatible with applications that use earlier versions of VR-Link if they support versions of the RPR FOM listed above.

DIS only

Logger 3.11 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 works with the MÄK RTI 2.x, and DMSO RTI NG v6.

Logger 3.11 works with older versions of the MÄK RTI, but we recommend using it with release 2.x.

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 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 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 has the following changes and new features:

- ♦ Faster timeskip during playback
 - New file format
 - New configuration parameters
 - New configuration options in the Configuration dialog box
 - File conversion utility
- ♦ Improved console functionality in Windows
- ♦ API Changes
- ♦ Close File option
- ♦ Updated parameters for lgrTrim.

Faster Timeskip During Playback

The new indexing features support faster skipping during playback, particularly for large Logger files. When you record an exercise with indexing enabled, the Logger periodically saves checkpoints with the complete state of the exercise. When you skip forward through the recording, the Logger moves to the nearest checkpoint prior to the requested skip time, then it fast forwards through the recording to the requested time. This process is transparent to Logger users; however, you must enable indexing for it to work.

If you do not enable indexing, recordings are made identically to those in previous versions of the Logger. These recordings, as well as those made with previous versions of the Logger cannot take advantage of the faster timeskip feature.

New File Format

The Logger file format has been changed to support indexing. Logger 3.11 can play recordings made with previous versions of the Logger. However, previous versions of the Logger cannot play indexed recordings made with Logger 3.11.

Converting Non-Indexed Recordings to Indexed Recordings

If you load a logger recording that is not indexed, the Logger asks you if you want to convert it to indexed format. If you click Yes, the Logger asks you if you want to save a backup copy of the non-indexed recording. If you answer yes, the recording is saved in the format *filename.bak.lgr*. If you always want to convert non-indexed files, set the **autoCheckpoint** parameter to 1 in *logger.mtl*. When **autoCheckpoint** is set to 1, the conversion prompts are not displayed.

You can convert non-indexed recordings directly using the utility *lgrconvert.exe*. This is a command line utility. The syntax for using it is:

```
lgrconvert [arguments] input_file output_file
          checkpoint_frequency
```

where:

- ♦ *input_file* – specifies the logger recording that you want to convert
- ♦ *output_file* – specifies a file name for the converted recording
- ♦ *checkpoint_frequency* – specifies the interval, in packets received, at which to create checkpoints.
- ♦ *arguments* – specifies any of the following optional arguments:
 - *-h* – displays the syntax for the utility
 - *-Q* – specifies use of large files (64-bit file IO)
 - *-F fedFileName* – for HLA only, specifies the FED file to use. Default: *VR-Link.fed*.
 - *-f fom_mapper* – for HLA only, specifies a FOM Mapper. Default: RPR FOM 1.0.

i

When you convert a recording with the HLA Logger, it converts the recording using the FOM that you specified when you started the Logger, which might not be the FOM used to record the exercise. For example, you record an exercise using RPR FOM 1.0. Now, you run the Logger using RPR FOM 2.0, draft 17. You open the recording and convert it. It is converted to RPR FOM 2.0, draft 17.

New Configuration Parameters

To support the new indexing features, the Logger has the following new configuration parameters:

- ♦ *enableIndexing* – enables or disables indexing.
- ♦ *checkpointFreq* – specifies how often, in packets, to checkpoint the recording.
- ♦ *autoCheckpoint* – specifies whether or not to automatically convert non-indexed recordings to indexed format. Default: 0 (do not convert).

New Configuration Options in the Configuration Dialog Box

The Configuration dialog box now has options to enable and disable indexing and to set the checkpoint frequency.

Improved Console Functionality in Windows

The behavior of the text-based interface in Windows has been improved. It now supports the following keystrokes:

- ♦ Esc clears the line
- ♦ Tab inserts a space
- ♦ Backspace backspaces.

API Changes

The following classes in the Logger API have been renamed:

Table 1: Class name changes

Old Class	New Class
LoggerFederateAmbassador	DtLoggerFederateAmbassador
TcpSendObj	DtTcpSendObj
DumpObj	DtDumpObj
FilePrinter	DtFileDumpPrinter
DIFom	DtLgrFom
StringRef	DtStringRef
LaSocketMgr	DtSocketMgr
DtDISSimulate	DtDisSimulate
DtDISSource	DtDisSource
DtDISRPRDBWriter	DtDisRPRDBWriter

Close File Option

In previous versions of the Logger, you could not close a recording. You could only open a different recording, which effectively closed the previously open one. In Logger 3.11, you can close recording with the File → Close File option. When you close a file, the Logger returns to a neutral state.

Updated Parameters for lgrTrim

The options to lgrTrim have been updated as follows:

- ♦ `-t start_time [end_time]` – trims the recording to all packets between the specified *start_time* and the *end_time* or end of the recording.
- ♦ `-n start_packet [end_packet]` – trims the recording to all packets between the specified *start_packet* and *end_packet* or end of the recording.
- ♦ `-h` – prints a help message.

Fixed Bugs

The following problems have been fixed:

- ♦ There are currently no crashing problems when using the MÄK RTIsPy LRC GUI with the Logger.
- ♦ In DIS, you no longer get an error message when you play a recording immediately after you have recorded it.
- ♦ Skip time in HLA no longer performs poorly if you try to skip a large amount of time or try to skip backwards in the recording.

Known Problems

Logger 3.11 has the following known problems:

- ♦ When you export data to a database, if objects in the simulation have articulated parts, you may receive errors message indicating that there is a problem copying the articulated parts data.

