



VR-Link 4.0.3 Release Notes

This release note provides the following release-specific information for VR-Link Release 4.0.3:

Systems Supported	2
Disk Space Requirements	2
Compiler Compatibility on Windows	2
License Manager.....	3
Using Libraries and Binaries Built with Visual Studio 2005 and Later.....	3
Patch Required for AMD Dual-processor Windows PCs.....	4
Backward Compatibility	4
Network Compatibility.....	4
RPR FOM Versions Supported.....	5
RTI Support	5
New Features and Changes	5
Known Problems	7

Copyright © 2011 VT MÄK, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
VR-Exchange™ and VR-Vantage™ are trademarks of VT MÄK. MÄK Technologies®, VR-Forces®, RTIsPY®,
B-HAVE®, and VR-Link® are registered trademarks of VT MÄK.
Document ID: VRL-4.0.3-3-110725

Systems Supported

Table 1 lists the platforms currently supported by MÄK VR-Link 4.0.3. Please contact MÄK if you are interested in purchasing VR-Link for other platforms. Application code must be built with the indicated compilers in order to link to VR-Link libraries.

Table 1: Platforms supported

Platform	Compiler
Red Hat Enterprise Linux Workstation 4.0	default compiler
Red Hat Enterprise Linux Workstation 5.0. (32 bit and 64 bit libraries)	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 and 64 bit) Microsoft Visual C++ 10.0 (32-bit and 64 bit)

Disk Space Requirements

Linux installations require approximately 502 MB of disk space for Red Hat WS 4.0 and 581 MB for Red Hat WS 5.0. On Windows, you need approximately 550 MB of disk space.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1, 8.0, 9.0, and 10.0 (some products are not available on all compilers). When you run MÄK products together, for example, the Logger and a VR-Vantage application, we strongly recommend that you run versions compiled with the same compiler. Mixing products compiled with different versions of the compiler on the same computer can result in program instability.

License Manager

To run the licensed version of VR-Link, you must install license management software. VR-Link 4.0.3 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 VR-Link 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 VR-Link 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>

A license manager FAQ is available at:

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

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.

Unfortunately, the C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later cannot 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 distributes 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.

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. The MSVC++ 10.0 version uses the default values for these flags.

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.

Backward Compatibility

A high degree of compatibility was maintained between 4.0 and 3.13.x. With the exception of the *DtRadioTransmitterRepository* class, all applications should be source compatible. If you are upgrading from VR-Link 3.12 or earlier, be aware that several class interfaces may have changed. Please consult the release notes for all intermediate releases to review any changes. Specifically, articulated parts were updated in VR-Link 3.11.x, and the Networking API was significantly updated in VR-Link 3.12. Please see *VR-Link Developers Guide* for details about the current implementation of articulated parts.

Network Compatibility

HLA only

VR-Link 4.0.3 is compliant with:

- ♦ RPR-FOM 0.5, 0.7, 0.8, 1.0, and a subset of 2.0 (draft 6, 14, and 17)
- ♦ MÄK RTI 2.x, 3.x, 4.x (HLA 1.3 and 1516-2000 only)
- ♦ MÄK RTI 4.0.4 (HLA Evolved)
- ♦ Pitch RTI 1.3 C++ interface.

Other RTIs that support the HLA 1.3 specification, the SISO DLC HLA API 1516 version of the IEEE 1516 specification (SISO-STD-004.1-2004), and HLA Evolved. To use an RTI with VR-Link it must use the same operating system and be built with the same compiler.

DIS only

VR-Link 4.0.3 supports DIS 4, 5, and 6, and can therefore interoperate with DIS applications of any of these versions.

RPR FOM Versions Supported

VR-Link 4.0.3 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. By default, VR-Link 4.0.3 uses RPR FOM 1.0.

VR-Link does not officially support RPR FOM 2 Draft 18 at this time. However, the draft appears to be similar enough to RPR FOM 2 Draft 17 that the 2.0017 FOM Mapper may be used for federates wishing to interoperate with RPR FOM 2 Draft 18.

If you want to use a version of the RPR FOM other than 1.0, pass the version number (0.5, 0.7, 0.8, 2.0006, 2.0014, or 2.0017) to the *DtRprFomMapper* constructor and pass the resulting object to the *DtExerciseConn* constructor. Also, make sure you are using a federation execution name that corresponds to the right FED file. For example:

```
DtExerciseConn conn("VR-Link20017", "MyApp", new DtRprFomMapper(2.0017));
```

VR-Link examples like *f18* and *hlaNetdump* have a command line option, `--rprFomVersion`, that you can use to choose a RPR FOM version, using one of the version numbers listed in the previous paragraph.

RTI Support

VR-Link 4.0.3 has been tested with the Pitch RTI 1.3 and MÄK RTI 4.0.4.

If you use the MÄK RTI, remember to make sure that the VR-Link can find your FED file or FDD file, and optional *rid.mtl*. Put the FED file or FDD file in the directory from which you are running, or set the environment variable `RTI_CONFIG` to the directory that contains it. See *MÄK RTI Reference Manual* for a list of the options for specifying the location of the *rid.mtl* file.

Applications built with VR-Link 4.0.3 for HLA can use any RTI that conforms to the relevant dynamic link compatible HLA standard (1.3, SISO DLC 1516, HLA Evolved) and was built using the same operating system and compiler.

New Features and Changes

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

The HLA Evolved libraries included with the MÄK RTI 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 which 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.

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

When using HLA Evolved:

- VR-Link 4.0, 4.0.1, and 4.0.2 are only compatible with MÄK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.
- VR-Link 4.0.3 and later releases are only compatible with MÄK RTI 4.0.4 and later releases.
- VR-Exchange 2.0 and MÄK Data Logger 5.0 are only compatible with MÄK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.

Known Problems

This release of VR-Link has the following known problems:

- ♦ The VR-Link Code Generator may have problems with complex FOMs. We have made every effort to verify that it works with a broad range of FOMS and FED files. However, we are unable to verify that it works for all FOMs. In addition, small errors in the FOM may prevent the Code Generator from correctly processing the FOM at all. If the Code Generator is not able to process your FOM, we would be happy to help. Please send a copy of your FOM to support@mak.com and we will work with you to correctly generate your FOM.
- ♦ VR-Link correctly encodes and decodes EnvironmentalProcess objects for RPR FOM 2, Draft 14. However, the RPR FOM specifies a way of encoding which prevents VR-Link from correctly decoding modified or custom EnvironmentalRecords. Unfortunately, many MÄK products built with VR-Link use customized EnvironmentalRecords. If you use these products or if you use customized records in your applications, please do not use the RPR FOM 2 Draft 14 FOM Mapper. This is not a problem in other FOM versions, and it has been resolved in future versions of the FOM.
- ♦ The RTI 1516 API specifies that all strings passed to and from the RTI are handled as wide strings. VR-Link (and the MÄK RTI) store strings internally as narrow strings. This means that true multibyte wide strings used by RTI Federates may not be handled correctly when received by VR-Link. VR-Link handles name reservation wide strings correctly, however, as this is a 1516-only function. VR-Link provides conversion functions from Wide To Narrow/Narrow to Wide string conversion in *vlStringUtil.h*.
- ♦ On non-windows platforms, the classes *DtSharedMemoryPoolManager* and *DtSharedMemoryPoolClient* (*vlShmPool.h*) sometimes have difficulty creating a large memory pool (greater than 1 MB) if the pool name is longer than two or three characters. The classes refuse to create the memory pool even if the requested pool is smaller than the system resource SHMMAX. Setting the pool name to fewer than three characters will work around this problem.
- ♦ Section 6.10, “[Interoperability Between HLA 1.3 and IEEE 1516 Federates](#)”, in *VR-Link Developers Guide* describes interoperability issues that arise when you try to run an HLA 1.3 federate with an IEEE 1516 federate. In particular it describes the requirement to use an identical FED or FDD file for all federates. In future releases, we intend to rectify this situation, and allow run-time interoperability between federates that have loaded a 1.3-style FED file and federates that have loaded a 1516-style XML file that otherwise includes the same set of classes, attributes and parameters.

