



VR-Link 4.0.9 Release Notes

This release note provides the following release-specific information for VR-Link Release 4.0.9:

Systems Supported	2
Disk Space Requirements	2
Compiler Compatibility on Windows	2
License Manager.....	3
Network Compatibility	3
Backward Compatibility	4
RPR FOM Versions Supported	5
RTI Support	5
New Features and Changes	6
Known Problems	6

Copyright © 2013 VT MÄK, 150 Cambridge Park Drive, 3rd Floor, Cambridge, MA 02140 All rights reserved. VR-Exchange™, VR-TheWorld™, and VR-Vantage™ are trademarks of VT MÄK. MÄK Technologies®, VR-Forces®, RTIspey®, B-HAVE®, and VR-Link® are registered trademarks of VT MÄK.
Document ID: VRL-4.0.9-3-130806

Systems Supported

[Table 1](#) lists the platforms currently supported by MÄK VR-Link 4.0.9. 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, 6.0 (32 bit and 64 bit libraries)	default compiler
SUSE 11 Linux (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 and 6.0. 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 4.0.9 uses FLEXlm 11.11 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 32 bit: <ftp://ftp.mak.com/out/MAKLicenseManager-win-setup.exe>
- ♦ Windows 64 bit: <ftp://ftp.mak.com/out/MAKLicenseManager-win64-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>

Network Compatibility

HLA only

VR-Link 4.0.9 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 or later (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 4.0.9 supports DIS 4, 5, and 6, and can therefore interoperate with DIS applications of any of these versions.

Backward Compatibility

VR-Link 4.0.9 is highly compatible with past versions. Note the following:

- When moving from VR-Link 3.x to 4.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.
- The encoding and decoding of the EnvironmentRecData in the EnvironmentProcess class in RPR FOM 2 draft 17 changed in VR-Link 4.0.4. Using the new encoder and decoder functions breaks compatibility with older versions of VR-Link, so you must set the rprFomRevision to at least 4 to use the corrected functions.
- VR-Link 4.0.4 made some small changes to the class interfaces for the HLA *DtExerciseConnInitializer* and *DtExerciseConn*. In HLA 1.3 and HLA 1516-2000 each federate could specify a Federate Type. This type was simply a string, and it was frequently referred to as Federate Name. In VR-Link we also referred to it as Federate Name, and many of our variable and function names reflected this, especially in *DtExerciseConn* and its initializer class. However, in HLA Evolved a new and separate parameter was added called Federate Name. The idea is that Federate Type can be used to distinguish different categories of federates, and its only practical use is in save-and-restore.

The Federate Name is optional and is used to uniquely identify a federate. The name must be unique, and a federate will not be able to join a federation if it duplicates the name of another federate. If it is not specified, the RTI assigns a unique name to the federate. As a result of this change, some of our VR-Link functions and variables were inconsistent and confusing. This also led to different behavior in HLA Evolved than in HLA 1.3 and 1516-2000 for federates that used the same functions.

In VR-Link 4.0.4 we eliminated these inconsistencies, but this means that some functions have changed names. Now Federate Type will always have the same meaning regardless of HLA version, and Federate Name only exists in HLA Evolved. This means that if you previously used the function `DtExerciseConn::fedName()` you should now use `DtExerciseConn::fedType()`. If you previously used `DtExerciseConnInitializer::setFederateName()` or `DtExerciseConnInitializer::federateName()` you should now use `DtExerciseConnInitializer::setFederateType()` or `DtExerciseConnInitializer::federateType()`.

RPR FOM Versions Supported

VR-Link 4.0.9 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 4.0.9 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 4.0.9 has been tested with the MÄK RTI 4.1 for HLA 1.3, HLA 1516, and HLA Evolved. It has also been tested with Pitch's pRTI 4.3.1 for HLA 1.3 and HLA Evolved.

If you use the MÄK RTI, remember to make sure that 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 4.0.9 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 4.0.9 is a maintenance release. It has the following new features and updates:

- ♦ Support for DI-Guy in HLA Evolved.
- ♦ Updated the Logger Control Toolkit to match the current release of the MÄK Data Logger.

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 5.10, “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.

