VR-Link 4.0 Release Notes

This release note provides the following release-specific information for VR-Link Release 4.0:

Copyright © 2010 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-4.0-3-101110

Systems Supported

Table 1 lists the platforms currently supported by MÄK VR-Link 4.0. 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)

Disk Space Requirements

Linux installations require approximately 80 MB for VR-Link files and an additional 120 MB for documentation files.

On PCs, you need 320 MB for VR-Link files and 120 MB for all documentation files.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1, 8.0, and 9.0. 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 can result in program instability.

License Manager

To run VR-Link, you must install license management software. VR-Link 4.0 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. For customers who plan to use MÄK products on machines that do not have a compiler installed, MÄK has historically distributed a copy of the C Runtime Libraries with MÄK products. These libraries were put in the *bin* directory used by the MÄK products. MÄK products would then use the libraries in the *bin* directory and customers would not have a problem if copies of the libraries were not already installed.

Unfortunately, with the release of the new C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later, the libraries can no longer 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 is distributing 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.

Running the installer requires Administrator privileges for the machine the installer is run on. MÄK has chosen to not integrate the MÄK installer and the Microsoft installer so as not to require users to have Administrator privileges to install MÄK products. Therefore, if you who do not have a compiler installed, or get error messages like “Software has not been installed correctly, please re-install”, you must apply the patch.

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."

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 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. The HLA Evolved version of VR-Link is compatible only with MÄK RTI 4.0 and later, or other HLA Evolved-compliant RTIs.
- Pitch RTI 1.3 C++ interface.

Other RTIs that support the HLA 1.3 specification, the SISO DLC HLA API 1516 (SISO-STD-004.1-2004 version of the IEEE 1516 specification, and HLA Evolved.)

DIS only

VR-Link 4.0 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 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 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 has been tested with the Pitch RTI 1.3 and MÄK RTI 4.0.1.

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.

New Features and Changes

VR-Link 4.0 adds support for the HLA Evolved API. Other new features and changes include:

- Support for HLA Evolved
- Changes to radio transmitters
- Added the ability to remove a region from an interaction subscription region list
- Added the ability to turn off articulated parts dead reckoning for gateway-like applications.
- In DIS, you can now send and receive SEES PDUs through entity publishers and reflected entities.
- VR-Link is no longer distributed with a TENA interface. Customers who have TENA requirements should contact MAK for more information.

Support for HLA Evolved

HLA Evolved (IEEE 1516-2010) is the latest version of the HLA 1516 specification. It provides the dynamic-link compatibility that the original 1516 specification did not. It includes features such as update rate reduction and modular FOMs.

VR-Link now supports HLA Evolved through an additional library, *vlHLA1516e*, which complements the HLA 1.3, and HLA 1516 implementations.

To use HLA Evolved, federates must be recompiled with the DtHLA_1516_EVOLVED preprocessor macro defined. HLA Evolved federates in VR-Link are compiled exactly the same way as HLA 1.3 and HLA 1516 federates, except the preprocessor definitions change, as does the link line. Please see *VR-Link Developers Guide* for more details.

Additionally, the VR-Link Code Generator has been updated to support the HLA Evolved FOM Scheme and loading FOM Modules.

Radio Transmitter Changes

The *DtRadioTransmitter* class has been modified to correct omissions from the RPR FOM 2.0 specification and to improve ease of use across protocols. Previously, when using a radio transmitter, end users had to set and retrieve the data in the SpreadSpectrum attribute themselves from a generic datatype. Now those attributes are automatically encoded from and decoded into the radio transmitter state repository. The Link16 extension has been similarly modified. Please see the class documentation for details about how to use these methods.

Documentation Updates

VR-Link Developers Guide has been updated for this release. TENA documentation has been removed. Most other changes are minor error corrections.

Bug Fixes

VR-Link 4.0 fixes the following bugs:

- The Code Generator now builds Windows projects with the correct _SECURE_SCL compiler flag. (42588)
- The *Dt3dPoint* class did not initialize correctly. (42187)
- Fixed reliance on case sensitivity in the HLA 1.3 FED file. (37808)
- Fixed file descriptor leaks in *DtExerciseConn*. (40963)
- VR-Link now responds to requestDivestitureConfirmation calls, thereby successfully allowing a negotiated divestiture. (41280)
- VR-Link did not correctly set state to closed in *DtInetTcpSocket* when closeSocket() was called. (42138)
- Moved *DtRecordSet* to the *vl* library because it is protocol-independent, allowing it to be used in other protocol-independent libraries. (41523)
- Fixed broken reference in *VR-Link Developers Guide*. (42796)
- Removed compiler warnings with gcc++ 4.1.1. (42700)
- Made the Code Generator use system style file dialog boxes. (42646)
- *DtInetUdpSocket* recv() and recvToChain() were not working. (42629)
- The Code Generator crashed with certain XML file formats. (42636)
- Made external library environmental variables consistent in project files created by the Code Generator. (42592)
- A license was not always returned when a destructor is called in certain situations. (42528)
- Fixed an obscure crash when DtExerciseConn::usingAdvisories() is disabled and the RTI had an exception. (42529)
- The default project file for hlaNetdump did not include all the required source files (42458)
- DtInetUdpSocket::joinMulticastGroup() did not join with the correct device. (41637)
- Added support for *DtSuppEmisEntStatePdu* to the *DtEntityStatePublisher*. (47120)
- VR-Link crashed when receiving 0-typed articulated parts. (41723)
- The *DtEnvironmentTypeRecord* constructor incorrectly hid the base class constructor. (41863)
- VR-Link sent unnecessary EE PDUs. (42330)
- VR-Link handles HLA 1516 name reservation faster. (41843)
- Fixed const correctness problem with SwitchArg::getValue(). (41836)

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.