



VR-Exchange 2.0.1 Release Notes

This release note provides release-specific information for VR-Exchange Release 2.0.1.

System Requirements.....	2
Platforms Supported.....	2
External Library Requirements	2
Using Libraries and Binaries Built with Visual Studio 2005 and Later	3
License Manager.....	3
Network Compatibility.....	4
RTI Support	4
Support for HLA Evolved.....	4
Brokers Provided.....	6
New Features and Updates.....	8
Documentation Updates	8
Bug Fixes	9
Known Problems	10

Copyright © 2012 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: VRE-2.0.1-2-120417

System Requirements

This section lists supported platforms and software requirements.

Platforms Supported

VR-Exchange 2.0.1 supports the operating systems listed in [Table 1](#).

Table 1: Platforms supported

Broker	Platform	Compiler
HLA, DIS	Red Hat Enterprise Linux 4.0, 5.0	Default compiler
HLA, DIS, TENA	PC with Windows XP/-Vista/Windows 7	Microsoft Visual C++ 7.1, 8.0
HLA, DIS	PC with Windows XP/-Vista/Windows 7	Microsoft Visual C++ 9.0 (32-bit) Microsoft Visual C++ 10.0 (64-bit)



You must have administrator privileges to install MÄK products on Windows Vista.

External Library Requirements

VR-Exchange 2.0.1 is built with VR-Link 4.0.4 and pThreads 2.70. The TENA broker is built with TENA 5.2.2.

TENA users must install TENA middleware and any libraries required by their LROM. HLA users must install an RTI.

VR-Exchange 2.0.1 uses the following versions of Qt:

- ♦ Qt 4.7.4 – VC++ 10.0
- ♦ Qt 4.7.3 – VC++ 9.0
- ♦ Qt 4.5.0 – VC++ 7.1 and 8.0, Linux.

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.

License Manager

To run the licensed version of VR-Exchange, you must install license management software. VR-Exchange 2.0.1 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 VR-Exchange 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-Exchange 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>

License manager documentation is available at:

<http://www.mak.com/license-support.html>

Network Compatibility

HLA only

VR-Exchange 2.0.1 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 Evolved support is only with MÄK RTI 4.0.4 or later)
- ♦ 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-Exchange it must use the same operating system and be built with the same compiler.

DIS only

VR-Exchange 2.0.1 supports DIS 4, 5, and 6, and can therefore interoperate with DIS applications of any of these versions.

RTI Support

We recommend using the latest version of the MÄK RTI. However, VR-Exchange should work with other RTIs that conform to the HLA 1.3 specification, the IEEE 1516 SISO DLC API, or HLA Evolved and are built using the same compiler as your version of the VR-Exchange.

When using the MÄK RTI, remember to make sure that the VR-Exchange 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 the directory that contains them.

Support for HLA Evolved

VR-Exchange 2.0.1 updates the HLA Evolved version to be compatible with MÄK RTI 4.0.4, VR-Link 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 4.0 through 4.0.3 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 that 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.

This problem affects MÄK products built against the MÄK RTI and VR-Link that support HLA Evolved, such as VR-Exchange 2.0. VR-Exchange 2.0.1 has been built against VR-Link 4.0.4 and is now compliant with the final version of the standard.

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

When using HLA Evolved:

- VR-Exchange 2.0 is only compatible with MÄK RTI 4.0, 4.0.1, 4.0.2, and 4.0.3.
- VR-Exchange 2.0.1 and later releases are only compatible with MÄK RTI 4.0.4 and later releases.

Brokers Provided

VR-Exchange 2.0.1 includes brokers for:

- ♦ HLA 1.3
- ♦ HLA 1516 (SISO Standard DLC C++ API for IEEE 1516)
- ♦ HLA Evolved
- ♦ TENA 5.2.2
- ♦ DIS.

[Table 2](#) lists the translators for each broker.

Table 2: Broker translators

Translator	HLA	DIS	TENA
Acknowledge	X	X	X
Action Request	X	X	
Action Response	X	X	
Aggregate	X	X	X
Collision	X	X	X
Comment	X	X	X
Create Entity	X	X	
Data	X	X	X
Data Query	X	X	
Designator	X	X	X
Detonation	X	X	X
Emitter System	X	X	X
Entity State	X	X	X
Environment Process	X	X	X
Event Report	X	X	
Fire	X	X	X
Gridded Data	X	X	
IFF	X	X	X
Logger Control	X	X	
Radio Receiver	X	X	X
Radio Signal	X	X	X
Radio Transmitter	X	X	X
Remove Entity	X	X	

Table 2: Broker translators

Translator	HLA	DIS	TENA
Repair Complete	X	X	
Repair Response	X	X	
Resupply Cancel	X	X	
Resupply Offer	X	X	
Resupply Received	X	X	
Service Request	X	X	
Set Data	X	X	X
Start	X	X	X
Stop	X	X	X
Transfer Control	X	X	



VR-Exchange can translate environmental processes. However, it cannot convert custom data inside these PDUs. Therefore, it is recommended that you disable environmental process translation if you are using VR-Forces and translating between DIS and HLA.

New Features and Updates

VR-Exchange 2.0.1 is primarily a maintenance release. It has the following minor changes:

- Support for Microsoft Visual C++ 10.0 (64 bit only).
- Support for HLA Evolved (IEEE 1516-2010). This includes an updated list of HLA Connections Settings in the broker configuration dialog box.
- Stealth Control PDUs (from the Stealth Control Toolkit) are no longer supported. The ViewControl Interaction translator has been removed.
- Published and reflected object tables in the broker dialog boxes can now be sorted by clicking on the column headers.
- For HLA 1.3 and 1516, the “Federate Name” parameter has been changed to the more correct term “Federate Type”. Federate Type can be used to distinguish different categories of federates, and its only practical use is in save-and-restore. The Federate Name parameter has a special meaning for HLA Evolved. Therefore it could no longer be used to specify the federate type, as in previous versions. The Federate Type parameter is available for all versions of HLA. The Federate Name parameter is only used in HLA Evolved.
- Added a configuration option for the subnet mask to the DIS broker.
- The DIS broker configuration dialog box now has a combo box which lists the valid network interfaces that can be used for the DIS Multicast Device.

Documentation Updates

The documentation of the Broker APIs and Portal API is now included in the class documentation.

Bug Fixes

VR-Exchange fixes the following problems that were present in previous releases:

- ♦ Fixed issues in translating radio signals and radio transmitters. 45130
- ♦ The class documentation for the examples listed the same *main.cxx* file for all four examples. 42547
- ♦ Fixed an HLA broker crash that could occur when an Environment Process ID is 0:0:0 and the broker publication conditions specified that a valid ID was not needed. 42885
- ♦ Fixed terminology used in HLA broker configuration. For HLA 1.3 and HLA 1516, Federate Name has been renamed Federate Type. HLA Evolved has configuration options for both Federate Name and Federate Type. 44283
- ♦ Fixed the messageDump example project to update dependencies on old VR-Link library names. 44305
- ♦ Fixed issues in the example projects so that they now correctly compile and link. 46575
- ♦ The DIS broker now displays the correct values in the Marking Text and DIS ID fields for published and reflected objects. 46121
- ♦ The DIS broker did not translate the Request ID in the Set Data PDU. 47839
- ♦ Fixed several issues with IFF translation. 47840
- ♦ The HLA translator was incorrectly always publishing IFF objects as class EmbeddedSystem.IFFRRB. 47916
- ♦ Fixed translation of the Real-World Time and Simulation Time in the DIS Start/Resume PDU. 47841

Known Problems

VR-Exchange has the following known problems:

- On Linux you might encounter the following message:

```
VR-Exchange caught exception: DtSharedMemoryPoolManager()  
create has error with shmget: Invalid argument().
```

This error may occur:

- When the queue manager cannot reserve enough shared memory for the internal message queues. To correct the error do one of the following:
 - Increase the maximum shared memory allocations on your platform. On Linux the root user can do this by modifying the *shmmax* file:

```
echo "33554432" > /proc/sys/kernel/shmmax
```



This number is typically the maximum number of bytes allocatable in shared memory. Please consult your operating system documentation first.

- Reduce the number of buckets in the message queues.
- Because of write permissions in the */tmp* directory. When VR-Exchange creates shared memory queues, it creates temporary files in */tmp*. The files are called: */tmp/OQ.shared*, */tmp/IQ.shared*, and */tmp/CQ.shared*. If those files exist and the person running VR-Exchange does not have permission to open them for writing, VR-Exchange will print out the error. To correct the problem, delete the files.
- Under Linux, the TCP connection sometimes fails to start. If this happens, you will be unable to successfully start any brokers. Shutting down the portal and starting it again usually fixes the problem.
- If you are using the MÄK RTI in lightweight mode, misconfiguring the HLA connections can cause a feedback loop to occur. A symptom of this is seeing the number of entities rapidly increase, with the same Entity ID. Although this is not a bug, it is mentioned here to help you diagnose this problem should it occur.

To avoid such situations, when using the RTI in lightweight mode, make sure the first three characters of each connection's HLA Execution Name are unique.