



VR-Exchange 2.2 Release Notes

This release note provides release-specific information for VR-Exchange Release 2.2.

System Requirements.....	2
Platforms Supported.....	2
External Library Requirements	2
License Manager.....	3
Network Compatibility.....	3
RTI Support	3
Support for HLA Evolved.....	4
Brokers Provided.....	5
New Features and Updates.....	7
Documentation Updates	7
Bug Fixes	7
Known Problems	8

Copyright © 2013 VT MÄK, 150 CambridgePark 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®, RTIsby®, B-HAVE®, and VR-Link® are registered trademarks of VT MÄK.
Document ID: VRE-2.2-2-130410

System Requirements

This section lists supported platforms and software requirements.

Platforms Supported

VR-Exchange 2.2 supports the operating systems listed in [Table 1](#).

Table 1: Platforms supported

Broker	Platform	Compiler
HLA, DIS	Red Hat Enterprise Linux 5.0	Default compiler
	Red Hat Enterprise Linux 6.0	Default compiler (64-bit)
HLA, DIS, TENA	PC with Windows XP/- Vista/Windows 7	Microsoft Visual C++ 8.0
HLA, DIS	PC with Windows Vista/Windows 7	Microsoft Visual C++ 10.0 (32-bit and 64-bit)



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

External Library Requirements

VR-Exchange 2.2 is built with VR-Link 4.0.7 and pThreads 2.7.0. 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 2.2 uses Qt 4.7.4.

License Manager

To run the licensed version of VR-Exchange, you must install license management software. VR-Exchange 2.2 uses FLEXlm 11.8. 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> (for 32 bit computers)
- ♦ Windows: <ftp://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe> (for 64 bit computers)
- ♦ 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

The VR-Exchange 2.2 HLA brokers are 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.
- ♦ Pitch RTI HLA Evolved 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), or 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

The VR-Exchange 2.2 DIS broker 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

The VR-Exchange 2.2 HLA Evolved broker is compatible with MÄK RTI 4.0.4 or later, VR-Link 4.0.4 or later, 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. MÄK 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 2.2 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

Table 2: Broker translators

Translator	HLA	DIS	TENA
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	
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 2.2 has the following new features:

- ♦ Support for Red Hat Enterprise Linux 6 (64-bit).
- ♦ Support for Microsoft Visual C++ 10 (32-bit).
- ♦ Added a subnet mask configuration option to the DIS broker.
- ♦ Source code for brokers is included in the installation package.
- ♦ The Data View has been replaced with a Statistics graph. The new graph plots the same information contained in the old Data View. It also plots timing and object/interaction counts from each of the brokers. For details, please see [Section 3.7, "The Statistics View"](#), in *VR-Exchange Users Guide*.
- ♦ You can now search the object list in the Object View. For details, please see [Section 3.4.4, "Searching the Object List"](#), in *VR-Exchange Users Guide*.
- ♦ You can now monitor the translation process for specific objects. For details, please see [Section 3.8, "Monitoring Translations"](#), in *VR-Exchange Users Guide*.
- ♦ VR-Exchange now has a plug-in API. This allows you to modify brokers without completely rebuilding them. For details, please see API documentation.

Documentation Updates

VR-Exchange Users Guide and online help have been updated for this release.

Bug Fixes

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

- ♦ Only the lower 32 bits of the frequency for radio transmitter objects was being translated. 48680
- ♦ Changes to the DIS Broker's Timeout Interval parameter weren't being saved to the configuration file. 48654
- ♦ Only the lower 32 bits of the frequency for radio transmitter objects was being translated. 48680, 48682, 48718
- ♦ Fixed bug where Emitter Systems were being displayed in the Portal with the wrong name. This was preventing filtering from working correctly. 48717
- ♦ Translating entities with multiple emitter systems would cause all but one of the emitter systems to drop and not be translated. 48880
- ♦ IFFs were not being given unique IDs, and were being dropped as if they were duplicate objects. 48760
- ♦ If a portal entity did not have a DIS ID, the DIS broker was unable to assign it one itself, even when Remap DIS IDs was enabled. 49483

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.