



VR-Exchange 2.7 Release Notes

This release note provides release-specific information for VR-Exchange Release 2.7.

System Requirements	2
Platforms Supported	2
External Library Requirements	2
Compiler Compatibility on Windows	3
License Manager	3
Network Compatibility	4
RTI Support	4
Support for HLA Evolved	5
Brokers Provided	6
Additional Brokers	8
New Features and Updates	8
Fixed Bugs	8
Known Problems	9

Copyright © 2021 MAK Technologies, 150 Cambridge Park Drive, Third Floor, Cambridge, MA 02140. All rights reserved. VR-Exchange™, VR-TheWorld™, VR-Vantage™, DI-Guy™, and DI-Guy Scenario™ are trademarks of MAK Technologies. MAK Technologies®, VR-Forces®, RTIspy®, B-HAVE®, and VR-Link® are registered trademarks of MAK Technologies.
Document ID: VRE-2.7-2-210330

System Requirements

This section lists supported platforms and software requirements.

Platforms Supported

VR-Exchange 2.7 supports the operating systems listed in [Table 1-1](#).

Table 1-1: Platforms supported

Broker	Platform	Compiler
HLA, DIS	Red Hat Enterprise Linux 7.0	Default compiler (64-bit)
	Red Hat Enterprise Linux 8.0	
	PC with Windows 10	Microsoft Visual C++ 12.0 (64-bit)
		Microsoft Visual C++ 14.0 (64-bit)
DDS	Red Hat Enterprise Linux 7.0	Default compiler (64-bit)
	Red Hat Enterprise Linux 8.0	
	PC with Windows 10	Microsoft Visual C++ 12.0 (64-bit)
		Microsoft Visual C++ 14.0 (64-bit) (RTI DDS only.)

The VC 14 build is binary compatible with all MAK VC 15 products.

If you require TENA support, contact support@mak.com.

External Library Requirements

VR-Exchange 2.7 is built with VR-Link 5.6.

VR-Exchange 2.7 RH7 uses Qt 5.5.1. VC12 uses Qt 5.10.1. RH8 and VC14 use Qt 5.14.0.

DDS Support

VR-Exchange has the following DDS brokers:

OpenSplice DDS Broker. Requires installation of version 6.3 of the OpenSplice DDS Community Edition. Please contact www.prismtech.com/opensplice/opensplice-dds-community for availability of this version. This broker is supported on all platforms except VC14.

RTI DDS Broker. Requires the RTI Connext DDS Software. You can download it from <https://www.rti.com/>. The RTI DDS broker is supported on all platforms.

Compiler Compatibility on Windows

MAK provides versions of product releases that have been compiled with different versions of Microsoft Visual C++. When you run MAK products together on the same computer, 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. The exception to this is that products built with VC 14 and VC 15 are binary compatible.

License Manager

To run the licensed version of VR-Exchange, you must install license management software. VR-Exchange 2.7 uses FLEXlm 11.13.0.2. If you are upgrading from a version of the 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:
<http://ftp.mak.com/out/MAKLicenseManager-win-setup.exe> (for 32-bit systems)
<http://ftp.mak.com/out/MAKLicenseManager-win64-setup.exe>
- ♦ Linux: <http://ftp.mak.com/out/MAKLicenseManager-linux-setup.tar.gz> (for 32-bit systems)
<http://ftp.mak.com/out/MAKLicenseManager-linux64-setup.tar.gz>

License manager support is available at:

<http://www.mak.com/support/licenses>

Network Compatibility

VR-Exchange 2.7 is compliant with RPR FOM 1.0 and RPR FOM 2.0 and includes FOM mappers for many draft versions of the RPR FOM. It is compliant with 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.

VR-Exchange 2.7 supports DIS 4, 5, 6, and 7, and can therefore interoperate with DIS applications of any of these versions.

RTI Support

We recommend using the latest version of the MAK 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 MAK 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 2.7 HLA Evolved broker is compatible with MAK 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 MAK 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 MAK 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 MAK 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 MAK RTI. MAK 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 MAK 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 MAK RTI versions will need to be recompiled against the new version.

Because VR-Link's HLA libraries are built against the MAK 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 MAK products built against the MAK 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 MAK RTI will not be compatible when using HLA Evolved. This change does not affect compatibility between VR-Exchange and the MAK RTI when using HLA 1.3 or 1516-2000.

When using HLA Evolved:

- ♦ VR-Exchange 2.0 is only compatible with MAK 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 MAK RTI 4.0.4 and later releases.

Brokers Provided

VR-Exchange 2.7 includes brokers for:

- ♦ HLA 1.3 RPR FOM 1 and 2.
- ♦ HLA 1516 RPR FOM 1 and 2 (SISO Standard DLC C++ API for IEEE 1516).
- ♦ HLA Evolved RPR FOM 1 and 2.
- ♦ DIS, versions 4–7.
- ♦ DDS (RH7, VC12, VC14).

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

Table 2: Broker translators

Translator	HLA	DIS	DDS	TENA
Acknowledge	X	X		X
Action Request	X	X		
Action Response	X	X		
Active Sonar	X	X		
Additional Passive Activities	X	X		
Aggregate	X	X		X
Areal Object	X	X		
Collision	X	X		X
Comment	X	X		X
Create Entity	X	X		
Data	X	X		X
Data Query	X	X		

Table 2: Broker translators

Translator	HLA	DIS	DDS	TENA
Designator	X	X		X
Detonation	X	X	X	X
Emitter System	X	X		X
Entity State	X	X	X	X
Environment Process	X	X		X
Event Report	X	X		
Fire	X	X	X	X
Gridded Data	X	X		
IFF	X	X		X
Linear Object	X	X		
Logger Control	X	X		
Point Object	X	X		
Propulsion Noise	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.

Additional Brokers

The following brokers are available on request. Please contact your MAK salesperson for details.

Brokers:

- ♦ AFMSTT Broker.
- ♦ ERF Broker.
- ♦ MATREX Broker (Version 7.1)
- ♦ MRF Broker (Version 6.2 / 7.1).

Experimental Brokers (These brokers support a subset of the FOM):

- ♦ ADS-B Broker.
- ♦ AviationSimNet Broker (Version 3.4).
- ♦ COT Broker.

New Features and Updates

VR-Exchange 2.7 has the following new features:

- ♦ Support for VR-Link 5.6 to support VR-Forces 4.9 and VR-Vantage 2.7.
- ♦ Added a flag in Common Broker Settings that dictates whether to load the VR-Forces plug-ins.

Fixed Bugs

This release fixes the following issue that was a problem in previous releases.

- ♦ VR-Exchange now correctly translates non-entity receiver IDs for interactions.

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
```

i

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.
- ♦ On Linux, if you filter entities by geospatial location and you set the degrees, minutes, and seconds to their maximum values, the down arrows in the spin boxes look like they are unavailable. However, they still work.
- ♦ If you are using the MAK 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.

