



VR-Link 5.6

Release Notes

This release note provides the following release-specific information for VR-Link Release 5.6:

Systems Supported.....	2
Disk Space Requirements.....	2
Compiler Compatibility on Windows.....	2
License Manager.....	3
Network Compatibility.....	3
Backward Compatibility.....	4
Using the Compatibility Directory.....	4
Using the Migration Script.....	5
RTI Support.....	5
New Features and Changes.....	5
Fixed Bugs.....	6

Copyright © 2021 MAK Technologies, 150 Cambridge Park Drive, 3rd 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: VRL-5.6-3-210129

Systems Supported

[Table 1](#) lists the platforms currently supported by MAK VR-Link 5.6. Please contact MAK 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 7.0 (64-bit libraries)	default compiler
Red Hat Enterprise Linux Workstation 8.0 (64-bit libraries)	default compiler
PC with Windows 10	Microsoft Visual C++ 12.0 (64-bit)
	Microsoft Visual C++ 14.0 (64-bit)

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

VR-Link for C# supports all Windows compilers.

VR-Link for Java supports all compilers and operating systems.

VR-Link for Java requires Java 8.

Disk Space Requirements

Linux installations require approximately 2 GB of disk space. On Windows, you need approximately 1.5 GB of disk space.

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-Link, you must install license management software. VR-Link 5.6 uses FLEXlm 11.13.0.2. 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:
<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-Link 5.6 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-Link, it must use the same operating system and be built with the same compiler.

VR-Link 5.6 supports DIS 4, 5, 6, and 7, and can therefore interoperate with DIS applications of any of these versions.

Backward Compatibility

VR-Link 5.6 has the following backwards compatibility issues:

- VR-Link versions 5.0-5.1.3 encoded environmental process DIS Version 7 PDUs incorrectly. This has been corrected. Unfortunately this means that VR-Link 5.3.1 will not inter-operate with those versions of VR-Link for DIS 7 Environmental Process PDUs. We recommend upgrading all VR-Link 5.0-5.1.3 federates if you are using that specific configuration.
- VR-Link 5.2 encoded data interactions incorrectly for RPR FOM 2.0. This is fixed in VR-Link 5.3.1. However, this incompatibility causes applications that were built with VR-Link 5.2 to crash when trying to decode VR-Link 5.3.1 interactions.

For backwards compatibility, the incorrect encoding is now provided as RPR 2.0 Revision 1. The correct encoding is RPR 2.0 Revision 2. Use Revision 1 to maintain compatibility with previous versions of VR-Link. Use Revision 2 to be compliant with RPR FOM 2.0.

- VR-Link 5.0 renamed header files to conform to MAK file naming conventions. If you have not already migrated to these new file names, the next section describes the changes.

Using the Compatibility Directory

VR-Link has a *compatibility* directory in its *include* directory. This directory has the same directory structure as the main *include* directory with header files that have the same names as previous versions of VR-Link. The header files in this directory do not duplicate the declarations in the renamed header files, they include the renamed header files themselves. To use the *./include/compatibility* directory, add it to your project's include directory list. You will not have to update the include statements in your project files.

RTI Support

VR-Link 5.6 has been tested with the MAK RTI 4.6 for HLA 1.3, HLA 1516, and HLA Evolved.

If you use the MAK 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 *MAK RTI Reference Manual* for a list of the options for specifying the location of the *rid.mtl* file.

Applications built with VR-Link 5.6 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 5.6 has these new features:

- DIS Spy is now a licensed product included with VR-Link.
- VR-Link Code Generator now generates VC12 and VC14 project files.
- All features of the Code Generator can now be run from the command line.
- Added setAll(x, y, z) and get(x, y, z) functions to *DtVector* classes.
- Added several bit mask enumerations for appearance bits to *disEnums.h*.
- Added experimental support for IFF Mode5 and ModeS to HLA1516 Evolved. This support requires using RPR FOM Version 2.9, as well as adding two experimental FOM Modules: *Enumerations_Experimental_IFF.xml* and *RPR-MAK_Experimental_IFF.xml*. This pulls concepts from RPR 3 while making adjustments to ensure it is compatible with RPR 2.0.
- Added transition speed attribute to DIGuy Action records (DIS) and DIGuy ActionStruct datatypes (HLA).

Updated default HLA connections to: *MAK-RPR2-1-6.fed* (HLA1.3), *MAK-RPR2-1-6.xml* (HLA 1516) and *MAK-DIGuy-6_evolved.xml* (HLA 1516 Evolved).

- For tactical data link (TDL) support in VR-Forces, VR-Link has a new FOM file, *L16-RPR_FOM_v2.0_1516-2010.xml* base. This new FOM is the *RPR_FOM_v2.0_1516-2010.xml* FOM with added JTIDS enumerations, JTIDSTransmitterStruct, and JTIDS_MIDS_SpectrumType variant for SpreadSpectrumType in RadioTransmitter. There are no related changes to the VR-Link API.
- Added *L16-RPR_FOM_v2.0_1516-2010.xml* FOM and *MAK-BOML16-2_evolved.xml* FOM Module to add support for the SISO-STD-002-2006 Link 16 Simulation standard.

Fixed Bugs

This release fixes the following issues that were problems in previous releases:

- ♦ VRL-245. Fixed crash caused by encoder when enabling setSendDiGuyAttributes or setSendDiGuyPdus on an entityPublisher before setting any DIGuy attributes.
- ♦ VRL-246. Made isNullId function const.
- ♦ VRL-247. Updated setDiGuyName and setName functions for the DtEntityStateRepository and DtDIGuyObjectStateRepository to take cons references as arguments.
- ♦ VRL-250. Fixed VR-Link vlString.h header to compile with Visual Studio 2019.
- ♦ VRL-259. MAK-DIGuy-5_evolved.xml had a bad modification date.
- ♦ VRL-260. VC14 VR-Link will no longer throw an exception when connecting with VC15 or VC17 RTI libraries.
- ♦ VRL-268. TimeConstrained example now prints the correct location values.
- ♦ VRL-270. DI-Guy shape variations attribute is now sent correctly across the network.
- ♦ VRL-278. Corrected attribute name TimeSlotId to TimeSlotID in the *MAK-BOML16-1_evolved.xml* FOM Module. This now matches the *MAK-BOML16-2_evolved.xml* FOM Module and associated encoders/decoders.
- ♦ VRL-286. setHighDensity attribute now correctly works in Java.
- ♦ VRL-290. A Multicast TTL value can now be set through the command line when using the Java API.
- ♦ VRL-291. The Operational Status bit within the IFF PDU is now set correctly when done through the Java API.
- ♦ VRL-292. You can now set parameters 1-6 as capable within the IFF System Status record using the Java API.
- ♦ VRL-293. You can now set parameters 1-6 of the IFF Fundamental Operational Data record using the Java API.
- ♦ VRL-302. dropMulticastGroup function now works when specifying a device.
- ♦ VRL-303. Fixed compiler error with vlMutex.h when using Visual Studio 2019.
- ♦ VRL-305. Corrected usage of DtDistance function in f18 example.
- ♦ VRL-306. Fixed potential memory overrun in DtSignalInteraction class.
- ♦ VRL-308. vrlCodeGenerator can now be run from the command line with multiple FOM inputs.