VR-Link 3.11 Release Notes

This release note provides the following release-specific information for VR-Link Release 3.11:

Document ID: VRL-3.11-3-070320

Systems Supported

Table 1 lists the platforms currently supported by MÄK VR-Link 3.11. 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 for HLA and DIS

Platform	Compiler
SGI IRIX 6.5	MipsPro7.3 C++ compiler (available from SGI)
Sun Sparcstation with Solaris 2.8 or later	SPARCompiler C++ version 5.2 (available from Sun)
Red Hat Enterprise Linux Workstation 3.0	default compiler
Red Hat Fedora Core 3	default compiler
Red Hat Enterprise Linux Workstation 4.0	default compiler
PC with Windows 2000/XP	Microsoft Visual C++ 7.1 Microsoft Visual C++ 8.0

Table 2: Platforms supported for TENA

Platform	Compiler
Red Hat Fedora Core 3	default compiler
Red Hat Enterprise Linux Workstation 4.0	default compiler
PC with Windows 2000/XP	Microsoft Visual C++ 7.1 Microsoft Visual C++ 8.0

Disk Space Requirements

On SGI IRIX, you need approximately 348 MB for VR-Link files and 120 MB for all documentation files. Other UNIX 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.

Building VR-Link on UNIX

VR-Link does not require you to use the MÄK build system, but if you choose to, you must use gmake 3.81 or later. You can download it from the following URL:

http://savannah.gnu.org/projects/make/

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1 and 8.0. When you run MÄK products together, for example, the Logger and the Stealth, 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.

TENA Compatibility

VR-Link 3.11 was built against TENA 5.2.1. The MÄK LROM version is 2.1.

FLEXlm Support

VR-Link 3.11 uses FLEXlm 11.4 for Windows, Solaris, and Linux versions. The IRIX version continues to use FLEXlm 9.2.

Libraries and Binaries Built with Visual Studio 2005

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), 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. The installer is named *vcredist_x86.exe* and is in the base directory of any installed MÄK product.

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

Backward Compatibility

Though MÄK tries to maintain a high degree of source compatibility between VR-Link releases, changes to both header files and library names in VR-Link 3.10 mean that VR-Link customers upgrading to VR-Link 3.11 from release 3.9.6 or earlier may need to modify project files on Windows and makefiles in UNIX.

Network Compatibility

HLA only

VR-Link 3.11 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
- Pitch RTI 1.3 C++ interface.

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

DIS only

This release supports DIS 2.0.4, IEEE 1278.1, 2.1.4, and IEEE 1278.1a, and can therefore interoperate with DIS applications of any of these versions.

RPR FOM Versions Supported

VR-Link 3.11 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 3.11 uses RPR FOM 1.0.

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 3.11 has been tested with the Pitch RTI 1.3 and MÄK RTI 3.0.

VR-Link 3.11 may work with older versions of the MÄK RTI, but we recommend using it with release 3.0.

! VR-Link 3.11 for Sun Solaris is not compatible with versions of the MÄK RTI prior to version 2.4.

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 3.11 has the following new features:

- A new utility, the VR-Link Code Generator has been added
- The FOM Mapper Builder utility has been discontinued
- Upgraded TENA support to version 5.2.1
- Support for FLEXlm 11.4 (please see FLEXlm Support on page 3)
- Miscellaneous changes.

VR-Link Code Generator

The VR-Link Code Generator is a utility that generates HLA VR-Link extensions based on definitions provided in an OMT or XML definition file. The Code Generator creates all the files necessary to extend the VR-Link API to support any HLA FOM. The Code generator creates DtObjectPublisher, DtReflectedObject, and DtReflectedList classes, and all encoder and decoder functions required by VR-Link. The code generator also produces the Microsoft Visual C++ solution files and UNIX Makefiles required to build the extension library. You can link in the new extension library with the base VR-Link libraries to start using the custom, expanded API. For complete details, please see Chapter 9, Using the VR-Link Code Generator, in *VR-Link Developer's Guide*.

FOM Mapper Builder No Longer Supported

The FOM Mapper Builder utility is no longer supported and is not provided with this release. You can still build FOM Mappers manually as described in *VR-Link Developer's Guide*. As an alternative, you can use the VR-Link Code Generator to extend VR-Link to work with FOMs other than the RPR FOM.

Miscellaneous Changes

- In *tRefObjList.h*, the *DtReflectedObjectList* member functions addObjectAdditionCallback(), addObjectRemovalCallback(), removeObjectAdditionCallback(), removeObjectRemovalCallback() were made public. This is to make them consistent with the HLA and DIS reflected object list classes.
- The Launcher example has been updated to use the new exercise connection initialization class.
- *DtReferenceElipsoid* has been updated. geocentric->geodetic conversions now throw an exception if the conversion cannot be preformed in 5000 iterations.
- The DLL import/export macros for the TENA LROM Mappers were previously defined in the project files and Makefiles. These have been moved into *coLromMapExport.h* and *makLromMapExport.h*.
- The *DtEnvironmentTypeRecord* class was moved from the protocol-dependent vl libraries to the protocol-independent vl library.
- The WIN32 macro was being used in various places. This has been replaced with the proper _WIN32 macro, which is defined by the Microsoft compilers.
- Classes were added to support all UnderwaterAcousticEmission objects in RPR FOM 2 drafts 14 and 17. This includes the ActiveSonar, AdditionalPassiveActivities, and PropulsionNoise objects.
- ActionRequest and ActionResponse messages are now supported in the TENA version of VR-Link.
- You can turn off smoothing and dead-reckoning for entities by calling useSmoother(false) or useDeadReckoner(false) on the *DtBaseEntityStateRepository*. If useSmoother(false) is called on an entity, dead-reckoning is also turned off, since the smoother is also the dead-reckoner. To turn dead-reckoning back on, you must call useDeadReckoner() after you turn off smoothing.
- The *DtExerciseConn* can now report how many messages (object updates, object discoveries, object removals, and received interactions) were received during a drainInput(). To get this information, call lastDrainInputReceiveCount() on the exercise connection. This is reset each time drainInput() is called.
- The HLA DtExerciseConn::drainInput() member function now throws a DtException if an RTI exception is caught while ticking the RTI.
- You can now pass a "true" value into the *DtWindowsConsolePrinter* constructor (in *vlprint.h*) to disable the "Close" button of the Windows console.
- The assignment operators of the net types in *vlNetTypes.h* have the net type as the return value, not the local type.
- Each DIS object publisher now contains a member function called setDfltTimeThreshold(). You can pass in a time value, in seconds, to specify the default time threshold (time between DIS PDU heartbeats) for a specific class. This is useful if you want one default heartbeat interval for one type of PDU and another default heartbeat interval for another type of PDU, for example three seconds for entities and five seconds for emitter systems.

This behavior was inadvertently removed in version 3.9.4.

Documentation

The printed and PDF versions of *VR-Link Developer's Guide* have been updated for this release.

Bug Fixes

VR-Link 3.11 fixes the following bugs:

- The order of the initialization of members myPressAnyKey and _userSetOutput in *cmdLine.h* was causing GCC warnings.
- The call DtDecodingBuffer(0) was causing a crash with Windows VC++8.0.
- A debug assertion was being raised when changing the number of variable datums in *DtDataQueryInteraction* (on VC++8.0).
- TENA includes were added to *reflObjList.h* and *reflectedObj.h*. These protocol independant headers were missing the TENA includes, they only contained DIS and HLA.
- The *DtMtlEnvironment's* loadLispFile() method returned NULL if an error occurred in the first line of the file, and returned a non-NULL value if an error occurred later in the file. loadLispFile() now returns NULL if there is an error anywhere in the file, and the error and line number are printed out.

Known Problems

This release of VR-Link has the following known problems:

- 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 in *VR-Link Developer's 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.