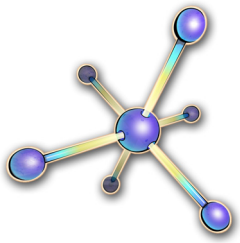

VR-Exchange 1.5 Release Notes

This release note provides release-specific information for VR-Exchange Release 1.5.

Document ID: VRE-1.5-2-090225

System Requirements

This section lists supported platforms and software requirements.

Platforms Supported

VR-Exchange 1.5 supports:

Table 1-1: Platforms supported

Broker	Platform	Compiler
HLA, DIS, TENA	Red Hat Enterprise Linux Workstation 4.0	Default compiler
HLA, DIS	Red Hat Enterprise Linux Workstation 5.0	Default compiler
HLA, DIS	Red Hat Fedora 7	Default compiler
HLA, DIS, TENA	PC with Windows XP	Microsoft Visual C++ 7.1, 8.0
	PC with Windows Vista	Microsoft Visual C++ 8.0

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

External Library Requirements

VR-Exchange 1.5 is built with VR-Link 3.12, pThreads 2.70, and Qt 4.4.2. 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.

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. 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) and later, 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 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.

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

! 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.

Patch Required for AMD Dual-processor Windows PCs

VR-Link-based products use a high resolution counter for time calculations on Windows PCs. Customers who are running Windows on PCs with multiple AMD Athlon 64-bit processors may notice clock jitter, which may cause time in MÄK products to run backwards. This occurs when the Windows scheduler changes the CPU the MÄK process is using. If the high resolution counters on each processor are not synchronized, the application may witness a decrease in the high resolution counter value stored in the processor causing an incorrect time calculation. To fix this problem customers, apply the AMD Dual-Core Optimizer patch provided by AMD. You can get the patch at:

http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_871_9706,00.html

---

! If you get an error when you try to access this URL, reload the page.

Brokers Provided

VR-Exchange 1.5 includes brokers for:

- HLA 1.3
- HLA 1516 (SISO Standard DLC C++ API for IEEE 1516)
- 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
Aggregate	X	X	X
Collision	X	X	X
Comment	X	X	X
Create Entity	X	X	
Data	X	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
IFF	X	X	X
Radio Receiver	X	X	X
Radio Signal	X	X	X
Radio Transmitter	X	X	X
Remove Entity	X	X	
Set Data	X	X	X
Start	X	X	X
Stop	X	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 1.5 has the following new features:

- Redesigned Portal with significantly improved ease-of-use features:
  - Ability to add and manage connections (broker configurations)
  - Ability to start and stop connections in the Portal
  - Updated display of stopped and crashed connections
  - Ability to display data about the VR-Exchange shared memory queue
  - Ability to load an alternative Portal configuration file
- Support for TENA 5.2.2
- New translators
- Improved performance:
  - Ability to span packets on multiple buckets
  - Improved performance loop
  - Improved performance of the polling interval
  - Updates from the Portal to HLA are updated immediately
- New translation features:
  - Support for jammer emitter beams
  - Fire and detonate interactions are linked in TENA
  - Ability to rename translated object names
  - Absolute timestamping is supported in HLA
- New configuration parameters:
  - Support for absolute timestamping in HLA
  - Ability to set multicast TTL and multicast Interface for TENA
  - Ability to specify heartbeat for HLA entities
  - Ability to configure persistent entity IDs.
- Configuration files are in XML format instead of MTL
- Connections do not require configuration files if they use default values

- New broker features:
  - Broker IDs are no longer required
  - Ability to override the DtExerciseConn.
- API improvements.

Redesigned Portal

The portal has been completely redesigned. Rather than simply being an informational display, it is now the focal point for configuring and managing VR-Exchange and brokers. This redesign makes VR-Exchange much easier to use. The Portal has the following features:

- You no longer need to use scripts to run brokers. (Support for scripts is maintained for backwards compatibility.) Each broker instance that you run is now configured as a connection. You add connections and specify their attributes in the Portal. You can configure connections to start when the Portal starts and you can enable and disable them from the Portal.
- Stopped brokers are now displayed as being unavailable in the GUI. If a broker has crashed, its connection is marked to signal the problem.
- The new Queue Status dialog box displays advanced internal information about VR-Exchange's status.
- You can configure VR-Exchange in the Portal. You can specify an alternative VR-Exchange configuration file at startup with the `-l` command-line option.
- The VR-Exchange and connection configuration files use XML instead of MTL.

New Translators

The DIS broker has the following new translators:

- Environment Process
- Emitter Systems
- Collision
- IFF
- Designator
- Event Report
- Create Entity
- Remove Entity.

The HLA broker has added:

- Designator
- Event Report
- Create Entity
- Remove Entity.

The TENA broker has added Environment Process and Set Data translators.

Improved Performance

VR-Exchange has the following performance improvements:

- VR-Exchange can now span packets on multiple buckets. The default bucket size is eight times smaller than previous releases. To reduce packet dropping, the default bucket count is five times larger.
- Setting **pollingInterval** to a high number (1 second or more) now improves performance, and does not hang the entire broker.
- All brokers now have an improved performance loop. In particular, heavy loads will no longer hang the GUI.

New Translation Features

VR-Exchange has the following new translation features:

- Jammer type emitter beams are now supported in HLA, DIS, and TENA. All other emitter beams are already supported.
- Fire and Detonate interactions are linked in TENA so that detonations have the origin location filled in.
- You can now define your own translated object names in a file (*hlaEntityNameMap.csv* and *tenaEntityNameMap.csv*). These files are lists of name pairs, in the format *oldName*,*newName*. If Use Marking Text as Object ID is enabled, *oldName* should be the marking text data. The published object will change the marking text as well as the object name.

New Configuration Parameters

VR-Exchange has the following new configuration parameters:

- Absolute Timestamping in HLA is now supported.
- You can now configure the multicast TTL and multicast Interface TENA values. To enable multicast TTL, set the multicast TTL attribute to 1. To enable the multicast interface, specify an address in the Multicast Interface attribute.
- You can now configure a heartbeat for HLA entities. To do this, give the Heartbeat attribute to a value greater than 0.
- You can now specify that entity IDs persist after disconnects. If this feature is enabled, when you replay an exercise, entities with common names will use the same entity IDs that they did in previous runs. To enable this feature, set Use Persistent Entity IDs to 1. The default is 0. You must also enable Fill Outgoing Entity IDs. This attribute saves entity IDs to the file *hlaPersistentEntityNumMap.csv*.

New Broker Features

Broker executables have the following new features:

- Broker IDs are no longer required. Broker names are the only thing that needs to be unique. Broker IDs still exist, but they are now assigned internally at runtime without any user input. The `--BrokerID` command line option and the **brokerID** parameter have been removed.
- You can disable translators at run time without disconnecting a broker.
- You can enable and disable a translator's debug options in real time.
- The HLA Broker's tie to a generic *DtExerciseConn* has loosened. Now custom brokers can override the *DtExerciseConn* if necessary.

API Improvements

Every DIS, HLA, and TENA translator now requires the enableTranslator() member function:

```
virtual void enableTranslator(bool enabled)
```

This function should connect your callback functions to and from the network and the portal. It provides the ability to enable and disable a translator in real time. When a translator is created, it should start disabled. (Previous versions had translators start enabled and never change.)

Here is an example of using enableTranslator():

```
void DtEntityTranslator::enableTranslator(bool enabled)
{
  if(enabled)
  {
    myHlaReflectedList->addEntityAdditionCallback(
      DtEntityTranslator::hlaObjectAddedCb, this );
    myPortalReflectedManager->addAdditionCallback(
      DtEntityTranslator::portalObjectAddedCb, this );
  }
  else
  {
    myHlaReflectedList->removeEntityAdditionCallback(
      DtEntityTranslator::hlaObjectAddedCb, this );
    myPortalReflectedManager->removeAdditionCallback(
      DtEntityTranslator::portalObjectAddedCb, this );
  }
}
```

Documentation Updates

The print and PDF versions of *VR-Exchange Users Guide* and the online help have been updated.

Bug Fixes

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

- Broker GUIs used to start in the background when selected the first time under Windows. They now correctly start in the foreground.
- VR-Exchange crashed if a user started a broker with the same name or a second copy of the Portal connected to the same queues.
- The useMarkingTextForTenaObjectId option did not work in TENA.
- VR-Exchange Portal icons for destroyed are no longer being incorrectly shown when an entity is damaged.
- Disabled translators no longer send messages.
- The `--version` and `--help` command-line options did not work.

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* dirrectory. 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.