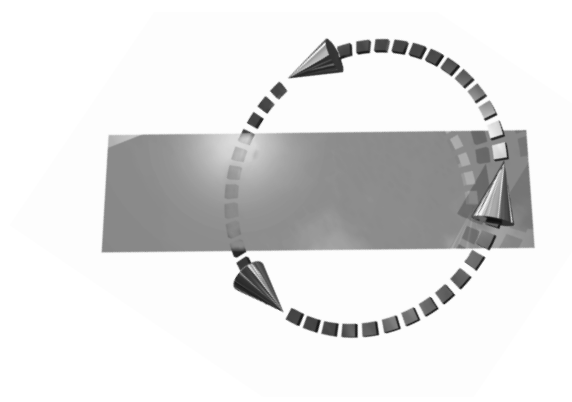




MÄK VR-Link 3.9.2 Release Notes

This release note provides the following release-specific information for MÄK VR-Link Release 3.9.2:

Systems Supported	2
Disk Space Requirements	2
Backward Compatibility	3
Network Compatibility	3
RPR FOM Versions Supported	3
RTI Support	4
New Features and Changes	5
Backwards Incompatibility	5
Change to vlPrint.h	6
Copy Constructors and Assignment Operators	6
Support for IFF Layer 2 in Protocol Independent API	6
Protocol Independent Header Files	6
Changes to Decoder Classes	7
DtDecodingBufferMgr Replaced	7
Changes to the DtUtmInit() Function	7
Documentation Updates	8
Bug Fixes	8
Known Problems	8
UTM Conversion Round-off Errors	9

Systems Supported

Table 1 lists the platforms currently supported by MÄK VR-Link 3.9.2. 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

Platform	Compiler
SGI IRIX6.5	MipsPro7.3 C++ compiler (available from SGI)
Sun Sparcstation with Solaris2.7 or later	SPARCompiler C++ version 5.2 (available from Sun)
PC with Red Hat Linux 8.0 and 9.0	GNU C++ (GCC) 3.2
PC with Windows 2000/XP	Microsoft Visual C++ 6.0 SP5 Microsoft .NET

You must use *libXml2* to build any HLA application.

Disk Space Requirements

On SGI IRIX, you need approximately 270 MB for VR-Link files and 60 MB for all documentation files. Other UNIX installations require approximately 50 MB for VR-Link files and an additional 60 MB for documentation files.

On PCs, you need 50 MB for VR-Link files and 60 MB for all documentation files.

Patching the Microsoft Visual C++ 6.0 Compiler

If you use Microsoft Visual C++ version 6.0, you must install a patch to a standard header file, in order to successfully build applications against the IEEE 1516 header files. The patch comes from Dinkumware, Ltd (the company that developed the Standard C++ Library header files for Microsoft), but for convenience, we are distributing it with VR-Link releases.

The patch fixes a bug that would cause applications to crash if they pass `std::sets` or `std::maps` from an application to a DLL or vice-versa, something that IEEE 1516 federates must do. For details about the patch go to http://www.dinkumware.com/vc_fixes.html.

To install the patch:

1. In the top level VR-Link directory, is a file called *XTREE-msvc++patch*. Rename this file *XTREE*.
2. Make a back-up version of the original *XTREE*.

3. In the Visual Studio *include* directory (for example, *C:\Program Files\Microsoft Visual Studio\VC98\Include*), replace the original version of *XTREE* with the patched version.

Backward Compatibility

VR-Link 3.9.2 is highly compatible with VR-Link 3.9.1. For example, the *talk* and *listen* example programs from VR-Link 3.9.1 and previous versions compile and link unmodified against VR-Link 3.9.2.

Network Compatibility

HLA only

VR-Link 3.9.2 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 2.1
- DMSO RTI NG 4.0, and 6.0 (but not 5.0). The Linux version is not compatible with DMSO RTI NG.

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 3.9.2 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 3.9.2 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-Link20006", "MyApp", new DtRprFomMapper(2.0006));
```

VR-Link examples like *fl8* and *hlaNetdump* have a command line option, `-f`, that you can use to choose an RPR FOM version by passing a RPR FOM keyword as the argument to `-f`. The keywords for specifying different versions of the RPR FOM with the `-f` startup option are:

- ♦ RPR FOM 0.5: `v1RPR05`
- ♦ RPR FOM 0.7: `v1RPR07`
- ♦ RPR FOM 0.8: `v1RPR08`
- ♦ RPR FOM 1.0: `v1RPR10`
- ♦ RPR FOM 2.0 draft 6: `v1RPR20006`
- ♦ RPR FOM 2.0 draft 14: `v1RPR20014`
- ♦ RPR FOM 2.0 draft 17: `v1RPR20017`.

RTI Support

VR-Link 3.9.2 has been tested with DMSO RTI NG v6, Pitch RTI 1.3, and MÄK RTI 2.2.

VR-Link 3.9.2 may work with older versions of the MÄK RTI, but we recommend using it with release 2.2.

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

This section describes the following features and changes to VR-Link:

- ♦ Backwards incompatibility for subclasses of *DtBaseHashKey*
- ♦ Change to *vlPrint.h*
- ♦ Copy constructors and assignment operators no longer implemented for non-assignable, copyable classes
- ♦ Backwards incompatibility for the Minefield State PDU
- ♦ Support for IFF Layer 2 added to protocol independent API
- ♦ New protocol independent header files
- ♦ Changes to *DtInteractionDecoder* and *DtHlaStateDecoder*
- ♦ *DtDecodingBufferMgr* replaced
- ♦ Changes to *DtUtmInit()*.

Backwards Incompatibility

The following changes have introduced backwards incompatibility:

- ♦ Previously *DtBaseHashKey* (*BaseHashKey.h*) had the member function:

```
virtual Const Dtstring &stringrep() Const = 0;
```

This caused people who subclassed it to maintain a copy of *DtString* to which that reference referred. This used a considerable amount of memory for no advantage. Often subclasses would compute the string even though it was never used.

VR-Link now defines this function with return by value:

```
virtual Const Dtstring Stringrep() Const = 0;
```

This means anyone who subclasses *DtBaseHashKey* must return a *DtString* by value too. This does not affect callers of the member function.

- ♦ The bug fix for the Minefield State PDU (see [“Bug Fixes,”](#) on page 1-8) creates an incompatibility with previous versions of VR-Link, because it changed the PDU layout.

Change to *vlPrint.h*

The global *DtWindowsConsolePrinter* instance “DtStandardWindowsConsole” has been removed from *vlPrint.h*. This caused a console window to be created for Windows applications even though it was not needed. If you want to create a console window and attach it to a stream, do the following:

```
DtPrinter *console = new DtWindowsConsolePrinter;

DtWarn.attachPrinter(console);
DtInfo.attachPrinter(console);
DtVerbose.attachPrinter(console);
DtFatal.attachPrinter(console);
DtDebug.attachPrinter(console);
```

Copy Constructors and Assignment Operators

Many classes in VR-Link are designed to not be copyable or assignable. This is typically represented with a protected copy constructor and assignment operator. Previously these member functions were implemented, but if they were used there would be a runtime error with a warning.

All the instances in VR-Link of protected copy constructors and assignment operators have been removed. This will cause an error at link time, if the member functions are referenced. There will no longer be runtime errors.

Support for IFF Layer 2 in Protocol Independent API

The Protocol Independent API now supports IFF Layer 2. Previously, only the DIS API supported it.

Protocol Independent Header Files

Many protocol-independent header files have been added.

Changes to Decoder Classes

The bug fix for crashing on Sun Solaris and Linux (see “[Bug Fixes](#),” on page 1-8) affected *DtInteractionDecoder* and *DtHlaStateDecoder*. For both of these classes:

- ♦ The static function `decodingBufferSize()` was removed.
- ♦ The static function `decodingBuffer(int length)` now has no default length value. You must specify a length value. If the function cannot provide a length of that size, the function will return a null pointer. The function may return a buffer which is larger than the specified size. In all other regards this function behaves the same as it previously did.



Previously, a call to `decodingBuffer(0)` allocated a buffer of size 20. This was an error.

DtDecodingBufferMgr Replaced

The class *DtDecodingBufferMgr* has been removed. It has been replaced by a global function called `DtDecodingBuffer(int length)` which returns a pointer to a memory pool of at least size `length`. If a pool of that size can not be allocated it returns 0. It is thread safe, different buffers exist for different threads.

Changes to the `DtUtmInit()` Function

The function used to initialize VR-Link’s global UTM Reference Point has changed. Previously `DtUtmInit()` (*utmCoord.h*) had a parameter named **verbose**, which did nothing.

```
void DtUtmInit(const DtDegMinSec& latRef, const DtDegMinSec& lonRef,
               int standardOffsets=1, int verbose=1 );
```

The **verbose** parameter has been replaced with a parameter to set the UTM reference Zone. A value of 0 allows VR-Link to calculate the reference zone from the given latitude/longitude. However, some database systems contain latitude/longitude reference points or zones that do not match.

```
void DtUtmInit(const DtDegMinSec& latRef, const DtDegMinSec& lonRef,
               int standardOffsets=1, long zone = 0 );
```

Check your existing code to be sure it does not set the **verbose** parameter and change it if necessary.

The class *DtUtmReferencePoint* (*utmRefPt.h*) has also changed. The constructor and `init()` method now take a zone. The zone defaults to 0, which means the class should compute the zone based on the reference points specified.

Documentation Updates

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

Bug Fixes

VR-Link 3.9.2 fixes the following bugs:

- The fundamental data parameter in IFF Layer2Data is now decoded correctly in HLA.
- Incorrect coding of the MinefieldProtocolMode attribute in the Minefield State PDU has been corrected.
- VR-Link applications running on Sun Solaris or Linux no longer crash when you are using a FOM mapper and a federate exits.
- Calls to *DtOutputStream* for which the threshold for printing was not met were very slow, even though no printing occurred. For example, if *DtNotifyLevel=DtNLWarn* a call to *DtInfo << "Hi";* is a NO-OP, yet considerable time was spent in *DtInfo*. This has been considerably improved.
- For the IFF State Repository, calls to *setModeCode()* required three decimals (which were in octal) and did not work. *setModeCode()* now takes the exact bit layout as specified by the DIS 1278 standard. The state repository does no conversion.
- In HLA, when an emitter system received two beam objects with the same beam ID, when the first object was removed, VR-Link crashed. This has been fixed.

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*.

UTM Conversion Round-off Errors

The class *DtUtmReferencePoint* (*utmRefPt.h*) exhibits inconsistent coordinate conversion behavior when the reference points are on a zone boundary and the zone is computed by VR-Link.

This problem is caused by a floating point round off error that incorrectly computes the zone. It is evident when computed values are compared between different operating systems and machines. (Specifically positions computed on Windows XP do not match positions computed on Windows 2000.)

You can work around this problem with either of the following methods:

- ♦ Specify the zone explicitly. This is now possible in VR-Link 3.9.2.
- ♦ If you do not know the zone, you can set a *DtUtmReferencePoint* static variable with *setLongitudeFudgeFactor(double)*. This value should be very small, 1e-6 should suffice. This value will be added to the reference longitude to prevent the round-off error.

