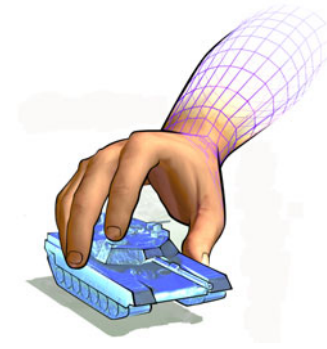




VR-Forces 3.10 Release Notes

This release note contains the following release-specific information for VR-Forces Release 3.10:

Systems Supported	3
Libraries and Binaries Built with Microsoft Visual Studio 2005	3
Building on Linux	4
Disk Space Requirements	4
Compiler Compatibility on Windows	4
MÄK Product Compatibility	4
Qt Release Compatibility	5
FLEXlm Support	5
Third-Party Library Requirements	5
Network Compatibility	5
Backwards Compatibility	6
RPR FOM Versions Supported	6
New Features and Updates	7
Changes to the VR-Forces Application	8
Global Plans	8
Spot Reports	8
Time Management	8
Load Balancing	8
Support for VMAP Files	8
Updated Raster Map Layers Dialog Boxes	9
Displaying Additional Object Information	9
Change to How Independent Tasks Affect Plans	9

Copyright © 2006 MÄK Technologies, 68 Moulton St., Cambridge, MA 02138
All rights reserved. MÄK Technologies®, VR-Forces®, RTIsPy®, and VR-Link® are registered trademarks of MÄK Technologies, Inc.
Document ID: VRF-3.10-3-061213

Miscellaneous Changes	9
Redesigned MÄK Terrain Database Tool.....	11
B-HAVE Module	12
API Changes	12
New Plug-In API Architecture	12
Changes to the Rotary Wing Model	12
Changes to Registration for Task and Set Data Request Messages	14
Simulation API Changes.....	15
GUI API Changes	15
Documentation Updates.....	16
Bug Fixes	17
Known Problems	19

Systems Supported

VR-Forces 3.10 is available for the platforms listed in [Table 1](#). For the availability of other platforms, contact your MÄK salesperson. For toolkit users, application code must be built with the indicated compilers in order to link to VR-Forces libraries.

Table 1: Platforms supported

Operating System	Compiler	glibc
Red Hat Enterprise Linux Workstation 3	GNU C++ (GCC) 3.2.3	2.3.2
Red Hat Enterprise Linux Workstation 4	GNU C++ (GCC) 3.4.4	2.3.4
Red Hat Fedora Core 3	GNU C++ (GCC) 3.4.4	2.3.6
Windows 2000 SP2, XP	Microsoft Visual C++ 7.1	
	Microsoft Visual C++ 8.0	

Applications running on newer AMD CPUs, without the latest CPU drivers from the manufacturer, may experience performance problems, regardless of whether those applications are provided by MÄK Technologies. It is highly recommended that the latest versions of the drivers be installed. The drivers are available from AMD at the following URL:

http://www.amd.com/us-en/Processors/TechnicalResources/0,,30_182_871,00.html

Libraries and Binaries Built with Microsoft 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 *vc8dist_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>

Building on Linux

The example Makefiles require a patched version of gmake 3.80, which has been included for your convenience in the *./mkbin* directory. This version of gmake is based on version 3.80, but includes a memory fix that is needed to use our makefiles. The source code for the modified gmake is freely available from <http://ftp.mak.com/out/gmake3.80-patch1.tar.gz>. This patch will be included in future versions of gmake.

Before you compile the examples, go to the top level of your installation and create a symbolic link 'VR-Link' to your VR-Link installation and another called 'RTI' to your RTI installation.

Disk Space Requirements

A full installation of VR-Forces requires approximately 600 MB of disk space. Terrain databases use approximately 110 MB and documentation (primarily the class reference) uses 200 MB.

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 on the same computer, for example, VR-Forces 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.

MÄK Product Compatibility

To build VR-Forces applications, you must link with VR-Link 3.10.1. If you are building for HLA and want to link with the MÄK RTI, use MÄK RTI 3.x.

VR-Forces 3.10 is a parallel release to MÄK Plan View Display 2.10.

Qt Release Compatibility

If you want to do development using the VR-Forces GUI API, you need to use Qt, a cross-platform GUI toolkit from Trolltech. The VR-Forces GUI was built with Qt release 3.3.5. A VR-Forces developer's license includes a Qt development license. MÄK provides you with a Qt license key. However, you must download the correct version of Qt from www.trolltech.com.

FLEXlm Support

VR-Forces 3.10 uses FLEXlm 10.8. You must copy the files in *./flexlm10.8* to your license server directory. You do not need to change license files (unless your maintenance agreement has expired.)

Third-Party Library Requirements

DtTerrainProfileWidget is based, in part, on the work of the Qwt project (<http://qwt.sf.net>). To compile examples or user projects, you must download the QWT distribution at <ftp://ftp.mak.com/out/qwt-0.4.1.zip>. This distribution is known to work with VR-Forces. Set the MAK_QWTDIR environment variable to point to the location of the QWT library.

Network Compatibility

HLA only

VR-Forces 3.10 was built against VR-Link 3.10.1 and is compliant with:

- RPR-FOM 1.0 and a subset of 2.0 (draft 6 and 17)
- MÄK RTI 3.x.x and other RTIs that meet the HLA 1.3 or IEEE 1516 SISO DLC API specifications and are built with the same compilers as VR-Forces.

VR-Forces 3.10 is compatible with applications that use earlier versions of VR-Link if they support versions of the RPR FOM listed.

i

VR-Forces is no longer compatible with the DMSO RTI-NG v6, because that RTI was not built using any of the compilers supported by VR-Forces 3.10. VR-Forces 3.10 should be compatible with commercial successors to the DMSO RTI if they conform to the HLA 1.3 or 1516 (SISO DLC API) specifications and are built with a compiler supported by VR-Forces.

DIS only

VR-Forces 3.10 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.

Backwards Compatibility

The VR-Forces 3.10 APIs are not compatible with previous releases of VR-Forces. Scenarios and object parameter databases from previous versions are forward compatible.

RPR FOM Versions Supported

VR-Forces 3.10 has built-in support for versions 1.0 and 2.0 (draft 6 and 17) of the RPR FOM. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, VR-Forces 3.10 uses RPR FOM 1.0.

If you are building an application with the VR-Forces toolkit and you want to specify a version of the RPR FOM through code, pass the version number (for example, 2.0006) 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));
```

In order to support RPR FOM 1.0, we have added the following extensions (which are supported in later versions of the RPR FOM):

- ♦ EnvironmentProcess
- ♦ GriddedData
- ♦ EmbeddedSystem.IFF
- ♦ EmbeddedSystem.IFF.NatoIFF
- ♦ EmbeddedSystem.IFF.NatoIFF.NatoIFFTransponder
- ♦ EmbeddedSystem.IFF.NatoIFF.NatoIFFInterrogator
- ♦ EmbeddedSystem.IFF.SovietIFF
- ♦ EmbeddedSystem.IFF.SovietIFF.SovietIFFTransponder
- ♦ EmbeddedSystem.IFF.SovietIFF.SovietIFFInterrogator
- ♦ EmbeddedSystem.IFF.RRB
- ♦ BaseEntity.AggregateEntity
- ♦ ObjectRoot.BaseEntity.PhysicalEntity.Lifeform.

For both RPR FOM 1.0 and 2.0 VR-Forces 3.10 relies on the LgrControl and View-Control custom MÄK extensions.

New Features and Updates

VR-Forces 3.10 has the following new application features and updates:

- ♦ Redesigned and updated Terrain Database Tool, including introduction of projects and replacement of MAP files with MAK Terrain Descriptor (MTD) files. The TDB Tool is documented in *TDB Tool User's Guide*.
- ♦ B-HAVE module
- ♦ Global Plans
- ♦ Spot Reports
- ♦ Support for HLA time management
- ♦ Load balancing when running multiple back-ends.
- ♦ Support for loading multiple object parameter databases in a scenario
- ♦ The Raster Map Layers dialog box has been updated
- ♦ Additional information is displayed in object information dialog boxes, including entity consoles
- ♦ Support for VMAP files
- ♦ New procedure for creating aggregate entities
- ♦ Support for recently opened scenario and terrain database lists
- ♦ Rewind pauses back-end option
- ♦ Context sensitive task and set data request menus
- ♦ Support for displaying a watermark on the map
- ♦ Change to how independent tasks affect plans.

VR-Forces 3.10 has the following new API features and updates:

- ♦ New Plug-in API
- ♦ Improvements to the Task Manager
- ♦ Rotary wing model reorganization
- ♦ Terrain MAP API
- ♦ Terrain Drawing API

Changes to the VR-Forces Application

This section describes new features and changes that primarily affect the VR-Forces GUI and end-users, as opposed to developers.

Global Plans

Global plans are plans that are not associated with a particular entity. From a global plan, you can create and delete all objects available on the Create menu. You can assign tasks and set data requests to entities, and can attach to and detach from the MÄK Stealth. Conditional logic is available in global plans. All of the commands available in global plans are now also available in individual entity plans. For details, please see Chapter 13, *Writing Plans*, in *VR-Forces User's Guide*.

Spot Reports

When an entity detects another entity, it can send a report of the contact. This is called a spot report. VR-Forces entities can now send spot reports. You can configure the VR-Forces GUI to hide the map display of all members of a force except those that have been reported via spot reports. You can also make the display of graphical objects subject to spot reports. This provides a simple fog-of-war implementation. All entities are still visible in the object display lists. For details about spot reports, please see Chapter 8, *Creating and Managing Entities*, in *VR-Forces User's Guide*.

Time Management

VR-Forces now supports time management for HLA federations. For details, please see Chapter 3, *VR-Forces Concepts* and Chapter 4, *Starting VR-Forces*, in *VR-Forces User's Guide*.

Load Balancing

VR-Forces now supports automatic and manual load balancing. This feature is distinct from remapping, which is invoked only if a back-end is missing. If you create a scenario and decide that you want to run it using more back-ends than you used to create it, you can redistribute the objects in the scenario to additional back-ends. For details, see Chapter 5, *Creating and Running Scenarios* in *VR-Forces User's Guide*.

Support for VMAP Files

You can now load vector data from the VMAP standard into VR-Forces. You can import based on a region of interest (defined by lat/long) and by a thematic layer as well as an optional specific element within that thematic layer. For details, please see Chapter 6, *Metafile Records* in *MÄK Terrain Database Tool User's Guide*.

Updated Raster Map Layers Dialog Boxes

The Raster Map Layers dialog box has the following changes:

- ♦ You can change the transparency value using the up and down arrow keys
- ♦ When you select a layer, the dialog box displays a thumbnail version of the raster image.

Displaying Additional Object Information

Object information dialog boxes have new tabs, as follows:

- ♦ Air platform objects that have IFF enabled display a tab showing IFF information
- ♦ Entities have a console tab that shows the task they are executing, console messages sent from plans, and other status messages, depending on the information level selected. For details, please see Chapter 9, *Displaying Entity Information* and Chapter 13, *Writing Plans*, in *VR-Forces User's Guide*.

Developers can use the entity console window to display their own diagnostic messages. The *DtVrfObject* class provides accessors to a set of iostream objects, used to send text messages to the entity's information window in the GUI. For example, to send a DtWarn-level message to an entity's console window:

```
objectConsoleWarn() << "Cannot complete task." << std::endl;
```

The available streams are analogous to the *DtOutputStream* classes in VR-Link (*DtWarn*, *DtInfo*, and so on.) See *vrfObject.h* for the list of available console output streams.

Change to How Independent Tasks Affect Plans

In previous releases of VR-Forces, when you gave an entity an independent task, if the entity had a plan, it stopped working on the current task, completed the independent task, then went to the next task in the plan. If a plan had a trigger, firing of the trigger would interrupt the independent task.

Now, when you issue an independent task, the entity abandons its plan and executes the independent task. After completing the independent task, the entity waits for further tasks to be assigned. As noted in *VR-Forces User's Guide*, when you issue a global command to an entity, it has the same effect as issuing an independent task.

Miscellaneous Changes

- ♦ You can configure VR-Forces so that when you rewind a scenario it automatically pauses, rather than immediately restarting.
- ♦ The File menu now lists the most recently opened scenarios and terrain databases so that you can quickly select them.
- ♦ You can now attach objects to entities after you create them and detach them from entities. Previously you could only attach an object to an entity when you created it and could not detach it.

- The task and set data request menus are now context sensitive. They only list the tasks and sets that are appropriate to the selected entity.
- You can now associate more than one OPD file with a scenario. This allows you to include custom objects in a scenario, as well as keeping all the default VRF objects without having to directly modify the default VR-Forces OPD file. All VRF examples that include customized OPD files now use this mechanism to specify their customizations. For details, please see [Section 5.1.1, "Specifying Multiple Object Parameter Database Files"](#) in *VR-Forces User's Guide*.
- You can enable and disable display of a splash screen on startup. To configure this option, edit the `showSplashScreen` parameter in *pvd.mtl*.
- When you edit graphical objects you can move them a small amount (nudge them) from the keyboard. To nudge a graphics, hold down the **Cntrl** key while pressing the arrow or number keys normally used for movement.
- You can preserve empty overlay layers as part of scenario. To enable this option, set the `preserveEmptyOverlayLayers` to 1 in *pvd.mtl*.
- You can now specify the range of DIS application IDs VR-Forces uses to identify an application as a VR-Forces back-end. By default, the range is 3000-4000. You can change this range by editing the `appldRange` parameter in *vrfSim.mtl*.
- You can now specify that *vrfSim* load an alternate *physicalWorldParams.mtl* file in the following ways:
 - Edit the `physicalWorldFile` parameter in *vrfSim.mtl*
 - Start *vrfSim* with the `--physicalWorldFile` command line argument
 - Specify a file in the Scenario Settings panel in the Choose Terrain dialog box when you create a new scenario or edit the **Physical-World** parameter in a scenario file.
- You can now edit plans in place, with the following restrictions:
 - You cannot change the operator in a resource or simulation time conditional expression. However, you can change the values of parameters in the expression.
 - You cannot edit compound conditionals (AND, OR, NOT).
 - You cannot change the True/False value of a conditional expression.
- The alignment angle between the ballistic gun turret and the target is now configurable in the OPE file through the parameter `max-alignment-angle`.
- The aggregate-as operation is now more customizable in the GUI. You can configure the composition and type of aggregate to create in the Aggregate As dialog box.
- A Leo2A4 tank has been added to the Create Entity menu.
- The initial sector of responsibility for an entity type can now be set in the OPE file through the parameter `initial-sector-of-responsibility`.
- A new command line option, `--outputLogFile`, lets you specify the name of the log file to write console output to. This option is available for both the GUI and the back-end.

- ♦ The plan editor has a new conditional expression for testing altitude.
- ♦ Scenario files are now saved and loaded relative to the *.scn* file, making scenario files more portable. The relative path rule applies to the following files in a scenario file:
 - plan
 - global plan
 - object map
 - order of battle
 - overlay
 - selection group.Other filenames in a scenario file (terrain database, parameter database, additional OPD files) continue to be referenced relative to the *.bin** directory.
- ♦ You can now draw contour lines on the mini-map.
- ♦ A new parameter, **rounds-per-trigger-pull**, the ballistic gun controller component lets you configure the quantity field in fire and detonate interactions.
- ♦ You can now attach and detach control objects to entities through the right-click menu.

Redesigned MÄK Terrain Database Tool

The MÄK Terrain Database Tool (TDB Tool) has been enhanced to add support for the new MÄK Terrain Project and MÄK Terrain Descriptor formats. MÄK Terrain Descriptor (MTD) files replace MAP files and serve the same function, specifying a terrain database and associated raster map images to load into VR-Forces. MÄK Terrain Projects allow you to create GDB terrain databases from supported database and vector input files. Projects can be saved so that you can repeatedly reproduce a database with a known set of parameters, as needed. The TDB Tool GUI now supports specification of the various import parameters that were previously only available from the command line or in MAP files.

The TDB Tool now allows you to add point, linear, and areal vector features to a GDB database.

TDB Tool documentation has been removed from *VR-Forces Configuration Guide* and is now provided in *MÄK Terrain Database Users Guide*. Please refer to this manual for complete details about how to use the TDB Tool.

B-HAVE Module

Shortly after the release of VR-Forces 3.10, MÄK will release a VR-Forces add-on module called B-HAVE (Brains for Human Activities in Virtual Environments.) The B-HAVE Module for VR-Forces leverages advanced Artificial Intelligence (AI) technology to provide more complex and realistic behaviors within the VR-Forces simulation environment.

For more information about this separately-priced VR-Forces option, please choose Help → About B-HAVE in the VR-Forces GUI, or go to <http://www.mak.com/products/B-HAVE.php>.

API Changes

New Plug-In API Architecture

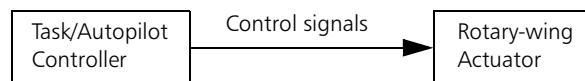
Developers can now create shared libraries that can be loaded at runtime so that you can add functionality to VR-Forces without recompiling the entire application.

For complete details, please see Chapter 1, *Introduction to the VR-Forces APIs*, Chapter 2, *The VR-Forces Simulation Engine*, and Chapter 11, *The VR-Forces GUI API* in *VR-Forces Developer's Guide*.

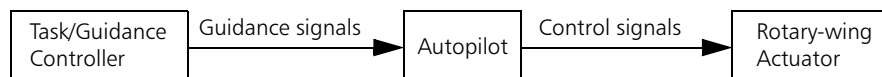
Changes to the Rotary Wing Model

The rotary-wing model has been refactored to make it easier to extend. Rotary-wing entity behaviors are unaffected by the changes. The API changes encompass all of the controllers, actuator, state repository and parameters associated with the rotary-wing model. The core concept behind these changes are the introduction of a separate autopilot controller, inserted between the task controllers and the rotary-wing actuator.

Previously, task controllers were directly connected to the rotary wing actuator as follows:



Now, all rotary-wing task controllers connect to a single autopilot controller, connected to the rotary-wing actuator.



Please see the following header files for details:

- ♦ Task controllers:
 - *rotWingJoyCtl.h*
 - *rotWingLandingController.h*
 - *rotWingTaskCtl.h*
 - *rotWingTurnToHeadingCtl.h*
 - *rotWngColAvod.h*
 - *rotWngEmbarkationController.h*
 - *rotWngFolwCtl.h*
 - *rotWngLoitCtl.h*
 - *rotWngMvAlg.h*
 - *rotWngMvToCtl.h*
 - *rotWngMvToLocCtl.h*
 - *rotWngPtlBtwn.h*
 - *rotWngPtrlRt.h*
 - *rotWngWaitCtl.h*
- ♦ Autopilot:
 - *rotWngPltCtl.h*
 - *simpRotWngPlt.h*
- ♦ Actuator: *rotWingAct.h*
- ♦ State Repository: *rotWingSR.h*
- ♦ Parameters: *rotWingPar.h*
- ♦ Component Descriptors:
 - *rotWngPltDesc.h*
 - *rotWngColAvDsc.h*
 - *rotWingRtTaskDesc.h*
 - *rotWingTaskDesc.h*
 - *rotWngActDesc.h*
 - *rotWngJoyDsc.h*

Changes to Registration for Task and Set Data Request Messages

VR-Forces 3.10 introduces some new functionality into the task manager, and a new set data manager that require some changes to the way that controllers register for task and set data request messages.

Task Messages

Any controller that registers for a task message and carries out a task must be changed to the new scheme. The change is as follows:

Old code:

```
myRadio->addRadioMessageCallback(DtRetreatTaskType,  
    retreatTaskCallback, (void *) this);
```

New code:

```
addTaskProcessorCallback(DtRetreatTaskType, retreatTaskCallback,  
    (void *) this);
```

The controller no longer calls `startTask()` when it begins processing a new task. The task manager already knows a new task is starting, and has cleaned up any previous task before the registered callback is called.

See the `addTaskSim` example for an example of a controller processing a task.

Set Data Request Messages

Any controllers that registers for a set data request message and carries out the set request must be changed to the new scheme. There is a new *DtSetDataManager* associated with each entity, which centralizes the distribution of set data requests. Code changes should be made as follows:

Old code:

```
radio->addRadioMessageCallback(DtSetHeadingRequestType,  
    setHeadingCallback, (void *) this);
```

New code:

```
entity()->setDataManager()-  
    >addSetDataProcessorCallback(DtSetHeadingRequestType,  
    setHeadingCallback, (void *) this);
```

Simulation API Changes

The Simulation API has the following additional changes:

- The deprecated class *DtTargetDetectionSensor* is scheduled to be removed from the API immediately following this release. See section 5.11 "Sensors" in *VR-Forces Developer's Guide* for a discussion of the Signature Sensor system that functionally replaces this component.
- The load-ammo-time and unload-ammo-time parameters are no longer saved to the order of battle file. They are now strictly read in from the entity's parameter database entry. As a result, the actual times used in scenarios may be different than in previous versions. Consult the OPE files to see the actual values used.

GUI API Changes

The GUI API has undergone substantial changes, designed to make common extensions to the application easier to make. Among the areas of API affected by the changes:

- Capturing and Displaying State Updates
- Creating and Modifying Symbols
- Creating and Modifying Toolbars and Menus
- Drawing the Terrain Map
- Event Signaller
- Event Controllers
- Event Handlers
- Managing Overlays

For details, please see Chapter 11, *The VR-Forces GUI API*, in *VR-Forces Developer's Guide*.

The GUI API has the following additional changes:

- The data member *DtTMApplication::myModelDataDict()* (*tmApplication.h*) has changed from a stack variable to a pointer.
- The signature of the member function *DtTMWindow::init()* (*tmWindow.h*) has changed. If you have overridden this member function, be sure to change the signature in your derived version, to ensure it gets called.
- If you have created subclasses for any of the following VR-Link state repository classes in the back-end, you must change the class you inherit from as shown in [Table 2](#) (declared in *vrfStateReps.h* and *controlObjSR.h*):

Table 2: State repository classes to change

VR-Link state repository	Replace with extended VR-Forces version
<i>DtEntityStateRepository</i>	<i>DtVrfVrlEntityStateRepository</i>
<i>DtAggregateStateRepository</i>	<i>DtVrfVrlAggregateStateRepository</i>
<i>DtEnvironmentProcessRepository</i>	<i>DtControlObjectStateRepository</i>

- ♦ Subclasses of the task, set, and conditional dialog boxes that display a list box (for example, those used in the fuelDepot, addTaskGui, and addConditionGui examples) now take an extra parameter to the init() member function:

```
virtual bool init(DtTMapArea* map, const DtSimTask* baseTask = NULL, int
    iNumListBoxes = 1);
virtual bool init(DtTMapArea* map,
    const DtSetDataRequest* baseSet = NULL, int iNumListBoxes = 1);
virtual bool init(DtTMapArea* map,
    const DtSimCondExpr* baseExpr = NULL, int iNumListBoxes = 1);
```

If your subclasses override init() member functions, make sure to add the new parameter. You can then dynamically case the incoming object (if it is not NULL) to populate your dialog with initial information to allow users to edit your new tasks, sets or conditionals.

- ♦ The *DtTdbDrawer* has been removed and replaced with a series of classes that are used specifically for a part of terrain drawing (polygon drawer, feature drawer, and so on.) If you have previously overridden the functionality to change how grid line labels are displayed, you must subclass the *DtTerrainGridLineDrawer* rather than the *DtTdbDrawer* as follows:

```
class MyGridLineDrawer : public DtTerrainGridLineDrawer
{
public:
    MyGridLineDrawer();
    virtual ~MyGridLineDrawer();

    static DtTerrainGridLineDrawer* create()
    { return new MyGridLineDrawer; };
};
```

And, in the main.cxx:

```
DtTacticalGdbDrawer::setTerrainGridLineDrawerCreatorFcn(MyGridLineDrawer
    ::create);
```

In the subclass of the grid line drawer, the member functions to override are the same (labelForHorizLine() and labelForVertiLine().)

Documentation Updates

VR-Forces manuals have been reprinted for this release.

Command-line options for loading plug-ins are briefly described in Chapter 4, [Starting VR-Forces](#), in *VR-Forces User's Guide*. More detailed instructions for loading plug-ins are in Chapter 11, [The VR-Forces GUI API](#), in *VR-Forces Developer's Guide*.

Bug Fixes

This release fixes the following problems:

- ♦ The signature sensor component did not work in repeatable mode.
- ♦ Tasks were not correctly assigned to entities with non-English names.
- ♦ Could not load OpenFlight files in geocentric coordinates OpenFlight versions earlier 16.1.
- ♦ Could not translate conditional expression text in the conditional view panel.
- ♦ Could not translate soil type text in the GUI.
- ♦ Ground entities did not avoid linear and areal terrain features.
- ♦ Entities could not avoid obstacle control objects.
- ♦ Could not configure Beam Type for Emitter Systems.
- ♦ Aggregate position and heading did not get updated while a scenario was paused.
- ♦ Small-wheeled vehicles such as the HMMWV did not support embarkation.
- ♦ Ground entities avoided water features by default. This interfered with behavior such as crossing bridges.
- ♦ IFF controller could not be configured through the OPD Editor.
- ♦ Given a very straight (but not quite) vector road, the path planner would "optimize" out some of the intermediary points, creating a route that is in some cases ran substantially off the vector road.
- ♦ Terrain profile widget did not display correct terrain height for units other than meters.
- ♦ If a fixed-wing entity was given a move-along task, a subsequent fly-along task could fail.
- ♦ User data associated with overlay objects was not saved to the order of battle file or sent to the front-end.
- ♦ The Overlay palette dialog box and create control objects dialog box did not show translated text correctly.
- ♦ DIs embarked on a 5 ton truck started out in sitting posture while the simulation was paused. When you pressed play, the DIs immediately stood up.
- ♦ A rotated ellipse with a fill pattern did not draw correctly.
- ♦ Shapefiles containing geodetic coordinates did not load correctly.
- ♦ Could not rotate or resize ellipse overlay objects.
- ♦ Sensor performance in large scenarios was poor.
- ♦ Could not configure "Embarked on Any" condition through the GUI.
- ♦ The "Any Subordinate Of Selected Aggregate" conditional expression modifier did not work. 10
- ♦ Selecting "Self" or Pattern in the "Entity Destroyed" conditional widget caused crash.
- ♦ Cruise missiles did not maintain altitude when targeting moving objects.

- The “decide to give up” behavior could halt execution of a plan.
- Local overlay objects were not saved with sufficient precision, resulting in a slight shift in position upon reload.
- Ground entities got stuck executing movement tasks if waypoints or points along routes were unreachable.
- Dismounted infantry turned at unreasonably high rates.
- Ground entities could get stuck inside an obstacle.
- DI aggregates accelerated unexpectedly when moving along a route.
- Soil types were not mapped according to *surfChar.map*.
- Signature sensors did not use a lifeform’s dynamic eyepoint.
- Using IFF could result in network packet flooding.
- Emitter beam angles were published with the incorrect width.
- The muzzle offset for the patriot launcher and T-80 were incorrect.
- Scenarios containing text from translated languages were sometimes unloadable.
- Terrain databases in Lambert projection were not saved correctly.
- Not all articulated part types were supported.
- When tasking multiple aggregates to move along the same route, only the first aggregate was able to complete the task.
- Range ring center lines (associated with the direction of the turret) were off by 180 degrees.
- The layout of some company and platoon level aggregate formations were wrong. This could result in some aggregates being unable to get in formation.
- The display of emitter beams timed out when a scenario was paused (DIS only).
- VR-Forces fixed-wing entities could not follow remote fixed-wing entities.
- Radio transmitters were not always displayed in HLA.
- Rotary wing entities crashed when embarked on a carrier.
- The `useAbsoluteTimeStamps` parameter in *vrfGui.mtl* and *vrfSim.mtl* was not supported in DIS.
- Raster map layers did not work under Linux.
- Conditional expressions comparing resources as percentages did not work.
- Friendly dismounted infantry did not take damage from indirect fire.
- Not all objects were deleted in an overlay layer when the layer got deleted.
- Plans containing embark tasks immediately followed by a disembark task did not work correctly.
- IFF information was not always correctly displayed in an entity’s information dialog box in HLA.
- Rotary wing entities followed the original route, even after the route was moved.
- Distributed aggregates were not correctly reconstituted after reloading a distributed scenario.

- ♦ The entity list in the GUI did not display entity types with kinds other than 1 or 11 (for example sensors with kind 9 did not show up in list).

Known Problems

VR-Forces Release 3.10 has the following known problems:

- ♦ The intervisibility display does not always reflect what simulated entities perceive in the back-end. For example, the intervisibility does not ignore water surface polygons while entity models do.
- ♦ Occasionally, VR-Forces is unable to load a scenario. When this happens, delete the contents of the *./data/temp* directory and try loading again.
- ♦ The Subordinate Manager allows duplicate entries to exist.
- ♦ Subsurface entities sink to unreasonable depths when destroyed.
- ♦ The map display does not correctly adjust entity icons when the force type is changed.
- ♦ DIs may unexpectedly change posture when climbing steep terrain.
- ♦ Determining when a scenario is completely loaded (all entities have been created and the aggregate hierarchy is completely constructed) is up to the user. Deciding when a scenario is completely loaded and ready to run can be difficult with large, distributed scenarios.
- ♦ Avoidance of linear vector features often fails at segment intersections.
- ♦ Auto-generated aggregate formation routes may contain spurious points if the original route contains sharp turns and short line segments.
- ♦ Intersection lines in intervisibility fans sometime change colors as you change the zoom level.
- ♦ Torpedoes can fly above water in certain circumstances.
- ♦ Fixed-wing entities crash when tasked to move into formation.
- ♦ Surface entities cannot move on the *California_High.gdb* terrain database.
- ♦ A disembark statement followed by a set formation statement in a plan does not work.
- ♦ You cannot use the lifeform acquire controller with a missile launcher.
- ♦ Ground entities' speed does not affect turning radius.
- ♦ Entity intervisibility with point features does not work correctly.
- ♦ Vector feature orientation is not displayed correctly in the GUI. Features are always displayed with an orientation of 0.
- ♦ Fixed-wing entities generate too many PDUs when moving to a waypoint in a straight line.
- ♦ Rotary-wing terrain avoidance sometimes fails.
- ♦ Putting the *vrfsim* console in select mode freezes the back-end.

- Rotary-wing entities do not maintain correct altitude in geocentric terrain when moving very large distances.
- You cannot create a new scenario with a Linux GUI and a Windows back-end.
- Aggregate subordinates cannot execute their own individual plans.
- In-frustum sensor geometry does not work for angles greater than 90 degrees.
- The GUI does not allow you to enter IFF modes in octal.
- Geometry files referenced by a MetaFlight file must be in OpenFlight format, and not in Vega Prime's binary format (*.vsb*). MÄK's terrain libraries do not support *.vsb* files.
- File paths that are used within a MetaFlight file to designate the OpenFlight data files must be fully qualified absolute paths. Relative paths, and paths that include environment variable expansions are not supported.
- Terrain databases processed with the balance option may display incorrectly.
- Surface entities are not configured with weapons.
- Playing a MÄK Data Logger recording that has the same site:app IDs as a VR-Forces application may cause VR-Forces to crash.
- Component descriptions in Appendix B, *Component Model Descriptions*, in *VR-Forces Developer's Guide* are out of date.
- You cannot load vector data on geocentric terrain.
- Loading *.c4b* files may cause application to hang on Linux.
- CADRG files cannot be configured through the TDB Tool GUI. You must manually edit MTD files to include them.
- Vector features created using the TDB Tool that have non-zero height above terrain do not work correctly with the VR-Forces path planning algorithm.
- When you record a scenario with MÄK Logger 4.0, at the beginning of the recording session, VR-Forces may freeze for a while due to the large number of messages required for subscription and publication.
- VR-Forces may become unstable, leak memory, or both when running with the MÄK RTI LRC GUI enabled (observed while running with MÄK RTI 3.1.) We recommend you use the web-based RTIsy GUI instead.
- Do not play the *maklanddemoHLA1516_vrlink_xml.lgr* file (included with MÄK Logger 4.0 release) while running VR-Forces using draft 17 of the RPR FOM. Running this file in this particular configuration produces memory leaks. Run VR-Forces using RPR FOM 1.0 when working with this Logger recording.

- ♦ Plug-ins built for HLA 1516 in debug mode do not load properly when they are specified in plugins.mtl. To load an HLA 1516 debug plugin, use the command-line, for example:

```
vrfSim --loadPlugin my1516plugind
```

If you need to use the plugins.mtl file to load plugins, then to load an HLA 1516 debug plugin, you must build vrfSim or vrfGui in debug mode for use with the plugins.mtl file. Specify the plugin as follows:

```
(loadPlugin "my1516plugind")
```

- ♦ Using IFF in HLA 1516 exercises can result in packet flooding.

