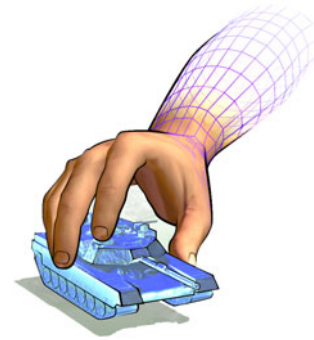




VR-Forces 3.11 Release Notes

This release note contains the following release-specific information for VR-Forces Release 3.11:

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	4
FLEXIm Support.....	4
Third-Party Library Requirements.....	5
Patch Required for AMD Dual-processor Windows PCs	5
Font Support on Linux.....	5
Network Compatibility.....	6
Backwards Compatibility	6
RPR FOM Versions Supported.....	7
New Features and Updates.....	8
VR-Forces 3D GUI	10
Entity Editor	10
Scenario Merge Tool.....	11
Upgrade to FLEXIm 11.4.....	11
New Naming Convention for DIS Directories	11
New Entities.....	11

Copyright © 2008 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-3.11-3-080212

New Tasks and Set Data Requests	11
Ability to Send and Receive Radio Messages.....	12
Support for External Communications Servers.....	12
Multiple Detection Levels for Discovered Targets.....	12
Use of OPCODE in the TDB Tool.....	12
Entity Behavior Changes	13
Configuration and Behavior Changes.....	13
Simulation Model Sets and System Definitions.....	13
Changes to OPE Files	14
Miscellaneous Changes	15
Changes to Command-line Options	15
GUI Features.....	15
Create a Plan and Assign it to Multiple Entities	15
Font Changes	16
Copy and Paste Objects	16
Entity Palette for Creating Objects	16
Icon Toolbars.....	16
API Changes	17
Human Movement Model	17
Entity-Based Plan Management.....	17
New Communications Mechanism.....	17
Support for Plug-ins in the OPD Editor.....	17
Entities Share Detection Lists	18
Versioning of VR-Forces Messages	18
Improved API Class Documentation	18
Upgrade to Qt 4.3.3	18
Miscellaneous API Changes	22
Documentation Updates.....	22
Bug Fixes	23
Known Problems	24

Systems Supported

VR-Forces 3.11 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
Red Hat Enterprise Linux Workstation 4, 5 *	Default compiler
Fedora Core 7	
Windows XP	Microsoft Visual C++ 7.1
	Microsoft Visual C++ 8.0 SP1
*The VR-Forces 3D Front End for Linux is only available on WS 4.	

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 gmake 3.81 or later. For operating systems with an older version of gmake (currently only Red Hat WS 4), a binary has been included for your convenience in the *./mkbin* directory. This version 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>.

Before you compile the examples, go to the top level of your installation and create a symbolic link called 'VR-Link' to your VR-Link installation, and another called 'RTI' to your RTI installation. You must use the versions listed under [MÄK Product Compatibility](#).

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 3.11.1. If you are building for HLA and want to link with the MÄK RTI, use MÄK RTI 3.x.

VR-Forces 3.11 is a parallel release to MÄK Plan View Display 2.11.

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 4.3.3. 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 3.11 uses FLEXlm 11.4. You must copy the files in *./flexlm11.4* 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.

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 MAK products to run backwards. This occurs when the Windows scheduler changes the CPU the MAK 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.

Font Support on Linux

The *fontconfig* provided with Red Hat Enterprise Linux Workstation 4 does not properly name the fonts. Newer operating systems are not affected by this issue. In order to load the fonts, you'll need to provide the following alias in *fontconfig*. The alias can either be added to the system configuration in */etc/fonts/fonts.conf* or it can be added to *~/.fonts.conf* on a per-user basis. For more information about *fontconfig*, please see <http://fontconfig.org/fontconfig-user.html>.

```
<alias>
  <family>MAK MilStd 2525b</family>
  <prefer><family>MilStd 2525b</family></prefer>
</alias>
```

Network Compatibility

HLA only

VR-Forces 3.11 was built against VR-Link 3.11.1 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 3.11 is compatible with applications that use earlier versions of VR-Link if they support versions of the RPR FOM listed.



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 3.11. VR-Forces 3.11 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 3.11 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 3.11 APIs are not compatible with VR-Forces 3.10.x or earlier. Scenarios and object parameter databases created with VR-Forces 3.10.x are forward compatible.

RPR FOM Versions Supported

VR-Forces 3.11 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 3.11 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 3.11 relies on the LgrControl and View-Control custom MÄK extensions.

New Features and Updates

VR-Forces 3.11 has the following new features and updates:

- ♦ VR-Forces 3D GUI
- ♦ Entity Editor tool for adding new units, editing entity systems, editing some entity attributes, creating simulation model sets and editing the Create menu
- ♦ Scenario Merge tool
- ♦ Upgrade to FLEXlm 11.4
- ♦ New naming convention for DIS directories
- ♦ New entities:
 - Dismounted infantry with rocket propelled grenades (RPGs)
 - Roadside IEDs (improvised explosive devices)
 - Suicide bombers (car bombs and humans configured with explosives).
- ♦ New tasks and set data request
 - Intercept and Destroy
 - Detonate Object
 - Send Radio Set, Send Radio Task, Send Text Message
 - Set Armed
 - Set Disembarked
 - Set Embarked.
- ♦ Ability to send and receive radio messages
- ♦ Support for external communications servers
- ♦ Multiple detection levels for discovered targets
- ♦ Support for CTDB C7I terrain databases
- ♦ Use of OPCODE, a collision detection library, is now available as an alternative to spatial subdivisions for faster terrain intersection calls.
- ♦ Entity Behavior Changes
- ♦ Configuration and parameter changes:
 - Context-sensitive parameter descriptions for the OPD Editor
 - Simulation model sets and system definitions
 - Changes to OPE files
 - Changes to command-line options.

- ♦ GUI additions and changes:
 - Create a plan and assign it to multiple entities
 - Ability to place entities on different floors of a building
 - Font changes
 - Copy and paste objects
 - Entity Palette for creating objects
 - Icon toolbars for common tasks
 - The LOS Filter has been removed
 - Map area compass
 - The keyboard shortcut for opening the View Plan window has changed to **Shift+P**
 - The keyboard shortcut for opening the Time Management dialog box has changed to **Ctrl+Shift+M**
- ♦ API changes:
 - Upgrade to Qt 4.3.3
 - Entity-based plan management
 - New communications mechanism to support sending and receiving radio messages
 - The vrfGuiUser solution file has been removed. The vrfGuiUser project has been moved to the frontEndExamples workspace file.
 - New human movement model
 - Support for plug-ins in the OPD Editor
 - Entities share detection lists
 - Versioning of VR-Forces messages
 - Improved API class documentation.

VR-Forces 3D GUI

The VR-Forces 3D GUI is an alternative front end to the traditional 2D plan view display GUI. The 3D GUI provides enhanced situation awareness. It allows you to place entities into a scene with a precise position and orientation. This is particularly useful for creating scenarios in urban terrain, in which entities must be correctly located relative to urban structures and within buildings that support multiple polygon levels.

The 3D GUI essentially embeds the MÄK Stealth in VR-Forces. As such, it supports Stealth plug-ins created with the Stealth plug-in API. It also supports Stealth view control messages and the Stealth remote draw API messages.

A VR-Forces license allows you to run an instance of the 2D front-end or the 3D front-end, but not both at the same time. The VR-Forces 3D GUI is documented in *VR-Forces 3D Front End User's Guide*. Configuration details are included in *VR-Forces Configuration Guide*.

Using GUI API Example Plug-ins in the 3D Front-end

Example plug-ins that modify menus or add dialog boxes should work equally well in the 2D and 3D front-ends. GUI API example plug-ins should be loaded the same way in both front-ends. However, there are some examples that manipulate the terrain map area that are only appropriate for use with the 2D front-end. The following example plug-ins are for use with the 2D front-end only:

- ♦ drawMissileImpact
- ♦ filterMenu
- ♦ showMissileImpact
- ♦ viewFilter
- ♦ intervisibilityExtension.

Entity Editor

The Entity Editor greatly simplifies the task of editing entity models. You can easily create new entity models based on existing ones or create new ones from the ground up. You can add components to entities by packaging them as weapons systems, sensors systems, movement systems, and damage models. You can add these systems and edit their properties through simple point and click lists and dialog boxes.

In addition to adding and removing systems, the Entity Editor lets you specify the image to use for an entity, its force, and its category (ground, surface, and so on). You can also create custom Create menus.

The Entity Editor is now the preferred method for editing entity parameters and components. However it does not provide access to all parameters. Advanced VR-Forces users can still use the OPD Editor for this purpose. For details, please see Chapter 5, *Editing Entity Models*, in *VR-Forces Configuration Guide*.

Scenario Merge Tool

The Scenario Merge tool is a new off-line tool that lets you merge scenarios. This facilitates use of templated scenarios and distributed scenario creation. For details, please see Chapter 4, *Merging Scenarios*, in *VR-Forces Configuration Guide*.

Upgrade to FLEXlm 11.4

VR-Forces 3.11 requires FLEXlm 11.4. To upgrade from an earlier version of FLEXlm, copy the files in *.flexlm11.4* to the directory from which you run FLEXlm. You do not need to update your license file unless your maintenance agreement has expired.

New Naming Convention for DIS Directories

Directories that contain files specific to DIS are now named binDIS rather than bin5. This change affects the project files provided with VR-Forces.

New Entities

VR-Forces has added several new entities to better support urban warfare scenarios, as follows:

- ♦ Dismounted infantry with RPGs
- ♦ IEDs. You can place IEDs in the scenario and set a fuse based on time, proximity, or immediate explosion.
- ♦ Suicide bombers. VR-Forces includes cars and lifeforms with explosives that can be detonated similarly to IEDs. You can also embark IEDs on platforms to create your own suicide entities.

These new entities are supported through the Intercept and Destroy and Detonate Object tasks and the Set Armed set data request.

New Tasks and Set Data Requests

VR-Forces has new tasks and set data requests that support the new communications models, IEDs, and embarkation, as follows:

- ♦ You can send set data requests to entities using the Send Radio Set task.
- ♦ You can send tasks to entities using the Send Radio Task task.
- ♦ You can send arbitrary text messages to entities over the radio using the Send Text Message task. Entities can test for receipt of text messages in their plans.
- ♦ The Set Embarked and Set Disembarked set data requests allow you to instantly embark or disembark entities in plans or through the GUI similarly to the Embark On and Disembark commands that are on the Entities menu.

- The Set Armed set data request allows you to arm an IED or suicide bomber entity. While they are armed by default, this set data request allows you to change their fuse type and other parameters.
- The Detonate Object set data request allows you to immediately detonate an IED or suicide bomber.
- The Intercept and Destroy task directs an entity to move to and attack another entity using its offensive capabilities.

Ability to Send and Receive Radio Messages

You can task entities to send radio messages to other entities. You can also test for receipt of text messages. For details, please see Chapter 10, [Assigning Tasks](#) and [The Receive Text Message Condition](#), in *VR-Forces User's Guide*. For details about configuring radio communication, please see [Section 8.5, "Configuring Radios"](#), in *VR-Forces Configuration Guide*.

Support for External Communications Servers

You can configure VR-Forces to use an external communications effects server to control the transmission of radio messages between VR-Forces entities. The communications effects server determines which entities are able to communicate with each other, and how long it takes a message to be transmitted from the originator of the message to the receiver of the message.

Any communications effects server that meets the defined interface can be used. Currently, QualNet with the Standard Interfaces library meets this interface. For details, please see Chapter 9, [Using VR-Forces with an External Communications Effects Server](#), in *VR-Forces Configuration Guide*.

Multiple Detection Levels for Discovered Targets

VR-Forces now classifies the quality of identification of targets based on distance and length of time since the target has been discovered. Contacts are classified as detected (I see something), classified (I know the platform), recognized (I know the entity type), and identified (I know everything about the object.) You can test for the identification level in plans. For details, please see [Section 8.6, "Target Detection"](#), in *VR-Forces User's Guide*.

Use of OPCODE in the TDB Tool

OPCODE is a third-party library that does triangle mesh intersections using a fast algorithm. Using the TDB Tool, you can configure terrain databases to use OPCODE or spatial subdivisions, for faster terrain intersection calls. When you create or edit databases, you can choose OPCODE, spatial subdivisions, or a simple intersection algorithm for terrain intersections. For details, please see [Section 2.7, "Optimizing Databases Using the TDB Tool"](#), in *MÄK Terrain Database Tool User's Guide*.

Entity Behavior Changes

- The damage models have changed slightly, so some scenarios may behave differently than in previous releases. For example, the T80 now has the same armor (and kill probabilities) as the M1A2. Significant behavior changes may be seen in scenarios containing these types of entities.
- Sensor parameter values have changed, due to the introduction of multiple detection levels.
- The new human movement model models motion of individual lifeform entities quite differently from the previous model. The new model does not model acceleration or rotational velocity (which the previous model included).

Configuration and Behavior Changes

This section describes changes to configuration parameters, entity models, and other issues primarily described in *VR-Forces Configuration Guide*.

Simulation Model Sets and System Definitions

The object parameter database and its supporting configuration files has been reorganized using the concept of a simulation model set. Simulation model sets are the organizational unit for the Entity Editor.

A simulation model set contains all the information necessary to list entities in the Entity Palette, configure their appearance in the map, and configure their behavior (through the object parameter database.) As a consequence, the files formerly in the `./data/parameters` directory are now in the `./data/simulationModelSets` directory. Some configuration files have been moved from the `./config` directory to appropriate subdirectories of *simulationModelSets*.

System definition files are configuration files referenced from an entity's OPE file. They contain the set of components needed to implement a particular functional characteristic of an entity, such as how it moves, how it senses its environment, how it takes damage, and which weapons it has. One result of this new way to organize components is that OPE files are a bit shorter and more readable than they were in the past. The primary benefit of this approach is that it is now much easier to add weapons systems to an entity than it was in previous releases. For details, please see [Section 5.1.1, "Simulation Model Sets"](#), in *VR-Forces Configuration Guide*.

Changes to OPE Files

The following changes have been made to object parameter entry (OPE) files:

- Each object now has its own plan manager subcomponent specified with the "plan-manager" parameter. For local objects the default should be "vrfoobject-plan-manager" and for remote objects it should be empty (""). In general only objects that have a task manager need a plan manager.
- All individual entities that have a sensor, have had their sensor updated. The changes were additions not modifications. A Detection Level Determinator was added to the sensors and an associated combat identification level table. This allows you to have a graded scale of detection as opposed to the old way, which was probability based (you detect it you know everything about it.) Now you learn information about the target over time.
- All objects now have a "radios" block. Most entities have one radio defined in this block. The old "radio" parameter that was part of the local/remote subcomponents has been removed.
- All objects that have a radio have had a "send-radio-message-controller" added to the controller list, which carries out the new Send Radio Message tasks.
- Sensors no longer have a **detection-forces** parameters. Instead, they have a **detect-only-hostile-forces** parameter. This parameter applies only to types specified in the detection-types of the sensor, and not to other types provided on an input port by other components.
- The target selection controller no longer has a parameter **hostile-forces**. It uses the new hostility manager to know which forces it should target.
- The Spot Report Generator's parameter **object-types-to-spot-report** list format has changed. It no longer specifies a list of force types, It is now a simple list of object types, much like other parameters in the OPE file. Force types to send spot reports on are now determined by the new parameters **send-spot-reports-on-hostile-forces** and **send-spot-reports-on-neutral-forces** and the use of the hostility manger.
- The **first-time-placed** parameter has been removed from aircraft OPE files. Fixed wings are no longer only placed on the ground when they are first being placed.
- The **initial-desired-height** parameter for rotary-wing entities has been removed. They are no longer placed at this starting height since they can now be placed at any altitude.
- We have officially dropped backward-compatibility for the old format OPD file (the version that existed prior to the introduction of OPE files).

Miscellaneous Changes

- The OPD Editor displays a context-sensitive help for parameters. When you select or move the mouse cursor over a parameter, it display a description of the parameter.
- The default for `forceParameterDbReload` in *vrfSim.mtl* has been changed to 1. When editing a simulation model set with the Entity Editor, all the changes made to the symbology and create menu take effect immediately however without `forceParameterDbReload` the OPD changes don't take effect until after you switch OPD's or restart the back-end. This is too confusing for the novice level user the Entity Editor was designed for, so `forceParameterDbReload` is now set by default.
- The `exerciseStartTime`, `entityAdvisory`, and `simTime` parameters have been removed from *vrfSim.mtl*.

Changes to Command-line Options

The following command-line options have been removed:

- `-j`
- `-k`
- `-E` | `--entityAdvisory`

GUI Features

This section describes enhancements in the VR-Forces front-end.

Create a Plan and Assign it to Multiple Entities

You can now write a plan and assign it to many entities at the same time. This may be particularly useful for crowd scenes and use with the B-HAVE Module. For details, please see [Section 13.8, "Writing a Plan for Multiple Entities"](#), in *VR-Forces User's Guide*.

Placing Entities Inside a Building

If a terrain database has buildings with multiple levels, you can place entities on any floor from both the 2D and 3D GUIs. When you add an entity to a scenario, you can specify its altitude. VR-Forces identifies the different altitudes available in the building and lets you specify which one to place the entity on. You can also specify any arbitrary altitude in the building and VR-Forces will clamp the entity to the nearest surface. For details, please see [Section 8.2.3, "Specifying an Altitude for a New Entity"](#), in *VR-Forces User's Guide*.

Font Changes

The MIL-STD-2525B font, from MÄK Technologies, which is used to represent entity symbols in the GUI, is now the default font on both Windows and Linux.

The 2525A font from Mapsyms is no longer distributed, but it can be downloaded from the Mapsyms website: www.mapsyms.com.

It is no longer necessary to install the font into the operating system because it is now loaded via a call to `QFontDatabase::addApplicationFont`. The font is referenced by relative path in its new location: `../data/symbols/fonts`. You can load additional fonts by using the `addFont` parameter in `vrfGui.mtl`.



On Linux, if you manually installed fonts for a previous release of VR-Forces, uninstall them before running VR-Forces 3.11. Loading an older version of the font may cause symbol display issues.

Copy and Paste Objects

You can now copy and paste entities and graphical objects. Entity plans, appearance, heading, and rules of engagement get copied with the entity. You can copy entities between scenarios, in which case you can keep their original names. Entities copied within scenarios get renamed. For details, please see [Section 7.6, "Copying and Pasting Objects"](#), in *VR-Forces User's Guide*.

Entity Palette for Creating Objects

The Utility toolbar has a new tab, Create, which houses the Entity Palette. The Entity Palette allows you to quickly select and create entities without using the menu system. You can create a list of frequently created entities to speed up entity selection. For details, please see [Section 8.2.1, "Using the Entity Palette to Create Entities"](#), in *VR-Forces User's Guide*.

Icon Toolbars

VR-Forces now has icon toolbars for common tasks such as opening and saving scenarios and changing the drawing mode.

API Changes

This section describes the major changes to the VR-Forces APIs.

Human Movement Model

In previous releases, individual lifeforms used components derived from the automotive components. Lifeforms now use components designed to more closely simulate human movement. For details about new lifeform components, please see the class documentation.

Entity-Based Plan Management

In previous versions of VR-Forces, plans were managed by a central Plan Manager. The *DtSimPlanManager* has been replaced by a *DtVrfObjectPlanManager* which is attached to each object. The centralized functions of *DtSimPlanManager*, such as reading and writing plan files, have been moved to *DtVrfObjectPlanAdministrator* which is owned by *DtVrfObjectManager*. The new architecture is much cleaner because now the *DtSimTaskManager* can directly communicate with the plan manager without having to send interface messages. It's also more efficient because there is no longer any need to do lookups in a secondary object hash in *DtSimPlanManager*. For details, please see Chapter 9, [Plans](#), in *VR-Forces Developer's Guide*.

New Communications Mechanism

In previous versions of VR-Forces, there was no distinction between messages used for internal VR-Forces, for example, a set data request message, and messages that simulated radio traffic. The communications mechanism has been changed to separate these types of messages. This change supports the new ability to send radio messages from entities to other specific entities. For details, please see Chapter 7, [Entity Communication](#) in *VR-Forces Developer's Guide*.

Support for Plug-ins in the OPD Editor

In previous version of VR-Forces, the OPD Editor was restricted to configuring the components shipped with VR-Forces. The OPD Editor now supports the VR-Forces plug-in architecture, so that you can update it to handle components created by customers.

Entities Share Detection Lists

Because of the new detection levels feature (please see [Multiple Detection Levels for Discovered Targets](#)), targeting of hostile entities depends, in part, on the certainty of detection. Entities can now share their detection lists through the use of spot reports. The spot report information augments the contacts that an individual entity has made. For details, please see [Section 8.6.1, "Target Detection and Spot Reports"](#), in *VR-Forces User's Guide*.

Versioning of VR-Forces Messages

In the past, when different versions of VR-Forces tried to play together in a simulation, differences in message formats could cause crashes. Messages will now be tagged with the VR-Forces version so that in the future, VR-Forces can filter out messages from incompatible versions.

Improved API Class Documentation

The HTML class documentation now includes additional pages that provide high-level descriptions of the VR-Forces component architecture and describe how components and systems are interrelated. The new documentation also includes lists of the components used by each entity type and has links to the parameter descriptions in the OPD Editor online help. This updated documentation replaces several appendices that were formerly included in *VR-Forces Developer's Guide*.

Upgrade to Qt 4.3.3

The VR-Forces GUI and all of its tools (OPD Editor, TDB Tool, and so on) have been migrated to Qt 4.3.3. The Qt API changed extensively from Qt 3.x to Qt 4.x. If you have modified the VR-Forces GUI and want to migrate to VR-Forces 3.11, you must upgrade to the supported version of Qt and migrate your customized code.

This section covers the main points you should consider when upgrading source code from Qt 3.x to Qt 4.x. If you do not find information in there that you need, please look at the <http://doc.trolltech.com/4.2/porting4.html> web site or email support@mak.com with your specific questions.

General Advice

When upgrading to the new API, you should check the prototype of the method you are overriding for changes. The main changes have been made to the types of container classes and constructors. If you find that a slot is not being called, a method not being called, or a constructor not being compiled, it is due to a method prototype change. In general, the compiler will catch most of the issues with upgrading.

gen.pro Changes

The new qmake system supports single pass project file generation, so, there is no longer a need for a separate merge program. This means, however, that the "debug" and "release" (or !debug) configuration options have changed to DebugBuild and Release-Build. If you have either of these tags in your *.pro* file, change them accordingly. The supplied project generation system has been updated to reflect these changes.

Example Project Files

In previous installations, the project files for examples were pre-generated. With Qt 4.X, this has been changed. The QTDIR environment variable has been removed in favor of hard coding the path to the Qt installation. Because of this change, it is now necessary to generate the example project files when you want to build them. On Windows, make sure the qmake application can be found in your Windows path and then run the "generateExamples" batch script from the examples directory. Once run, you can open the examples solution to build the example projects.

The exceptions to this are the back-end examples and terrain toolkit examples. Since the back-end and terrain toolkit examples do not need Qt in order to be built or run, those project files are distributed.

Constructors

- ♦ The "name" parameter has been removed from all Qt based objects. If you need to name the object class that you are creating for identification purposes, in your constructor use the setObjectName() method to set the name. You can use the name passed into the old constructor, however, the name is not passed to the parent object.

- ♦ The constructor for a popup menu has changed from:

```
MyPopupMenu(QObject* parent, const char* name);
```

to:

```
MyPopupMenu(QWidget* parent, bool display = true);
```

UI Files

All *.ui* files must be converted using the `uic3` program (in `Qt_installdir/bin`). This program, while helpful, does not complete the conversion process. After the conversion you must inspect the resultant *.ui* file for any object that references Q3 (for example, `Q3GroupBox`). If such objects are present, hand edit them to remove the Q3 nomenclature. In many cases, it might be easier to simply recreate your *.ui* file using the new Qt designer for 4.X.

Classes generated by the new *.ui* files cannot be subclassed. If your class used to look like this:

```
class MyWidget : public MyWidgetFromUi
{
    ....
};
```

it must now have the *.ui* generated class as a member variable, like this:

```
class MyWidget : public QWidget
{
public:
    MyWidget(QWidget* parent, const Qt::WindowFlags& flag);
public:
    Ui::MyWidgetFromUi myUi;
};

MyWidget(QWidget* parent, const Qt::WindowFlags& flags) :
    QWidget(parent, flags)
{
    myUi.setupUi(this);
}
```

Container Classes

The container classes have been completely changed for Qt 4.X to resemble standard template classes. Most container classes have been converted to either a `QVector`, a `QList`, or a `QHash` class. The major container changes are:

- `QPtrVector<type>` and `QValueList<type>` are now `QVector` (for example, `QValueList<DtPoint>` and `QPtrVector<DtPoint>` is now `QVector<DtPoint>`).
- `QDict<type>` is now `QHash<QString, type>`
- `QIntDict<type>` is now `QHash<int, type>`
- `QPtrDict<type>` is now specific to a particular pointer type. For example, if you create a dictionary based on a `MyObject`, the declaration looks like:
`QHash<MyObject*, type>`

Menu Changes

The menu system API has changed. We have created a system similar to 3.X, to minimize the amount of migration work needed. To support this, the following changes have been made:

- ♦ The `initMenuItem` (and `initToggleMenuItem`) methods used to take two parameters when a menu item was added. Now they only take a single text parameter. So, if your init call looked like this:

```
insertMenuItem(DtNoMenuItem, DtFilterForceTypeAction,
    trUtf8( "Force Type" ), trUtf8( "Force Type" ),
    trUtf8( "Force Type Filter." ));
```

it should now look like this:

```
insertMenuItem(DtNoMenuItem, DtFilterForceTypeAction,
    trUtf8( "Force Type" ), trUtf8( "Force Type Filter." ));
```

- ♦ `QActionGroup` is now `DtActionGroup`. If you create an action group in `initActionGroups()`, make sure to create a `DtActionGroup` class. If you return an action group in `getActionGroupForType`, make sure to change the prototype to return a `DtActionGroup`. When creating an action group, if your code looked like:

```
myFilterActionGroup = new QActionGroup(parent(), "myFilterActionGroup");
```

it should now look like:

```
myFilterActionGroup = new DtActionGroup(this);
```

If you want your action group to be an exclusive check area, make sure to call `setExclusive(true)` on the action group (or `false` if you do not want it to be exclusive).

`DtModelKey` is now a `QString` and is no longer its own class. If you have code which looks like:

```
data->key().string()
```

you can remove `.string()`.

The constant `WFlags` has been changed to `Qt::WindowFlags`.

Miscellaneous API Changes

- *DtTargetDetectionSensor*, *DtTargetDetectionPSR*, and *DtTargetDetectionSensorDescriptor* have been removed. They were replaced in VR-Forces 3.9 by the object sensor model. For details, please see Chapter 5, *Creating New Components*, in *VR-Forces Developer's Guide*.
- The *DtFixedWingPilotControllerDescriptor* now has **at-time** and **near-time** parameters, which used to be hard-coded in the controller's `atGoal()` and `nearGoal()` functions.
- `DtSimTaskManager::removeAllReferancesToController()` was misspelled. It has been renamed to `removeAllReferencestoController()`.
- `DtVrfObjectManager::defineCustomeStateRepositories()` and the *DtVrfVrEntityStateRepository* class it created have been removed; the derived state repository provided a work-around for a VR-Link bug which has since been fixed.
- A new front-end class, *DtVrfGuiDisplayOptions* has been introduced. If you have (VRF only) subclassed options from *DtPvdDisplayOptions*, you must now subclass from *DtVrfGuiDisplayOptions*. If you do not, unpredictable results may occur.
- Many elements that were deprecated in previous releases have been removed from the GUI and simulation APIs.
- The `windowsInfoExtension` example has been renamed `addGuiCoordinateSystem`.
- The `viewControllerObjects` example has been renamed `viewFilter`.
- The `rangeFilter` example was renamed to `vrfGuiRangeFilter`.

Documentation Updates

For details about installing TrueType fonts on Linux systems, please read *./data/symbols/ttf/Readme.txt*. This replaces the section in Chapter 2.

All manuals and online help have been updated for this release. Appendixes B and C in *VR-Forces Developer's Guide* have been removed. A new manual, *VR-Forces 3D Front End User's Guide*, describes the new 3D GUI front-end.

The `-j` and `-k` command-line options, described in the PDF version of *VR-Forces User's Guide*, have been removed from VR-Forces. The **simTime** and **exerciseStartTime** parameters, described in the PDF version of *VR-Forces Configuration Guide*, have been removed from VR-Forces.

The `-q` command-line option for *vrfSim* is Windows only. To create a log file on Linux, running in separate mode, redirect output to a file.

Section E.5, Attached Parts, in the PDF version of *VR-Forces Configuration Guide* describes features that are not supported. This section has been removed.

Section 5.3.5 in the PDF version of *VR-Forces Developer's Guide* incorrectly states that the minimum tick period can only be specified in real time. The time can be expressed as either real time (wall clock time) or simulation time, depending on the value of the **tick-period-uses-real-time** parameter.

The tutorial in *VR-Forces User's Guide* that describes the breaching scenario is slightly different from the breaching scenario shipped with VR-Forces. The vehicles named M9 Ace in the tutorial correspond to the vehicles named Ar Utl in the scenario.

Some of the default vehicle models illustrated in [Section 12.5, "Models Supplied with the 3D Front End"](#), in *VR-Forces Configuration Guide* have been replaced with new models.

Bug Fixes

This release fixes the following problems that were present in prior releases:

- ♦ The GUI does not allow you to enter IFF modes in octal.
- ♦ 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.
- ♦ CADRГ files cannot be configured through the TDB Tool GUI. You must manually edit MTD files to include them.
- ♦ Loading *.c4b* files may cause application to hang on Linux.
- ♦ Using IFF in HLA 1516 exercises can result in packet flooding.
- ♦ Vector feature orientation was not displayed correctly in the GUI and was not taken into account by the back-end.
- ♦ Ground vehicles that are not capable of pivoting (such as a HMMWV) appeared to jitter when given a turn to heading task.
- ♦ Rotary-wing entities ignored a skip task directive when executing movement tasks.
- ♦ Fixed-wing entities sometimes appeared to partially sink into the deck of a carrier.
- ♦ The Set Appearance paint scheme attribute was ignored by HLA.
- ♦ Plans are no longer evaluated when paused. This caused unexpected behavior, such as conditional expressions being evaluated and triggers firing.
- ♦ Creating a custom spot report view did not work.
- ♦ The OPD Editor did not allow users to set the name of a given component.
- ♦ Puffs (hazard clouds) were always drawn as circles, even if they should have been drawn as ellipses.
- ♦ The TDB Tool could not load project files that contain filenames that have spaces in them.
- ♦ If you took control of a ground vehicle or fixed-wing with a joystick and then released control by joystick, the ground vehicle would not respond to any future movement tasks (immediate tasks or tasks received as part of a plan).
- ♦ Transmitters on moving entities caused packet flooding.
- ♦ Task complete reports were not addressed to specific recipient (they were sent to a broadcast address).
- ♦ VR-Forces did not produce repeatable results, because the detonation manager did not store data in a deterministic manner.

- When running the fixed-wing model at lower frame rates (for example, 2 Hz), the aircraft sometimes lost substantial altitude when transitioning to the next vertex (> 50 meter drops in altitude have been observed). This problem was only discernible at lower frame rates.

Known Problems

VR-Forces Release 3.11 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.
- 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.
- 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.
- ♦ 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.
- ♦ Terrain databases processed with the balance option may display incorrectly.
- ♦ Surface entities are not configured with weapons.
- ♦ You cannot load vector data on geocentric terrain.
- ♦ 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.
- ♦ 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.
- ♦ Rotary-wing entities are not able to reach moving waypoints or land on moving ships. A rotary-wing entity will approach a given waypoint or landing point, but will never completely reach the goal.
- ♦ Reorganizing an aggregate while it is executing a task may produce unexpected results. The aggregate may be unable to complete its task.
- ♦ When control objects are placed on moving objects (such as a route on a moving aircraft carrier), they produce a flood of network packets.
- ♦ Open File dialog boxes may be slow to display file lists. This is due to underlying problems with the Qt toolkit.
- ♦ When you execute a task that requires use of an Open File dialog box, you may receive an error message similar to:
`Exception Processing Message .....`
This is due to problems in the Qt Toolkit. To work around this problem, insert a floppy disk in the A: drive.
- ♦ Ground vehicles can climb up unrealistically steep slopes (greater than 70 degrees in some cases).
- ♦ Features cannot be selected on geodetic terrain databases in the TDB Tool.
- ♦ Embarked entities cannot shoot or be shot at by other entities.
- ♦ The DtOrganizationManager cannot handle changing the force of an entity. It is unable to update the echelon ID of the entity correctly.
- ♦ Fixed-wing entities do not respond to Turn to Heading tasks.
- ♦ Dead lifeform entities may unexpectedly block line of sight. Their bounding volumes are always upright, regardless of whether or not they are dead.
- ♦ Rotary-wing entities do not avoid other entities. They will fly through one another.

- ♦ The target acquisition and selection model can sometimes get 'stuck' trying to acquire a target. If a target has been selected but is unable to be acquired for some reason, it will continue in a futile attempt to acquire that target, even if there are other more viable targets available.
- ♦ If you set the pan increment to 100% for small terrains, the map does not pan by 100%. This is due to the algorithm used to calculate the distance that the map is offset.
- ♦ Transmitter lines are not drawn if the sender is offscreen.
- ♦ If you use the MÄK Logger to record VR-Forces scenarios in HLA1516 (particularly with RPR FOM version 2.0 draft 17), and you are using the MÄK RTI, VR-Forces and the Logger may become unresponsive. To avoid this problem, disable `RTI_enableLrcWebService` in the *rid.mtl* configuration file.
- ♦ The MÄK Logger does not record batch file runs if a previous recording exists. By default, the Logger does not allow you to overwrite existing files using the remote control API. So, once you record a batch run, you cannot record a subsequent run of the same batch. You can work around this problem by editing the *lgrConfig.xml* file to permit overwriting of Logger files. Make the following edits in the configuration file for each protocol for which you want to record batch runs:

Add the following lines to the <defaults> section:

```
<var name="processRemoteControl" type="bool" value="true"/>
<var name="recordRemoteControl" type="bool" value="true"/>
<var name="allowDeleteRemoteControl" type="bool" value="true"/>
```

Add the following lines to the <commands name="startup"> section:

```
<command domain="Simulation" protocol="DIS"
  command="SetRemoteControlSettings">
  <param name="process" var="processRemoteControl"></param>
  <param name="record" var="recordRemoteControl"></param>
  <param name="allowDelete" var="allowDeleteRemoteControl"></param>
</command>
```

Use the appropriate value for the protocol attribute.

- ♦ If you create an aggregate composed of rotary-wing entities, in which some or all of them are on the ground, when you start the scenario, they will be destroyed. To prevent this from happening, set the aggregate's altitude to a reasonable height above terrain (such that all rotary-wing entities are airborne) before starting the scenario.
- ♦ Embarked entities controlled by joystick cannot disembark. For example, if you use a joystick to control a fixed-wing taking off from a carrier, the aircraft will take off, but it will remain in an embarked state.
- ♦ Certain GeoTIFF images may fail to load on linux. If you encounter such an image, please contact MÄK support.