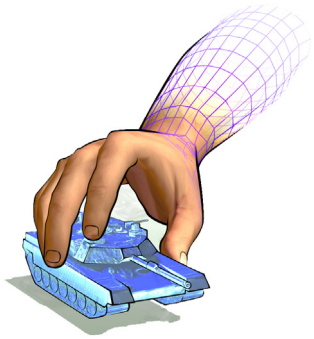

VR-Forces 3.12 Release Notes

This release note contains the following release-specific information for VR-Forces Release 3.12:

- Systems Supported 2
  - Using Libraries and Binaries Built with Visual Studio 2005 and Later 2
  - Building on Linux 3
  - Disk Space Requirements 3
  - Compiler Compatibility on Windows 3
  - MÄK Product Compatibility 3
  - Qt Release Compatibility 4
  - FLEXlm Support 4
  - Third-Party Library Requirements 4
  - Patch Required for AMD Dual-processor Windows PCs 4
  - Font Support on Linux 5
- Network Compatibility 5
- Backwards Compatibility 5
- RPR FOM Versions Supported 6
- New Features and Updates 7
  - Support for MS-Windows 64-Bit 11
- Documentation Updates 12
  - The Entity Editor and OPD Editor Plug-in API 14
- Bug Fixes 16
- Known Problems 17

Document ID: VRF-3.12-3-090714

Systems Supported

VR-Forces 3.12 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
Windows XP	Microsoft Visual C++ 7.1
	Microsoft Visual C++ 8.0 SP1
	Microsoft Visual C++ 9.0 SP1 32 and 64 bit libraries*
Windows Vista	Microsoft Visual C++ 8.0 SP1
	Microsoft Visual C++ 9.0 SP1 32 and 64 bit libraries*

* 64 bit libraries are available only for *vrfSim* (simulation engine).

Using Libraries and Binaries Built with Visual Studio 2005 and Later

All MÄK products built with Microsoft Visual Studio require the C Runtime Library to function. The C runtime libraries have always been available from Microsoft for download, they are also installed on a user's machine when a Microsoft compiler is installed. The C runtime libraries are not part of the normal Windows installation. For customers who plan to use MÄK products on machines that do not have a compiler installed, MÄK has historically distributed a copy of the C Runtime Libraries with MÄK products. These libraries were put in the *bin* directory used by the MÄK products. MÄK products would then use the libraries in the *bin* directory and customers would not have a problem if copies of the libraries were not already installed.

Unfortunately, with the release of the new C Runtime Libraries required by Microsoft Visual Studio 2005 (MSVC++8.0) and later, the libraries can no longer just be copied into the *bin* directory of an application. The libraries need to be installed correctly into Windows system folders. (The process is actually a little more complicated, a manifest file needs to be created to tell Windows where to find the libraries.)

To accommodate this change, MÄK is distributing the Windows installer for the C runtime libraries with all MÄK products released for MSVC++8.0 and later. The 32-bit installer is named *vcredist_x86.exe*; the 64-bit installer (if supported) is named *vcredist_x64.exe*. They are in the base directory of any installed MÄK product that requires them.

Running the installer requires Administrator privileges for the machine the installer is run on. MÄK has chosen to not integrate the MÄK installer and the Microsoft installer so as not to require users to have Administrator privileges to install MÄK products. Therefore, if you who do not have a compiler installed, or get error messages like "Software has not been installed correctly, please re-install", you must apply the patch.

For more information see this Microsoft URL:

http://msdn2.microsoft.com/en-us/library/ms235299.aspx

You must ensure that the preprocessor defines `_SECURE_SCL=0`, and `_HAS_ITERATOR_DEBUGGING=0` are set for MSVC++8.0 and MSVC++9.0 builds. If these are not set, random crashes and assertions may be encountered during runtime.

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, including the 3D Front End requires approximately four GB of disk space.

Compiler Compatibility on Windows

MÄK provides versions of product releases that have been compiled with Microsoft Visual C++ 7.1. 8.0, and 9.0. When you run MÄK products together on the same computer, for example, VR-Forces and MÄK 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 3.13.2. If you are building for HLA and want to link with the MÄK RTI, use MÄK RTI 3.x.

VR-Forces 3.12 is a parallel release to MÄK Plan View Display 2.12.

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 Qt Software. The VR-Forces GUI was built with Qt release 4.5. Qt is available as a free download under the LGPL version 2.1 license at www.qtsoftware.com. This version should be satisfactory for most VR-Forces customers. If you need a Qt commercial license, you must purchase the license from Qt Software.

If you are using Qt 4.5 with Microsoft Visual Studio 2005 or 2008, you must download a hotfix. Details are available at:

http://doc.trolltech.com/4.5/compiler-notes.html#visual-studio-windows

FLEXlm Support

VR-Forces 3.12 uses FLEXlm 11.6. You must copy the files in *./flexlm11.6* to your license server directory. You do not need to change license files (unless your maintenance agreement has expired.)

Third-Party Library Requirements

VR-Forces 3.12 uses Boost 1.37.

Patch Required for AMD Dual-processor Windows PCs

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

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

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

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 3.12 was built against VR-Link 3.13.2 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 3.12 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 3.12. VR-Forces 3.12 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 3.12 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 3.12 APIs are not compatible with VR-Forces 3.11.x or earlier. Scenarios and object parameter databases created with VR-Forces 3.11.x are forward compatible.

RPR FOM Versions Supported

VR-Forces 3.12 has built-in support for versions 1.0 and 2.0 (draft 6 and 17) of the RPR FOM and RPR FOM 2.0, draft 17 with DI-Guy extensions. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, VR-Forces 3.12 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 3.12 relies on the LgrControl and ViewControl custom MÄK extensions.

New Features and Updates

VR-Forces 3.12 has many new features and updates. Features are cross-referenced to the relevant sections in documentation.

- New licensing requirements. An extra license feature is now required to run VR-Forces applications built in debug mode. This extra license is only required for the Windows versions. There is no additional cost for this license; as long as you are under active maintenance you can send your license to keys@mak.com and ask to have it regenerated with the new license.
- Installation key requirement: The VR-Forces Development Kit requires an installation key. Please obtain the installation key from your MÄK salesperson before you install VR-Forces.
- DI-Guy integration. Support for DI-Guy characters is fully integrated into VR-Forces. It is no longer necessary to install the Human Characters for MÄK Products product install and configure use of DI-Guy characters. The fdc_fuelman characters that was included with the Human Characters product is not part of VR-Forces 3.12 due to undesirable visualization behaviors. If you need it, you can add it using the Entity Editor.
- Simulation connections: You now choose the simulation protocol to run in the Simulation Connections dialog box, rather than from the Windows Start menu or by running customized batch files or scripts (although you can still use scripts and batch files if you want to.) (Section 4.1, "Starting VR-Forces", in *VR-Forces Users Guide*)
- Support for Microsoft Visual C++ 9.0 32 and 64 bit libraries. For more information, please see "Support for MS-Windows 64-Bit," on page 11.
- Some terrain databases that were included with prior releases have been dropped from this release. The list of dropped terrains include Monterrey, Colorado-dted, Hunter, and Antigua-shape.
- New terrain databases have been included in this release. These include Little Pond and San Luis Obispo. The terrain directories have been reorganized to be more consistent.
- The Component Code Generator is no longer supported.
- 2D front end changes:
  - Console display in object information dialog boxes: The console view on the object information dialog boxes is no longer a tab. It is a a permanent window attached to the bottom of the dialog box. (Section 10.3, "Configuring Entity Console Messages", in *VR-Forces Users Guide*)
  - Output Log display in main window: By default, the front-end displays a console log that displays front-end and back-end console messages. You can hide or display the console like any other toolbar.
  - Concave areas are now supported.

- Message received notification: When entities receive a console message, their icons change to indicate that a message is available to be viewed. (Section 10.3.3, "Displaying Object Console Warning Icons", in *VR-Forces Users Guide*)
- Configurable toolbars: You can now customize the icon toolbars for frequently used tasks. You can edit existing toolbars and create new toolbars using any of the menu commands. (Section B.4, "Customizing Toolbars", in *VR-Forces Users Guide*)
- Display line-of-sight over several points using the Terrain Profile dialog box. (Section 18.9, "Analyzing Terrain Using a Terrain Profile Line", in *VR-Forces Users Guide*)
- The Entity Information dialog box now displays an entity's current task in the State tab.
- Improved configuration of measurement units in dialog boxes. (Section 6.4.11, "Changing Measurement Units", in *VR-Forces Users Guide*)
- Plugins dialog box: The Plugins dialog box displays a list of the plug-ins loaded. (Section 4.11, "Viewing a List of Loaded Plug-ins", in *VR-Forces Users Guide*)
- Sound is no longer supported.

◆ Behavior and modeling changes

- Concurrent tasks: A concurrent task does not interrupt an entity's current task. It executes at the same time as the task, similarly to a set data request. This new behavior is primarily for the Send Radio Task, Send Radio Set, and Send Text Message tasks. None of the movement tasks can run concurrently with each other. (Section 11.1.1, "Concurrent Task Execution", in *VR-Forces Users Guide*)
- Support for collateral damage of entities: In previous releases of VR-Forces, entities were damaged only as the result of direct fire. If a lifeform was standing next to a vehicle and the vehicle was damaged, the lifeform was not affected. The new collateral damage model allows entities to take damage from the destruction of other entities within a certain range. Collateral damage is configured in the damage tables. (Section 8.3.3, "Damage Probability Tables", in *VR-Forces Configuration Guide*)
- Support for tactical smoke: Entities that are configured to do so can now fire tactical smoke. Smoke is created as a group of hazard clouds that expand and then dissipate. Tactical smoke blocks line-of-sight. (Section 17.6, "Displaying Tactical Smoke", in *VR-Forces Users Guide*)
- Ability to add components to remote entities: You can now attach VR-Forces components to remote, non-VR-Forces entities. This feature is primarily beneficial for attaching sensors to entities, such as UAVs, so that VR-Forces can monitor what the entities are detecting. This feature does not allow you to control a remote entity in any way. (Section 15.9, "Writing Plans for Remote Entities", in *VR-Forces Users Guide* and Section 8.12, "Configuring VR-Forces Components on Remote Entities", in *VR-Forces Configuration Guide*)

- Improved modeling of aggregates: VR-Forces now models aggregated and disaggregated aggregate entities (formerly called true aggregates and pseudoaggregates). Aggregates can transition between the two states to meet the needs of a simulation. Aggregation and disaggregation can be done manually or automatically. (Section 3.6, "Aggregate Entities" and Chapter 9, *Working with Aggregate Entities*, in *VR-Forces Users Guide*)
- Restart Plan command can be issued in a plan or through the GUI. (Section 15.11, "Restarting a Plan", in *VR-Forces Users Guide*)
- Abandon Plan command can now be included in a plan. (Section 15.12, "Abandoning a Plan", in *VR-Forces Users Guide*)
- Orbit task: Fixed wing entities can fly in a circle. (Section 11.6.17, "Orbit", in *VR-Forces Users Guide*)
- Set Notify Level set data request: Lets you set the notification level for the entity console. (Section 12.22, "Set Notify Level", in *VR-Forces Users Guide*)
- Task interruption prompt: VR-Forces displays a message when an independent task will interrupt a task or plan. Independent tasks have always interrupted the current task or plan. This message is a usability feature to ensure that you know about the effect of issuing a new task. (Section 11.1.2, "Overriding Tasks and Plans", in *VR-Forces Users Guide*)
- Set Surrender set data request: Lifeforms can surrender (prisoner-of-war). While surrendered, a lifeform will not fire on another entity and will not be fired on. (Section 12.32, "Set Surrendered", in *VR-Forces Users Guide*)
- Laser targeting: VR-Forces now supports laser-targeted weapons systems. They are mounted on helicopters by default and a lifeform that can lase targets is also included. (Section 13.3.2, "Lasing Targets", Section 11.6.10, "Lase Target", Section 12.19, "Set Lase Autonomous", and Section 12.20, "Set Laser Code", in *VR-Forces Users Guide*)
- Indirect artillery: VR-Forces supports indirect artillery. Indirect artillery is not fired by specific entities. You simply configure detonations within a specific area. (Chapter 14, *Managing Indirect Artillery*, in *VR-Forces Users Guide*)
- Set Force set data request: Lets you set an entity's force at runtime. (Section 12.15, "Set Force", in *VR-Forces Users Guide*)

◆ Configuration changes:

- Session management: The VR-Forces session concept has been enhanced to improve ease-of-use when you run multiple front-ends in a session. When a front-end is part of a session, changes to the loaded scenario or terrain are automatically propagated to all front-ends that are in the session. (Section 4.2, "Managing a Front-end's Session Connection", in *VR-Forces Users Guide*)

- Improved management of performance versus quality for 3D viewers: The Networking Options dialog box has been replaced with the Performance Options dialog box, which provides more intuitive management of network thresholds. (Section 1.7, "Balancing Visual Quality Against Network Performance", in *VR-Forces Configuration Guide*)
- Force hostility matrix: Hostility relationships are no longer part of an entity's OPE file. They are now managed through the front-end and saved as part of a scenario in the hostility file. (Section 13.2, "Managing Force Hostility Relationships", in *VR-Forces Users Guide*)
- Ability to configure aggregates and aggregate formations in the Entity Editor. (Section 5.5, "Editing Aggregates", in *VR-Forces Configuration Guide*)
- Improved support for Qualnet: The Communication Model Settings dialog box lets you quickly enable the external communications model. (Section 9.2.2, "Enabling the External Communication Model", in *VR-Forces Configuration Guide*)
- *vrfSimSettings.mtl* file: Persistent back-end settings are now stored in a settings file.
- The **objectConsoleNotifyLevel** parameter in *vrfSim.mtl* lets you set the notification level for the object console.
- `--deviceAddress` command-line argument: Specifies the address of the ethernet card to use for UDP traffic. Use this argument if your computer has multiple network cards. (Or specify the device address in the Simulation Connection Settings dialog box when you start VR-Forces.)
- `--fullyLoadTerrain` command-line argument: Specifies that VR-Forces should load all subpages of a paged terrain when the master terrain is loaded. (*vrfSim* only)

◆ Scenario management changes:

- Automatic saving and checkpointing by simulation time: The standard Save Scenario feature has been integrated with checkpointing. When a scenario is saved, a checkpoint is created at the current simulation time. If the scenario has previously been saved at that time, you are prompted to confirm that you want to overwrite it. Once a scenario is run, it will not be saved to the base scenario name. This improved feature largely removes the possibility of overwriting a base scenario with a scenario at some simulation time other than 0:00:00:00. (Section 5.3, "Saving a Scenario", in *VR-Forces Users Guide*)
- Ability to name scenarios: When you create a scenario you can give it a name. This is a convenience feature for use with the new session behaviors. (Section 5.1, "Creating a Scenario", in *VR-Forces Users Guide*)

◆ API changes:

- Settings Manager: The back-end can now load and save settings. This feature is managed by the Settings Manager. Settings are saved to *vrfSimSettings.xml*. (Section 16.2, "The Settings Manager", in *VR-Forces Developers Guide*)

  - The separator character for specifying resources has changed from '.' to '|'. This allows you to use periods in resource names, for example, 12.7mm gun.

- TDB Tool improvements:
  - Improved ability to build large terrains: If the TDB Tool detects that the terrain conversion process may exceed the amount of available memory, it prompts you to change the conversion algorithm. (*Building Large Terrain Databases*, in *MÄK Terrain Database Tool Users Guide*)
  - Ability to load large bitmaps: The TDB Tool can detect when bitmaps are too large to load into memory and offers options for how to handle the file. (Section 2.3.4, "Loading Large Images", in *MÄK Terrain Database Tool Users Guide*)
  - Support for vector product format (VPF) terrains. This feature supersedes the support for VMAP terrain. (Section 2.4.4, "Importing VPF Files" and Section 7.1.8, "VPF Metafile Records", in *MÄK Terrain Database Tool Users Guide*)
  - Ability to load all pages of a paged terrain in one operation. In previous versions of the TDB Tool, you could load all pages in a paged terrain, but you had to load them page by page. In this release, you can load all pages at once by choosing **Project → Load All Paged Terrain**.
- The 3D Front End has been completely rewritten. It is now based on VR-Vantage™, instead of the Vega-Prime-based MÄK Stealth. For more information, please see *VR-Forces 3D Front End Users Guide* and VR-Forces 3D Front End Release Notes.

Support for MS-Windows 64-Bit

VR-Forces now includes a 64-bit version of the simulation engine (*vrfSim.exe*) on the windows platform (VC++ 9.0 only). For this specific platform, the VR-Forces installer comes with both 32-bit and 64-bit executables and DLLs, and both versions are available from the start menu. The 64-bit DLLs and executables included in the installation are identifiable by the string "_64" in their names.

Toolkit developers can build 64-bit versions of back-end examples and executables. The back-end example projects provided with the VC++ 9.0 installation can be built as either 32-bit or 64-bit. Follow the procedures for building examples described in the "Examples" section of the class documentation (found under "Additional API Documentation" on the main page).

- 64-bit support is available for the simulation engine only. All other applications (2D and 3D Front Ends) and tools (for example, the TDB Tool) included with VR-Forces are 32-bit executables.
- OpenFlight terrain databases cannot be directly loaded in the 64-bit back-end (the OpenFlight API is 32-bit only). Use the 32-bit TDB Tool application to first convert the OpenFlight to GDB terrain format and then load the resulting data into the simulation engine.

- A new command line argument, `--fullyLoadTerrain` has been added to the simulation back-end to allow you to force automatic loading of all terrain pages in terrain databases that are structured for paging. While this option will work for both 32-bit and 64-bit versions of the simulation engine, you may find it particularly helpful when constructing very large paged terrain databases (that is, terrain databases that exceed 4 GB in memory) that can only be loaded on a 64-bit system.

Documentation Updates

All manuals and online help have been updated for this release. Some dialog boxes have changed slightly since the manuals were completed. The following information was not available at the time the manuals were completed:

- A new manual, *VR-Forces Getting Started Guide*, has been added to the documentation set. This manual provides a quick introduction to the major features of VR-Forces and tips to new users to help them avoid common mistakes.
- *VR-Forces Users Guide* describes how an entity's state is preserved when it is copied and pasted. The manual does not make it clear that an entity's task is not preserved as part of the copy operation. A newly pasted entity does not retain the original entity's task. However, if an entity's plan is included in a Paste Special operation, the entity immediately starts executing the plan.
- If you are running multiple front-ends, you might not be prompted to preserve changes to a scenario, which could result in loss of these changes. VR-Forces prompts you to save changes to a scenario before events such as running or rewinding the scenario. The knowledge about changes to a scenario is maintained in the front-end. However, front-ends do not communicate this information to each other. If you are running multiple front-ends, the different front-ends do not know about changes made from another front-end. Therefore, if you make a change from front-end A, if you initiate a run or rewind operation from front-end B, you would not be prompted to save the changes. If you are running multiple front-ends, it is the responsibility of the persons using them to ensure that changes are saved.
- The Antigua-shape database is referenced in several places in the documentation set. However, it is no longer included in the release. For an example of a terrain that includes shapefile data, please see the Little Pond terrain.
- A Close button has been added to the Simulation Connections Configuration dialog box. Clicking Close saves changes to the configurations and exits the dialog box without starting VR-Forces. (Cancel cancels all changes and exits the dialog box.)
- The *configurationsSettings.xml* file, which is documented in *VR-Forces Configuration Guide*, has been renamed *connectionsSettings.xml*.

- Running VR-Forces in a simulation with multiple back-ends on machines configured with multiple network cards requires careful setup. You must identify and specify the IP address of the appropriate network card. You can specify the correct address in the application launcher, in or command line arguments, and in RTI configuration parameters (HLA-only). See the table below for the command line arguments necessary to configure VR-Forces under these conditions:

Table 2: Command line arguments for specifying network addresses

	DIS	HLA
vrfSim	`--deviceAddress`	Consult your RTI's documentation.
vrfGui	`--deviceAddress` `--hostAddressString`	`--hostAddressString`

  You must specify the internet address of the card (on the local machine) you want to receive TCP traffic on. *vrfGui.mtl* and *vrfSim.mtl* have parameters that are equivalent to `--hostAddressString`. Use the `--deviceAddress` command to specify the address of the card to use on the local machine for UDP traffic.

  For configuration details, please see the following:

  – Section 4.8, "Command Line Options", in *VR-Forces Users Guide*

  – Chapter 2, *VR-Forces Configuration Files*, in *VR-Forces Configuration Guide*.

- Qt Linguist has changed slightly from the screen shots in *VR-Forces Users Guide*. VR-Forces now includes the following translation files:
  – *entityEditor_untranslated.ts* – Entity Editor
  – *pvd_untranslated.ts* – VR-Forces front-end
  – *vrfgui_untranslated.ts* – VR-Forces Front-end
  – *vrfsim_untranslated.ts* – VR-Forces Front-end
  – *tdbtool_untranslated.ts* – TDB Tool
  – *vrfOpdEditor_untranslated.ts* – VRF OPD Editor
  – *cw_untranslated.ts* – FlexLM licensing dialog boxes.
- *VR-Forces Developers Guide* does not include any documentation for how to use the Entity Editor and OPD Editor Plug-in API. The following section provides an introduction to this API.

The Entity Editor and OPD Editor Plug-in API

The Entity Editor and OPD Editor plug-in API is a programming interface for customizing the Entity Editor and OPD Editor applications. VR-Forces provides a simulation API that allows developers to add functionality to the simulation engine, such as adding a new kind of sensor, or extending the parameters of an entity. The Entity Editor and OPD Editor Plug-in API complements the simulation API, enabling developers to customize those applications so that end-users can configure their custom extensions to the simulation engine using the Entity Editor and OPD Editor.

You can write a single plug-in that will work for both the Entity Editor and the OPD Editor. An Entity Editor or OPD Editor plug-in provides the following capabilities:

- Entity Editor: Enable new parameters and descriptors to be saved as part of simulation model set. The API does not currently support extending the graphics in the Entity Editor. However, a plug-in is still essential in order to ensure that new customer parameters and descriptors are saved as part of the simulation model set, whenever a user edits and saves their work.
- OPD Editor: Enable new parameters and descriptors to be saved as part of simulation model set and allow end-users to configure new parameters graphically.

! The Entity Editor and OPD Editor API does not support extending the editors with custom graphical elements. However, any built-in *DtReaderWriter* data member types used in extended descriptor or object parameter classes will be viewable and editable by an end-user in the OPD Editor.

Common Uses Cases

The most common situations in which VR-Forces customers decide they need to develop an Entity Editor or OPD Editor plug-in are:

- Configuring new kinds of components: The developer extends an entity with a new capability, for example, a new kind of sensor. This new component requires some parameters for configuration. The developer would like end-users to be able to configure the new sensor parameters using the OPD Editor.

  API examples that add new components include radarWarnRx, stopToShoot, and subtask.

- Extending object parameters: The developer extends an entity with new object parameter data, for example, color. The developer would like end-users to be able to configure the entity with this new parameter using the OPD Editor.

  The addEntity API example shows how to extend an entity with new parameter data.

In both of these use cases, the goal is to allow an end-user to configure the new component descriptors or object parameters using the OPD Editor. However, once end-users have configured their new custom extensions using the OPD Editor, they can save the work and then load it into the Entity Editor to complete other configuration activities (that do not involve the custom extensions). The only thing required to ensure that any custom extensions to the simulation model set are maintained is that the end-user loads the plug-in to the Entity Editor as well as the OPD Editor.

Writing an Entity Editor or OPD Editor Plug-in

To integrate a simulation engine plug-in (or an extended the simulation engine application) with the Entity Editor and OPD Editor:

1. Create a new project for your plug-in. We recommend that you copy an example project (such as stopToShoot) in the *./examples* directory and rename the project.
2. Edit the file *opdEditorPlugin.cxx*, as follows:
   a. Remove references to any component or parameter classes from the original project, and replace them with references to your custom descriptor and object parameter classes.
   b. Include the *editorPluginExtension.h* header file:

      ```
      #include “editorPluginExtension.h”
      ```

      Also include header files containing declarations of any new component descriptor and object parameter files you have created.
   c. Inside the function DtInitializePlugin(), add the calls necessary to include your custom classes in the plug-in. There are calls available to add sensors, controllers, actuators, and object parameter classes (see *editorPluginExtension.h*).

      For example, in the stopToShoot example, the plug-in contains a new kind of component descriptor. In the call to addController(), it passes in the name of the component (as it will appear in the OPD entry), the name of the descriptor (as it will appear in the OPD entry), and a creator function that returns a new instance of the descriptor class. The new descriptor is incorporated into the plug-in as follows:

      ```
      #include "editorPluginExtension.h"
      #include "stopFireDesc.h"
      …

      extern "C" {

        DT_DLL_PLUGIN bool DtInitializePlugin(DtEditorExtensionInterface*
         opdIface, DtPluginInformation& info)
        {
          …
          opdIface->addController(MyStopToFireType,
            MyStopToFireDescriptorType,
            MyStopToFireComponentDescriptor::creator);

        return true;
      ```

```
    }
  }
```

3. Build the plug-in. Given that you have modeled your new plug-in project on an existing project, the DLLs will be created in the *./bin* directory of your VR-Forces application.
4. (Optional) Create an MTL file to enable automatic loading of your plug-in. The easiest approach is to copy an existing MTL file from an example as a starting point. Edit the file and change the name of the original plug-in to the name of your plug-in. Alternatively, you can manually load the plug-in using the `--loadPlugin` command line argument.

   Copy the MTL file to the *./plugins* directory.

Once you have built your plug-ins and have copied the MTL file to the *./plugins* directory, your new custom descriptors and object parameters will be available for editing in the OPD Editor, and maintained across load and save operations in the Entity Editor.

Examples

The following example projects demonstrate how to write an Entity Editor and OPD Editor plug-in. They work in conjunction with a corresponding simulation API example that extends the VR-Forces simulation engine in some way.

- addEntity
- radarWarnRx
- stopToShoot
- subtask.

Bug Fixes

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

- You cannot create a new scenario with a Linux GUI and a Windows back-end.
- Fixed-wing entities generate too many PDUs when moving to a waypoint in a straight line.
- Embarked entities cannot shoot or be shot at by other entities.
- Fixed-wing entities did not respond to Turn to Heading tasks.
- The Object Information Toolbar did not work correctly. It has been removed from VR-Forces and the **objectInformationToolbar** parameter has been removed from *vrfGui.mtl*.
- Suicide bomber entities now start out in an unarmed state by default. Previously, they started out in a proximity mode. This could lead to unexpected detonations, depending where they were initially positioned in the terrain.
- Collision avoidance for individual lifeforms now works correctly on geocentric terrain databases.

- Images can now be correctly draped over a terrain in geodetic coordinates.
- The -m command line argument of TDB Tool now supports wildcards. This can be useful when importing paged terrain databases consisting of large numbers of individual files.
- Embarking a DI on another entity no longer results in the DI losing its current appearance. However, its posture may still change (for example, it may switch to a sitting posture, when embarked on a vehicle such as a truck).
- The `--sessionId` command line argument now works.
- You can now edit sensor geometry though the OPD Editor.
- VR-Forces now correctly ignores HLA Federation Save messages, and allows the federation to continue after the save.
- VR-Forces (and TDB Tool) no longer crash when loading C7L terrain databases containing water features.
- Non-VR-Forces entities can now be embarked on VR-Forces entities, through the API. In a non-VR-Forces VR-Link application, you can establish an embarkation relationship between an external simulator (embarking entity) and a VR-Forces entity (entity to embark on) by calling setHostGlobalId in remote simulator's entity state repository.

Known Problems

VR-Forces Release 3.12 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.
- 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.
- 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.
- 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.
- 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.
- 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 off screen.

- 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.
- Importation of polygonal terrain elevation data from shapefiles does not work. This does not affect importation of shape feature data.
- The default radar signature for ground vehicles is 500 meters. This is too short a distance for most fixed-wing aircraft to detect and fire on a ground vehicle. To make ground vehicles vulnerable to aircraft, increase the radar signature to a distance that is appropriate for the aircraft in your simulation.
- Lifeform entities cannot be configured with multiple weapon systems.
- Fixed-wing entities do not respond to 'Turn to Heading' tasks.
- Surface entities are not able to come to a stop as quickly as expected. The model does not take into account reverse thrust from engines.