



VR-Vantage 1.0 Release Notes

This document provides the following release-specific information for VR-Vantage 1.0:

Introduction to VR-Vantage	2
Systems Supported and System Requirements	3
VR-Vantage for Windows	3
3D Video Boards Supported	3
Patch Required for AMD Dual-processor Windows PCs	5
Network Compatibility	5
FOM Support	6
New Features and Product Updates	6
Differences Between Stealth 7.0 and Stealth 6.x	6
Installing VR-Vantage on Windows Vista	7
Building Example Applications	7
Building Example Applications	7
Documentation Updates	8
Loading and Saving Multichannel Configurations	8
Configuring Head-Up Displays	8
Attaching the Observer at Startup	9
Starting the Distributed Code Generator	10
Known Problems and Product Restrictions	10

Copyright © 2009 VT MÄK, 68 Moulton St., Cambridge, MA 02138 All rights reserved.
MÄK Technologies®, VR-Forces®, RTIspey®, and VR-Link® are registered trademarks of VT MÄK.
Document ID: VRV-1.0-3-090330

Introduction to VR-Vantage

The VR-Vantage product has three components: MÄK Stealth, Vantage-IG, and VR-Vantage Toolkit. The Stealth and Vantage-IG applications are end-user graphical user interfaces (GUIs) that provide a three-dimensional view of terrain, vehicles, and other objects. The features of the two application are largely the same, but are oriented towards different uses, as follows:

- The Stealth is optimized for customers who need a 3D environment for situational awareness, simulation debugging, or after action review. The Stealth provides the most data about your networked virtual world, and presents it in a clear and accessible way.
- VR-Vantage-IG is optimized for customers who need an out-the-window (OTW) scene or a view from a remote camera. Vantage IG is a configurable desktop image generator that is designed for easy integration into desktop trainers and war gaming simulations.



Because of the similarities between MÄK Stealth and Vantage-IG, in the remainder of this release note, unless stated otherwise, all descriptions of the Stealth and Stealth features apply equally to Vantage-IG.

The VR-Vantage Toolkit is an API that comes with all the software you need to completely rebuild visual applications like the MÄK Stealth or Vantage IG and also allows you to customize and extend these applications to fit your unique requirements. The toolkit also gives you the power to embed any of the MÄK Stealth or Vantage IG capabilities directly into your simulation applications. The VR-Vantage Toolkit is based on OpenSceneGraph, so you can leverage value-added plug-ins built by the OSG community and MÄK partners.

“OpenSceneGraph is an OpenSource, cross-platform graphics toolkit for the development of high-performance graphics applications such as flight simulators, games, virtual reality and scientific visualization. It is based around the concept of a Scene-Graph, providing an object-oriented framework on top of OpenGL.”
(<http://www.openscenegraph.org/projects/osg/wiki/About/Introduction>)

For details about the VR-Vantage Toolkit, please refer to *VR-Vantage Developers Guide*.

Systems Supported and System Requirements

This section describes platform support and system requirements for the VR-Vantage. For the most up-to-date information about systems supported, see the Product Versions page on the MÄK web site at: <http://www.mak.com/support/productversions.php>.

The VR-Vantage is available for the following operating systems:

Table 1: Platforms supported

Operating System	Compiler
Red Hat Enterprise Linux Workstation 4, 5	Default compiler
Windows XP	Microsoft Visual C++ 7.1, 8.0
Windows Vista	Microsoft Visual C++ 8.0

Third Party Library Support

VR-Vantage uses the indicated versions of the following libraries:

- ♦ VR-Link 3.13
- ♦ Qt 4.4.3
- ♦ OpenSceneGraph 2.6.0.

VR-Vantage for Windows

VR-Vantage for Windows requires the following:

- ♦ A Pentium-class PC (or higher) with minimum 2 GHz processor
- ♦ An OpenGL 2.0-compliant graphics card with 256 MB or more of memory.
- ♦ Windows XP SP2 or Windows Vista.
- ♦ 2 GB of disk space.
- ♦ 1 GB of RAM; more is desirable depending on the size of the terrain database and the number of models to be loaded.

3D Video Boards Supported

In general, VR-Vantage 1.0 should support any board that claims to support OpenGL 2.0. VR-Vantage has been tested with current versions of NVidia graphics cards. There are known problems with ATI graphics cards (in particular, the ATI Radeon X1950 Pro).

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, for example, the Logger and a VR-Vantage application, 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.

FLEXlm Support

VR-Vantage 1.0 uses FLEXlm 11.6.

Using Libraries and Binaries Built with Visual Studio 2005 and Later

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

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

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

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

For more information see this Microsoft URL:

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



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

Patch Required for AMD Dual-processor Windows PCs

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

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



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

Network Compatibility

HLA only

VR-Vantage 1.0 is compliant with:

- RPR-FOM 0.5, 0.7, 0.8, 1.0, and a subset of 2.0 (draft 6, 14, and 17)
- MÄK RTI 2.x, 3.x
- Pitch RTI 1.3 C++ interface.

Other RTIs that support the HLA 1.3 specification or the SISO DLC HLA API 1516 (SISO-STD-004.1-2004 version of the IEEE 1516 specification.)

DIS only

VR-Vantage 1.0 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.

FOM Support

VR-Vantage 1.0 has built-in support for versions 0.5, 0.7, 0.8, 1.0, and 2.0, drafts 6, 14, and 17, of the RPR FOM. It also supports VR-Link's ability to support alternative FOMs through the FOM Mapper. By default, VR-Vantage 1.0 uses RPR FOM 1.0.

If you want to use VR-Vantage with VR-Forces 3.9 or later and RPR FOM version 2, draft 17, use the *VR-Link20017-1.fed* file.

New Features and Product Updates

VR-Vantage 1.0 is a new product. It includes MÄK Stealth 7.0, Vantage IG 1.0, and VR-Vantage Toolkit.

Differences Between Stealth 7.0 and Stealth 6.x

Much of the basic functionality of Stealth 7.0 will be familiar to users of previous version of the Stealth. However there are some important differences. Some of the most important new features are:

- All configuration is done through the GUI, not through MTL configuration files
- You connect to simulations at runtime. You can configure connections through the GUI and can enable multiple connections at the same time.
- You can build your terrain dynamically at runtime by importing elevation data, raster images, a feature data. You can save the dynamically created terrains for quick loading in future sessions.
- The Stealth has a simplified, and yet highly flexible navigation model. All navigation is done from the keyboard using a layout similar to first person shooter games. However, you can remap the keys to suit your needs.
- View modes have been simplified. Keyboard navigation in the different view modes is more consistent and intuitive than previous Stealth releases.
- You can create multiple windows and channels through the GUI and configure them.
- The Stealth is built using Open Scene Graph, not Vega Prime. The toolkit includes the version of OSG used to build the Stealth. Because OSG is open-source software, you do not need any additional licenses to use the API or modify VR-Vantage applications.

Stealth 6.2 Features not Supported by Stealth 7.0

The following features that were part of Stealth 6.2 are not included in Stealth 7.0. They may be included in future releases.

- ♦ StealthXR
- ♦ Screen capture and video capture
- ♦ Mirroring
- ♦ Entity and aggregate information dialog boxes and other informational dialog boxes
- ♦ Aggregate display
- ♦ Group management
- ♦ Hazard clouds
- ♦ Support for metaflight, VSB, and Blueberry 3D terrains
- ♦ Range volumes
- ♦ Joysticks
- ♦ 3D overlay
- ♦ Line-of-sight lines
- ♦ Remote graphics API
- ♦ True-type font icons, image icons
- ♦ Height-above-terrain lines
- ♦ Display of VR-Forces tactical graphics.

Installing VR-Vantage on Windows Vista

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

When you install VR-Vantage large applications on Vista, there may be a delay, of up to several minutes, from the time you try to run the setup program to the time that an installation dialog box is displayed. This is due to how Vista scans setup programs before it executes them. If you experience this problem, turning off User Access Control can reduce or eliminate this delay.

Building Example Applications

The examples and solution files for the VR-Vantage examples were created in Visual C++ 7.0. To use them with Visual C++ 8.0, use the conversion wizard.

Documentation Updates

The VR-Vantage documentation set is completely new. The video tutorials provided with this release were created using the Early Adopter release. Minor aspects of the graphical user interface have changed since the videos were recorded.

The following sections update *VR-Vantage Users Guide*.

Loading and Saving Multichannel Configurations

To save a multichannel configuration:

1. Start VR-Vantage and one or more VR-Vantage Display Engines.
2. In the Display Engine Configuration Editor, connect the display engines.
3. Configure the display engines with the windows and channels that you want them to have.
4. For each display engine in the Display Engine Configuration Editor:
 - a. Select the display engine.
 - b. Click the Save Display Engine Configuration icon.
 - c. Type and name for the configuration.
 - d. Click Save.

To load a multichannel configuration:

1. Start VR-Vantage and one or more VR-Vantage Display Engines.
2. In the Display Engine Configuration Editor, connect the display engines.
3. For each display engine in the Display Engine Configuration Editor:
 - a. Select the display engine.
 - b. Click the Load Display Engine Configuration icon.
 - c. Select the configuration you want to load.
 - d. Click Load.

Configuring Head-Up Displays

In addition to enabling or disabling head-up displays from the Settings menu, as described in *VR-Vantage Users Guide*, you can configure head-up displays on the Application Settings dialog box, Head-Up Display Settings page. You can turn the HUDs on and off and you can adjust the smoothing applied to the HUDs.

Attaching the Observer at Startup

There may be times when you want to attach the observer to an entity as soon as VR-Vantage starts up. For example, suppose you are using Vantage IG as a flight simulator. You might want it to start up and immediately display the HUD view for the aircraft. VR-Vantage has a command-line argument, `--bindObservers`, that lets you bind the observer to an entity using saved views.

To start VR-Vantage with the observer bound to an entity:

1. Create a recording of the simulation that you want to use.
2. Attach the observer to the entity that you want it to be attached to when you start VR-Vantage. If you have multiple observers, create a saved view for each observer that you want to attach automatically.
3. Save the views. Views are saved to a file with the extension `.osrx`.
4. Start VR-Vantage with the `--bindObservers` command-line option and use your saved views file as the argument to `--bindObservers`, as follows:

```
vantageIG --bindObservers observer1,...,observern
../appData/settings/vantageIG/saved_views.osrx
```



If your display configuration has more than one observer, you can save multiple views and automatically bind the observers to the views. The observers are bound to views on a one-to-one basis in the order that the observers are listed in the command and the saved views are listed in the file. If you only have one observer, it is automatically bound to the first view in the file.

Tutorial

This tutorial implements the procedure described in the previous section using files provided with VR-Vantage and MÄK Data Logger. It attaches the observer to one of the aircraft in the `maklandDemo` recording.

1. Start Vantage IG.
2. Load the `makland` terrain.
3. Start MÄK Data Logger.
4. Load `maklandDemo` (either the DIS or HLA version).
5. Connect Vantage IG to the correct simulation network.
6. Run `maklandDemo`.
7. Attach the observer in Mimic mode to `FtrFw4`.
8. Save the view to a file, for example `FtrFw4.osrx`.

9. Shut down Vantage IG.
10. Start Vantage IG with the following command line:

```
vantageIG --bindObservers  
../appData/settings/vantageIG/FtrFw4.osrx
```

VR-Vantage includes an example saved view, *FtrFw4.osrx*, that you can use to try out this example.

Starting the Distributed Code Generator

VR-Vantage Developers Guide explains how to start the Distributed Code Generator from the command line. On Windows, you can also start it from the Start menu, as follows:

- Choose **Programs** → **MÄK Technologies** → **VR-Vantage x.x** → **Tools** → **VR-Vantage Distributed Code Generator**.

Known Problems and Product Restrictions

VR-Vantage has the following known problems:

- ♦ When you load a geocentric terrain, clouds are displayed in the center of the earth.
- ♦ Detonation icons are not displayed.
- ♦ Sometimes if you use **Alt+Tab** to switch applications, VR-Vantage does not receive the **Alt**-Up message. The result is that keyboard input may not work. To get the keyboard to work properly, press **Alt** several times.
- ♦ Embarked entities are not visible. When they disembark, they become visible.
- ♦ SpeedTrees do not align correctly on geocentric terrains.
- ♦ DI-Guy characters may have missing body parts if you run more than one display engine on the same computer. (This could be a master display engine and an external display engine or any other configuration of two display engines.)
- ♦ If a graphics card does not support shaders, you may receive the following error message when you load a terrain that uses them:

```
Shader [Branches] validation failed
```

To work around this problem, set the SpeedTree Performance Profile to Disabled, as follows:

- a. Choose Settings → SpeedTree Settings. The Application Settings dialog box opens to the SpeedTree Settings page.
- b. In the Performance Profile drop-down list, select Disabled.

- ♦ There may be some discrepancies between terminology and title descriptions in *VR-Vantage Users Guide* and the final GUI.
- ♦ VR-Vantage applications may freeze for a period of time if you close a terrain while you are connected to a simulation and there are DI-Guy characters in the scene. The application will eventually unfreeze after a variable period of time. The work-around is to disconnect from the simulation before closing the terrain.
- ♦ On external display engines, particle effects, such as smoke and trailing effects, sometimes restart spontaneously.
- ♦ The MSVC++ 8.0 version of VR-Vantage does not include the armored vehicle HUD.

