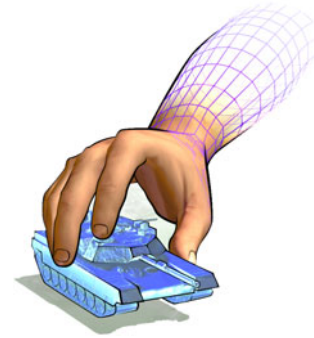




Plan View Display 2.11 Release Notes

This release note contains the following release-specific information for Plan View Display Release 2.11:

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
FLEXlm Support.....	4
Third-Party Library Requirements.....	5
Network Compatibility.....	6
Backwards Compatibility.....	6
RPR FOM Versions Supported.....	7
New Features and Updates.....	8
Upgrade for FLEXlm 11.4.....	8
New Naming Convention for DIS Directories	8
Use of OPCODE in the TDB Tool.....	8
Font Changes	9
Copy and Paste Objects.....	9
Icon Toolbars	9
Miscellaneous API Changes.....	9
Upgrade to Qt 4.3.3.....	9

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: PVD-2.11-2-080204

[Documentation Updates](#)..... 13

[Bug Fixes](#) 13

[Known Problems](#) 14

Systems Supported

Plan View Display 2.11 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 Plan View Display libraries.

Table 1: Platforms supported

Operating System	Compiler
Red Hat Enterprise Linux Workstation 4, 5 Fedora Core Linux 7	Default compiler
Windows XP	Microsoft Visual C++ 7.1 Microsoft Visual C++ 8.0 SP1

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 *vcredist_x86.exe* and is in the base directory of any installed MÄK product.

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

For more information see this Microsoft URL:

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

Building on Linux

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

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

Disk Space Requirements

A full installation of Plan View Display 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, Plan View Display 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 Plan View Display applications, you must link with VR-Link 3.11. If you are building for HLA and want to link with the MÄK RTI, use MÄK RTI 3.x.

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

Qt Release Compatibility

If you want to do development using the Plan View Display GUI API, you need to use Qt, a cross-platform GUI toolkit from Trolltech. The Plan View Display GUI was built with Qt release 4.3.3. A Plan View Display 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

Plan View Display 2.11 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 Plan View Display. 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.

Network Compatibility

HLA only

Plan View Display 2.11 was built against VR-Link 3.11 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 Plan View Display.

Plan View Display 2.11 is compatible with applications that use earlier versions of VR-Link if they support versions of the RPR FOM listed.



Plan View Display is no longer compatible with the DMSO RTI-NG v6, because that RTI was not built using any of the compilers supported by Plan View Display 2.11. Plan View Display 2.11 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 Plan View Display.

DIS only

Plan View Display 2.11 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 Plan View Display 2.11 APIs are not compatible with Plan View Display 2.10.x or earlier.

RPR FOM Versions Supported

Plan View Display 2.11 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, Plan View Display 2.11 uses RPR FOM 1.0.

If you are building an application with the Plan View Display 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 Plan View Display 2.11 relies on the LgrControl and ViewControl custom MÄK extensions.

New Features and Updates

Plan View Display 2.11 has the following new features and updates:

- ♦ Upgrade to FLEXlm 11.4
- ♦ Support for CTDB C71 terrain databases
- ♦ New naming convention for DIS directories
- ♦ Use of OPCODE, a collision detection library, is now available as an alternative to spatial subdivisions for faster terrain intersection calls.
- ♦ MapSyms fonts are no longer distributed
- ♦ GUI features:
 - Font changes
 - Copy and paste objects
 - Icon toolbars for common tasks
 - The LOS Filter has been removed
 - Map area compass
- ♦ Miscellaneous API changes
- ♦ Upgrade to Qt 4.3.3.

Upgrade for FLEXlm 11.4

Plan View Display 2.11 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 Plan View Display.

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*.

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 Plan View Display 3.11. Loading an older version of the font may cause symbol display issues.

Copy and Paste Objects

You can now copy and paste graphical objects. For details, please see [Section 6.6, "Copying and Pasting Objects"](#), in *MÄK Plan View Display User's Guide*.

Icon Toolbars

The Plan View Display now has icon toolbars for common tasks such as opening and saving scenarios and changing the drawing mode.

Miscellaneous API Changes

- ♦ Many elements that were deprecated in previous releases have been removed from the GUI API.
- ♦ The `windowsInfoExtension` example has been renamed `addGuiCoordinateSystem`.
- ♦ The `viewControlObjects` example has been renamed `viewFilter`.

Upgrade to Qt 4.3.3

The Plan View Display 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 Plan View Display GUI and want to migrate to Plan View Display 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 ReleaseBuild. 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 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`.

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.

Bug Fixes

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

- ♦ 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.
- ♦ Vector feature orientation was not displayed correctly in the GUI.
- ♦ 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.

Known Problems

Plan View Display Release PVD-2.11-2-080204 has the following known problems:

- ♦ Intersection lines in intervisibility fans sometime change colors as you change the zoom level.
- ♦ Entity intervisibility with point features does not work correctly.
- ♦ 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.
- ♦ You cannot load vector data on geocentric terrain.
- ♦ 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.

- ♦ Features cannot be selected on geodetic terrain databases in the TDB Tool.
- ♦ Transmitter lines are not drawn if the sender is offscreen.